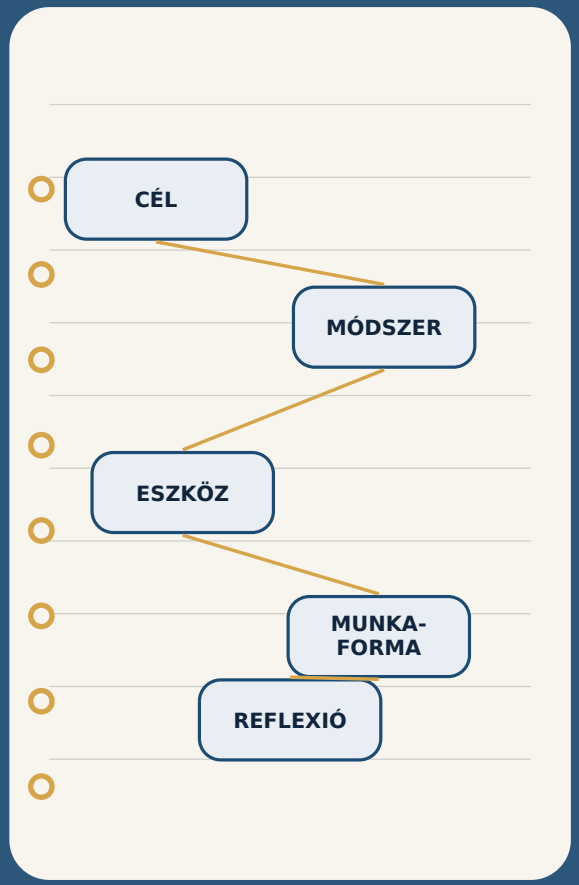
KOVÁCS HENRIETTE

OKTATÁSI ÓRATERVEZETEK

ÓRA IDŐKERETE, MENETE, CÉLOK, DIDAKTIKAI FELADATOK, NEVELÉSI-OKTATÁSI STRATÉGIA (MÓDSZEREK, TANULÓI MUNKAFORMÁK, ESZKÖZÖK)



MAGYAR KULTÚRA EMLÉKÍVEK KIADÓ

Kiadó:
Magyar Kultúra Emlékívek Kiadó

Főszerkesztő:
Kollár Ferenc

Kovács Henriette

OKTATÁSI ÓRATERVEZETEK

ÓRA IDŐKERETE, MENETE, CÉLOK, DIDAKTIKAI FELADATOK,
NEVELÉSI-OKTATÁSI STRATÉGIA
(MÓDSZEREK, TANULÓI MUNKAFORMÁK, ESZKÖZÖK)

ISBN:
978-615-5746-30-7

Minden jog fenntartva.
A kiadvány egyetlen részlete sem sokszorosítható
semmiféle formában a kiadó engedélye nélkül.

TARTALOMJEGYZÉK

Bevezetés 7
Óratervezet - Miért hasznosak a tömbök? 9
Az óra cél- és feladatrendszere 10
Fejlesztendő készségek, képességek 10
Fejlesztendő attitűd 11
Az óra didaktikai feladatai 12
Az óra menete 14
Feladatok 30
Reflexió 31
Óratervezet - A String osztály és a tömbök. Túlindexelés. 32
Az óra cél- és feladatrendszere 33
Fejlesztendő készségek, képességek 33
Fejlesztendő attitűd 34
Az óra didaktikai feladatai 34
Az óra menete 36
Feladatok 46
Reflexió 47
Óratervezet - Tömb elemeinek alapértelmezett értékei. Tömb feltöltése. 48
Az óra cél- és feladatrendszere 49
Fejlesztendő készségek, képességek 50
Fejlesztendő attitűd 50
Az óra didaktikai feladatai 51
Az óra menete 53

Feladatok 63
Reflexió 64
Óratervezet - Tömb elemeinek feltöltése billentyűzetről, kiíratása vesszővel elválasztva. 66
Az óra cél- és feladatrendszere 67
Fejlesztendő készségek, képességek 67
Fejlesztendő attitűd 67
Az óra didaktikai feladatai 68
Az óra menete 70
Feladatok 81
Reflexió 82
Óratervezet - Tömb elemeinek kiíratása vesszővel elválasztva. Inicializáló blokk. 83
Az óra cél- és feladatrendszere 84
Fejlesztendő készségek, képességek 85
Fejlesztendő attitűd 85
Az óra didaktikai feladatai 86
Az óra menete 88
Feladatok 100
Reflexió 101
Óratervezet - Véletlenszámok generálása. Tömb elemeinek feltöltése véletlenszámokkal. 102
Az óra cél- és feladatrendszere 103
Fejlesztendő készségek, képességek 104
Fejlesztendő attitűd 104
Az óra didaktikai feladatai 105
Az óra menete 107
Feladatok 121
Házi feladat 122
Reflexió 124

Óratervezet - Alapvető adatgyűjtési és feldolgozási algoritmusok 126
Az óra cél- és feladatrendszere 127
Fejlesztendő készségek, képességek 127
Fejlesztendő attitűd 128
Az óra didaktikai feladatai 129
Az óra menete 131
Feladatok 143
Reflexió 144
Óratervezet - Maximumkiválasztás. Tömbök elemein végzett műveletek. 146
Az óra cél- és feladatrendszere 147
Fejlesztendő készségek, képességek 147
Fejlesztendő attitűd 148
Az óra didaktikai feladatai 149
Az óra menete 151
Feladatok 165
Reflexió 166
Óratervezet - Összetett elemtípusú tömbök. Tömbelemek cseréje. 168
Az óra cél- és feladatrendszere 169
Fejlesztendő készségek, képességek 170
Fejlesztendő attitűd 170
Az óra didaktikai feladatai 171
Az óra menete 173
Feladatok 191
Reflexió 192
Óratervezet - Tömb elemeinek megfordítása. Gyakorlás. 193
Az óra cél- és feladatrendszere 194
Fejlesztendő készségek, képességek 194

Fejlesztendő attitűd 194
Az óra didaktikai feladatai 196
Az óra menete 198
Feladat 208
Reflexió 215
Foglalkozás tervezete – Az ismétlés - iteráció. Ciklusok fajtái, átalakíthatóságuk, alkalmazásuk. 216
A foglalkozás cél- és feladatrendszere 217
Fejlesztendő készségek, képességek 217
Fejlesztendő attitűd 218
Az óra didaktikai feladatai 219
A foglalkozás menete 221
Feladatok 236

BEVEZETÉS

A tanár szerepe kulcsfontosságú a pedagógiai, oktatási folyamatban, mert a tanuló fejlődését közvetlenül befolyásolja. A tanári munka során fejleszteni kell azokat a kompetenciákat, amelyek által a tanulók élet- és munkaképessé válnak, valamint nagyfokú támogatást kell nyújtani a tehetséges, a hátrányos helyzetű és a fejlődési zavarokkal küzdő tanulóknak, hogy képességeikhez mérten valósíthassák meg céljaikat.

Nagyon fontos a szakmai fejlődésért az elkötelezettség és szakmai felelősségvállalás. Munkánk során elengedhetetlen a szaktárgyi tudás naprakészen tartása. Az informatika területén pedig különösen fontos az önképzés és a folyamatos továbbképzés.

A szakmai tartalom mellett a módszertani megújulás is nagyon lényeges. Az informatikus hallgatók szinte egész életükben a véget nem érő (tovább)tanulásra lesznek kényszerítve, így fontos, hogy mind a tanulási, mind pedig az értékelési folyamatokban már olyan eszközöket és módszereket alkalmazzunk, amelyek motiválják őket, felkeltik érdeklődésüket.

A pedagógiai munka releváns része a pedagógiai eszköztár folyamatos fejlesztése és megújítása is. A generációváltás, a megváltozott világ mindig új kihívások elé állítja a pedagógust. A társadalmi változások arra kényszerítenek bennünket, hogy próbáljunk meg olyan új módszereket és eszközöket alkalmazni, amelyek a tanított generációknak segítik a tananyag befogadását. A gyorsan változó informatika területén ez még fontosabb, mint más területeken. A pedagógiai eszköztár fejlesztése elsősorban a tanulói visszajelzések, visszacsatolások alapján történik. Általános szabályok nincsenek egy-egy új módszer, nevelési elv használatára, hiszen ami az egyik tanulócsoportban működik, az még nem garantálja, hogy egy másikban is működőképes lesz.

A tanári tervező munka legkisebb egysége az órák, foglalkozások tervezete, mely tartalmazza az óra/foglalkozás általános adatait, tananyagát, cél- és feladatrendszerét, a fejlesztendő készségeket, képességeket, a fejlesztendő attitűdöt, az óra didaktikai feladatait, a tantárgyi kapcsolatokat, a felhasznált forrásokat; egy táblázatban az óra/foglalkozás időkeretét, menetét, célját, didaktikai feladatait, a didaktikai feladatok megvalósításához a nevelési-oktatási stratégiát, azaz az alkalmazott módszereket, tanulói munkaformákat, eszközöket; a táblaképet, a feladatokat és az első tíz óra esetén az órához tartozó reflexiókat. Ez a tanár személyes tervezete, mely különböző formákat ölthet.

A könyvben található tizenegy tervezet formailag egységes, melyek megalkotásakor az vezérelt, hogy az órák/foglalkozások menete átlátható, egyértelmű, könnyen követhető, letisztult, ezáltal könnyen tanítható legyen. Ennek eléréséhez a vizualitást hívtam segítségül.

A táblázatban az óra/foglalkozás menete szélesebb oszlopba került, hiszen itt található az oktatott tananyag, a konkrét feladatok, a megoldások, a feltett kérdések és a várható válaszok. Abban az esetben, ha nem tervezetet, hanem vázlatot írunk, akkor nem kell feltüntetni a várható tanulói válaszokat. A célok, didaktikai feladatok külön oszlopban, az óra menete mellett találhatók, a nevelési-oktatási stratégiát pedig, azaz az oktatási módszereket, a tanulói munkaformákat és az eszközöket három oszlop helyett egy oszlopban helyeztem el, megkülönböztetésükre halvány háttérszínezést alkalmaztam. Az oktatási módszereket a piros, a tanulói munkaformákat a kék, az eszközöket a sárga szín jelöli. A megjegyzéseket sem külön oszlopba, hanem a nevelési-oktatási stratégia oszlopába helyeztem el, háttérszín nélkül.

Tíz tervezet óratervezet, melyek tartalmilag összefüggnek, az első hat óratervezet egy tanítási napon hat tanítási órán, a többi négy szintén egy napon, négy tanítási órán valósult meg. A tizenegyedik tervezet egy foglalkozás tervezete, hiszen a szakképzésről szóló 2019. évi LXXX. törvény értelmében a szakképzésben nem tanórákat, hanem foglalkozásokat tartunk.

Az egész életen át tartó tanulással Life Long Learning (LLL) folyamatosan képezzük magunkat, korszerűsítsük és fejlesszük pedagógiai és szakmai tudásunkat, így személyes példamutatással tudjuk alakítani a tanulók értékrendjét, pozitív tulajdonságait, a tanuláshoz és társaikhoz, valamint a későbbiekben a munkához és a társadalomhoz való hozzáállásukat.

Kovács Henriette
szakmai és köznevelési szakértő

Budapest, 2022.

ÓRATERVEZET - MIÉRT HASZNOSAK A TÖMBÖK?

A pedagógus neve: Kovács Henriette

A szakmacsoport száma és megnevezése: 7. Informatika

A szakképesítés azonosító száma: 54 213 05

A szakképesítés megnevezése: Szoftverfejlesztő

Iskolai rendszerű szakképzés (a szakképzési évfolyamok száma: 2)

Szakmai követelménymodul: 10817-12 Hálózatok, programozás és adatbázis-kezelés

Tantárgy: Adatbázis- és szoftverfejlesztés gyakorlat

Programozási nyelv: Java

Fejlesztői környezet: NetBeans IDE 7.3

Osztály - csoport: I/13. S - SZ2

Az óra tananyaga: Miért hasznosak a tömbök?

Az óra cél- és feladatrendszere

A fejlesztendő attitűd, készségek, képességek, a tanítandó ismeretek (fogalmak, szabályok stb.) és az elérendő fejlesztési szint, tudásszint megnevezése:

- A tanuló tartsa be a számítógépes munka szabályait, különös tekintettel a balesetek megelőzésére. A berendezésekkel fegyelmezetten, a használati utasításokat pontosan követve dolgozzon. Legyen tisztában a számítógépes környezet alapvető ergonómiai kérdéseivel, az egészségvédelem lehetőségeivel számítógépes munkakörnyezetben.
- Felkelteni és folyamatosan ébren tartani a tanulók érdeklődését a szoftverfejlesztés iránt; *rávezetni a tanulókat a tömb adatszerkezet szükségességére, megismertetni és létrehozni ezt a gyors és hatékony listát, a tömböt*, mindezzel hozzájárulva a tanulók logikus gondolkodásának, problémamegoldásának, programozói szemléletének kialakításához. Kialakítani a tanulókban az informatikai ismereteik folyamatos bővítésének és megújításának igényét.
- Korszerű alkalmazói készség kialakítása: meglévő, - Internet segítségével - új alkalmazások kiválasztása a feladat tervezéséhez (Microsoft Office Excel, Structorizer), legyenek képesek arra, hogy mindezeket célszerűen használják.
- Az algoritmikus gondolkodás fejlesztése: a tanulók önálló, rendszerezett, logikus gondolkodásának fejlesztése; a logikus gondolkodás lényeges a problémamegoldásban, és az algoritmusok elkészítésében, kódolásában.
- Legyen képes a NetBeans fejlesztői környezet kényelmi funkcióinak készségszintű használatára.
- Önálló munkára nevelés és felzárkóztatás: a NetBeans fejlesztői környezet, a felhasználó tevékenységére azonnal reagál, így lehetőséget teremt az egyéni ütemű tanulásra, a pontos, kitartó, fegyelmezett munkára, a lemaradókkal való különleges foglalkozásra.
- Alkotó munkára nevelés: a végeredmény egy új termék, egy program lesz.
- Az esztétikai készség fejlesztése: igény és készség a számítógépes produktum esztétikus formájának kialakítására, a program helyes strukturálására.
- *Tudjon tömb típusú változót deklarálni, tömböt létrehozni, és elemeit kiíratni.*
- *Tudjon tömb elemeit tabulátorokkal elválasztva egymás mellett megjeleníteni.*
- *Legyen képes a tömbök elemeit elérni.*

Fejlesztendő készségek, képességek

- Absztrakt gondolkodás, algoritmikus gondolkodás, alkalmazásválasztás, alkalmazói képesség, alkotó képesség, diagnosztizálás,

döntésképesség, eszközválasztás, hallott szakmai szöveg értése, hibakeresés, logikus gondolkodás, megfigyelés, meggyőzőképesség, olvasott szakmai szöveg megértése angol nyelven, önállóság, önellenőrzés, precizitás, pontosság, problémamegoldás, programkód írása, programválasztás, struktogram elemzése, struktogram készítése, szakmai nyelvű beszédkészség, szintaktika követése, visszaemlékezés.

Fejlesztendő attitűd

- Legyen képes: számítógépet kezelni, szoftvereket használni.
- Tudja önállóan használni a hálózatot és annak alapszolgáltatásait. Tudjon adatokat megkeresni, elérni a hálózati szolgáltatások alkalmazásával.
- Legyen képes az adott probléma megoldásához kiválasztani az általa ismert módszerek, eszközök és alkalmazások közül a megfelelőt.
- Ismerje fel és legyen képes különféle formákban megfogalmazni a környezetében előforduló tevékenységek algoritmizálható részleteit.
- Legyen képes elemi és összetett adatok megkülönböztetésére, kezelésére, használatára.
- Adott feladat megoldásához legyen képes algoritmusokat megfogalmazni, tervezni és megvalósítani.
- Legyen képes alkalmazást (szoftvert) futtatni, tesztelni és magyarázó megjegyzésekkel ellátni, előkészítve ezzel a dokumentáció elkészítését.
- Legyen képes programozási feladatot ellátni, (kódolni), Java programnyelven implementálni.
- Legyen képes értelmezni és javítani a NetBeans által megjelenített hibaüzeneteket.
- Legyen képes programjait úgy megírni, hogy azt később Ő maga is, vagy tanulótársa is fel tudja használni.
- Legyen képes eligazodni és tudjon keresni a Java hivatalos dokumentációjában.

Az óra didaktikai feladatai

- A figyelem felkeltése
- A jelenség elemzése
- A megoldás sokoldalú elemzése
- A tanulók informálása a tanulási célról
- A tények elemzése
- Aktivizálás
- Alkalmazás
- Általánosítás
- Az új ismerteket megalapozó tények biztosítása
- Ellenőrzés
- Értékelés
- Figyelem felkeltése
- Fogalomalkotás
- Absztrakció
- Általánosítás
- Fokozatosság
- Motiváció
- Szemléletesség
- Ismétlés
- Visszacsatolás
- Új ismeret közlése
- Tudományosság
- Megerősítés
- Részösszefoglalás
- Rendszerezés
- Információ gyűjtése
- Gyakorlás
- Következtetés
- Problémafelvetés
- Rögzítés

Tantárgyi kapcsolatok: Adatbázis- és szoftverfejlesztés elmélet, Matematika, Angol, Hálózati ismeretek I., Informatikai alapismeretek

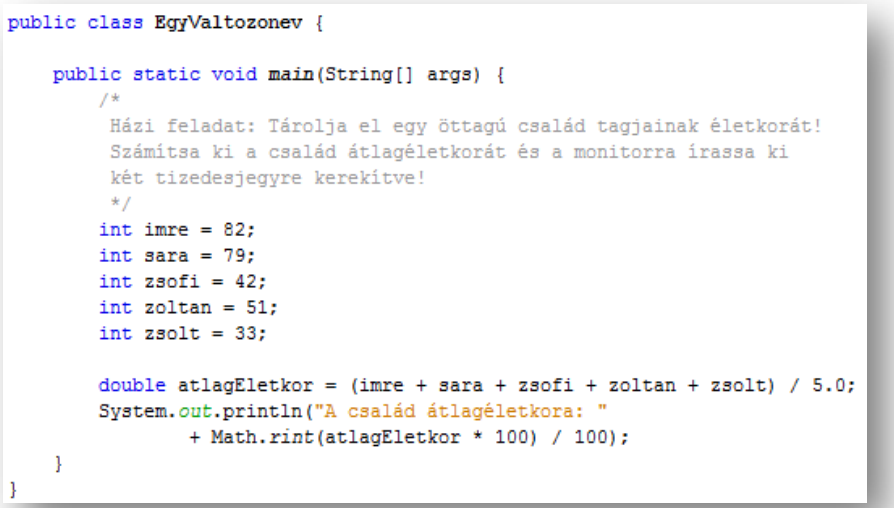
Felhasznált források (tankönyv, munkafüzet, feladat- és szöveggyűjtemény, digitális tananyag, online források, szakirodalom stb.):

Szerző(k)	Cím	Kiadó	ISBN kód
Vég Csaba – dr. Juhász István	Java – start!	Logos 2000	963 03 9005 1
Vég Csaba	Instant Java/Java EE/SOA I-II	Logos 2000	978-963-06-2601-9 978-963-06-2602-6
Dirk Louis – Peter Müller:	Java 5 Belépés a programozás világába	Panem	963 545 454 6
Benkő Tiborné, Tóth Bertalan	Együtt könnyebb a programozás Java	Computerbooks	963 618 337 6
Rogers Cadenhead	Tanuljuk meg a Java programozási nyelvet 24 óra alatt	Kiskapu Kft.	963 9637 07 6
Angster Erzsébet	Objektumorientált tervezés és programozás Java 1 - 2	4KÖR Bt.	963 00 6262 3 963 00 6263 1
Peter J. DePasquale	Java zsebkönyv	Kiskapu Kft.	963 9637 10 6
Nyékyné Gaizler Judit, Frohner Ákos, Kozsik Tamás (Szerkesztők)	Java 2 útikalauz programozóknak	ELTE TTK Hallgatói Alapítvány	963 463 305 6
Kathy Sierra, Bert Bates	Agyhullám: Java	Kiskapu Kft.	978 963 9637 79 5

http://docs.oracle.com/javase/7/docs/api/

Dátum: éééé. hh. nn.

<table>
<tr><th>IDŐKERET</th><th>AZ ÓRA MENETE</th><th>CÉLOK ÉS FELADATOK</th><th>NEVELÉSI-OKTATÁSI STRATÉGIA
MÓDSZEREK
TANULÓI MUNKAFORMÁK
ESZKÖZÖK
MEGJEGYZÉSEK</th></tr>
<tr>
<td>0 – 2’</td>
<td>Számítógép bekapcsolása, házi feladat megnyitása



Képzeljék el, hogy a házi feladatban nem egy öttagú család tagjai, hanem egy 18 lakásos társasház lakói szerepelnek. Mi okozná a nehézséget?
Válasz: A nehézséget az okozná, hogy sok változót kellene deklarálni.</td>
<td>Ellenőrzés
Önellenőrzés
Diagnosztizálás
Hibakeresés

Precizitás, pontosság
Szintaktika követése

A figyelem felkeltése
Motiváció</td>
<td>Számítógép
Házi feladat
A házi feladatok elektronikus úton, határidőre történő beadása; reflektálás részben elektronikusan, részben órán.

Számítógép
Projektor
Vetítővászon
Frontális munka

Megbeszélés (beszélgetés)
NetBeans IDE 7.3 Integrált Fejlesztői Környezet használata</td>
</tr>
</table>

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
2 – 3’	Hogy tudnánk megkönnyíteni a dolgunkat? Jó lenne, ha lenne olyan adatszerkezet, mely egy adott változónévvel több különálló adatot tud tárolni a megadott típusból. Pl.: Képzeljék el, hogy egyforma dobozokat rakunk le egymás mellé és a dobozokba beletesszük a számokat.  Mindegy, hogy egész számokat vagy valós számokat helyezünk el a dobozokban? *Válasz: Nem mindegy, hogy egész számokat vagy valós számokat teszünk a dobozokba.*	Problémamegoldás Szemléletesség Logikus gondolkodás Fokozatosság Absztrakt gondolkodás Az új ismerteket megalapozó tények biztosítása	Szemléltetés Projektor Saját készítésű kép CINEMA 4D programmal Kooperatív oktatási módszer

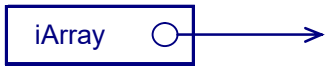
IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
3 – 4’	Mit szeretnénk még elhelyezni benne? *Válasz: Szeretnénk még elhelyezni benne pl. szöveget, logikai értéket.* Ezen igényeink megvalósítására Java-ban a tömb alkalmas.	Absztrakt gondolkodás Új ismeret közlése	Frontális munka Elbeszélés

4 – 7’	Tömb: Fix elemszámú, azonos típusú elemekből álló összetett adatszerkezet.
	Használatában az a nagyszerű, hogy nem kell az elemeknek egyedi neveket kitalálni, hanem egy gyűjtőnevet adunk az egész csoportnak.
	Hogyan deklarálunk egy változót? Mondjon egy példát!
	Válasz: Úgy deklarálunk egy változót, hogy megadjuk a típusát és a nevét!
	Pl.: `int a;`
	A tömb referencia típusú változó, melyet szintén deklarálnunk kell.
	1. elemek típusa
	2. tömbképző operátor: szögletes zárójelpár
	3. tömb azonosítója
	ide is tehetjük!
	`<elemtípus>[] <tömbAzonosító>;`
	`<elemtípus> <tömbAzonosító>[];`
	Pl.: `int[] iArray;`
		Fogalomalkotás, absztrakció, általánosítás
		Rögzítés
		Visszaemlékezés
		Visszacsatolás
		Rögzítés
		Alkalmazás
		Fogalomalkotás
		Megfigyelés
		Rögzítés
			Elbeszélés
			Magyarázat
			Megbeszélés
			Tábla
			Projektor
			Tábla

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK

7 – 9’	Pl.: `int[] iArray1;`
	`int iArray2[];`
	Hogyan tudunk egyszerre több változót deklarálni?
	Válasz: Úgy tudunk egyszerre több változó deklarálni, hogy megadom a típust és vesszővel elválasztva a változók neveit.
	
	Feladat: Döntse el, melyik tömb a következők közül?
	`int[] iArray1, iArray2;`
	`int iArray3[], iArray4;`
		Megerősítés
		
		Visszaemlékezés
		Visszacsatolás
		
		Döntésképesség
		Aktivizálás
		Gyakorlás
		Alkalmazás
			Számítógép
			Projektor
			Vetítővászon
			Frontális munka
			Kooperatív oktatási módszer
			
			Önálló munka

9 – 10’	Emlékezzünk vissza, mi történik a memóriában, ha deklarálunk egy primitív típusú változót! *Válasz: A fordító az ’a’ nevet egy meghatározott tárolóterülethez rendeli.* a Nézzük, mi történik a tömböknél! Milyen adatszerkezet a tömb? *Válasz: A tömb összetett adatszerkezet.*	Visszaemlékezés Ismétlés Megerősítés	Frontális munka Megbeszélés Tábla Szemléltetés

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
10 – 12’	Az így deklarált változó képes egy megadott elemtípusú tömbre mutatni. A változóban az objektum memóriabeli címét tároljuk. Deklaráció hatására nem jön létre objektum! Nézzük, mi történik a memóriában, amikor az iArray nevű int típusú elemekből álló tömböt deklaráljuk!  A deklarálással csak a referenciának foglalunk helyet, ezen a ponton még nincs tömb!	Hallott szakmai szöveg értése Új ismeret közlése Tudományosság Szemléletesség Megerősítés	Elbeszélés Szemléltetés Tábla
12 – 13’	A fejlesztői környezetben a Gyakorlat13 nevű projektben hozzon létre egy gyakorlat13 csomagot és ebben egy Gyakorlat13_1 nevű osztályt!	Önállóság Eszközválasztás Aktivizálás	NetBeans IDE 7.3 Egyéni munka
13 – 14’	Feladat: Deklaráljon egy iArray nevű, int típusú elemekből álló tömböt, a kódot lássa el megjegyzéssel, mely tükrözi, hogy mi történik a kódrészletben! Figyeljen a kód strukturáltságára!	Alkotó képesség Programkód írás	Egyéni munka

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK

14 – 15’	A kivetített kódrészlet alapján ellenőrizze munkáját!
	
	`public class Gyakorlat13_0 {`
	`  public static void main(String[] args) {`
	`  // 1. feladat`
	`  // tömb deklarálása`
	`  int[] iArray;`
		Szintaktika követése
		Precizitás, pontosság
		Önellenőrzés
			Szemléltetés
			Projektor

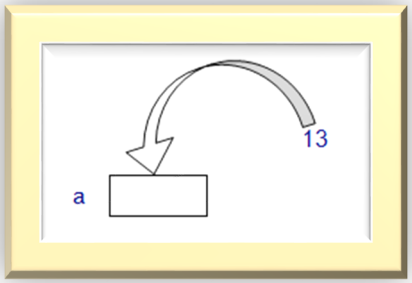
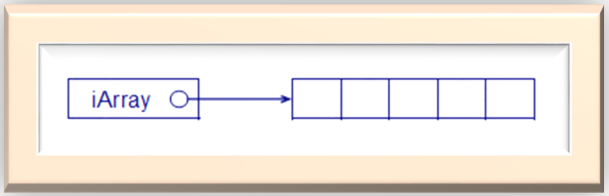
15 – 17’	Melyik operátorral hozunk létre objektumokat? (Gondoljon pl. a billentyűzet-ről való bevitelre!) *Válasz: A new operátorral hozunk létre objektumokat.* A tömböt külön létre kell hoznunk! Mit gondolnak, melyik operátorral tudunk tömböt létrehozni? *Válasz: A new operátorral tudunk tömböt létrehozni.* A tömböt a new operátorral hozhatjuk létre a program futása közben. 1. new operátor 2. elemek típusa 3. tömb mérete (elemek száma): nem negatív, int típusú kifejezés. `new <elemtípus>[<méret>];`	Visszacsatolás Logikus gondolkodás Fokozatosság Alkalmazói képesség Új ismeret közlése	Megbeszélés Elbeszélés Frontális munka Projektor

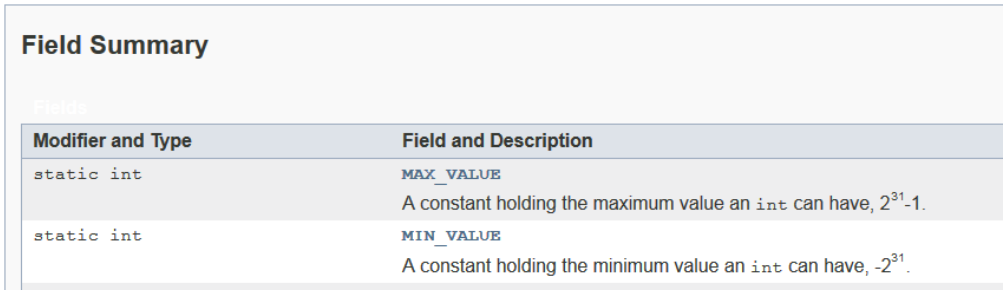
IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
17 – 18’	A tömb méretét mibe kell tenni? *Válasz: A tömb méretét szögletes zárójelbe kell tenni.* Feladat: Hogyan tud létrehozni egy 5 elemű, int típusú tömböt? Fel tudja használni az előbbi deklarálást? Mindenki a fejlesztői környezetben próbálkozzon, ha hibát jelez a NetBeans IDE, próbálja megtalálni a hibát! Ha nem megy, segítek. A kódot lássa el megjegyzéssel, mely tükrözi, hogy mi történik a kódrészletben!	Megfigyelés Alkalmazás Hibakeresés	Önálló munka Fejlesztői környezet

18 – 19’	A kivetített kódrészlet alapján ellenőrizze munkáját!
	`// tömb létrehozása`
	`iArray = new int[5];`
		Szintaktika követése
		Precizitás, pontosság
		Önellenőrzés
			Szemléltetés
			Projektor
			NetBeans IDE 7.3

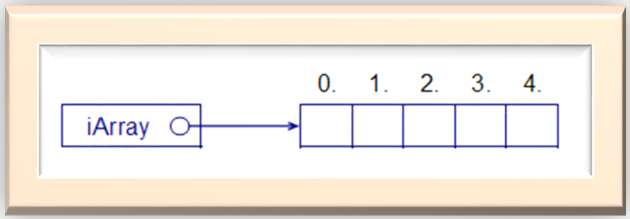
19 – 20’	Nézzük meg, mi történik a memóriában az értékadás során, primitív és összetett típusú változó esetén!
	`int a = 13;`
	`int[] iArray = new int[5];`
		A tanulók informálása a tanulási célról
		Visszacsatolás
			Tábla

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK

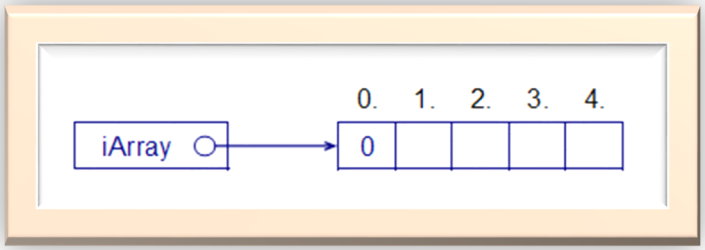
20 – 22’	Próbálja megfogalmazni a különbséget!
	`int a = 13;`
	
	
	Láthatjuk, hogy az érték (13) közvetlenül a változó tárolóterületére íródik be.
	`int[] iArray = new int[5];`
	
	
	A tömb adatelemei külön tárolóterületen helyezkednek el.
		Szakmai nyelvű beszédkészség
		Aktivizálás
		Szemléletesség
		Visszacsatolás
		
		Az új ismereteket megalapozó tények biztosítása
		A tények elemzése
		
		Általánosítás
		
		Tudományosság
		
		Rendszerezés
			Megbeszélés
			Kooperatív oktatási módszer
			
			Tábla
			
			Frontális munka

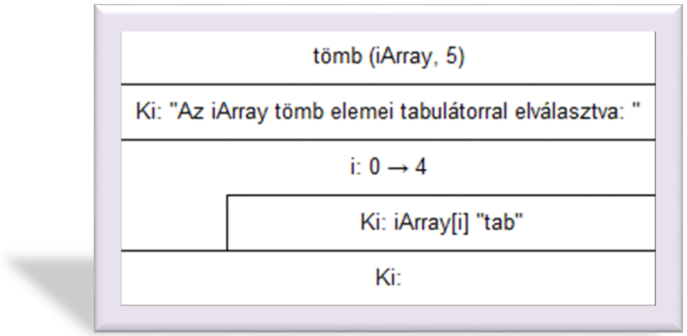
IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
22 – 25’	Hány eleme lehet egy tömbnek? Maximum `Integer.MAX_VALUE` eleme lehet a tömbnek. Nézzünk utána a Java hivatalos dokumentációjában, hogy mit is jelent ez? Hol találom meg a dokumentációt? *Válasz: A dokumentációt a böngészőben találom meg.* Mit írjunk be? *Válasz: Integer Java szavakat írjuk be.* Melyik oldalt keresem? *Válasz: Azt az oldalt keresem, amelyben bent van a docs, oracle szavak és a 7.* http://docs.oracle.com/javase/7/docs/api/java/lang/Integer.html  **Field Summary** Fields Modifier and Type / Field and Description static int — MAX_VALUE — A constant holding the maximum value an int can have, $2^{31}-1$. static int — MIN_VALUE — A constant holding the minimum value an int can have, -2^{31}.	Figyelem felkeltése Programválasztás Információ gyűjtése Olvasott szakmai szöveg megértése angol nyelven	Irányított, önálló munka Számítógép Internet Kooperatív oktatási módszer http://docs.oracle.com/javase/7/docs/api/

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
25 – 26’	Mivel tudnám kiszámolni ezt az értéket? *Válasz: Számítógép számológépével.*  	Alkalmazásválasztás	Irányított, önálló munka Számológép
26 – 28	Tehát: A tömb elemei külön tárolóterületen helyezkednek el. Láthattuk, hogy milliárdos nagyságrendű lehet a tömb elemeinek a száma. Vajon vannak-e a létrehozott iArray tömbnek értékei? Mi nem töltöttük fel értékekkel, a kérdés tehát az, hogy vesznek-e fel alapértelmezett értékeket? Hogy tudnánk meggyőződni róla? *Válasz: Ha látjuk, hogy vannak benne értékek.* Hogyan láthatjuk? *Válasz: Ha ki tudjuk íratni az értékeket, akkor biztosak lehetünk benne, hogy vannak benne értékek.*	Megerősítés Részösszefoglalás Logikus gondolkodás A figyelem felkeltése Problémafelvetés	Elbeszélés Megbeszélés

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
28 – 31’	Ahhoz, hogy ki tudjuk íratni a tömb elemeit, el kell őket érni. A tömb egyes elemeire indexekkel hivatkozunk.  Az elemek tehát indexelhetők, de vigyázzunk, mert az index minden esetben 0-ról induló egész szám. A tömbképző operátor nemcsak a tömb létrehozásakor, hanem a tömb egyes elemeinek elérésekor is használatos. Mit gondolnak, hogy hivatkozunk az iArray tömb első elemére? Ügyeljen az indexre! *Válasz: iArray[0]* Feladat: Írassa ki az iArray tömb első elemét a fejlesztői környezetben! A kódot lássa el megjegyzéssel, mely tükrözi, hogy mi történik a kódrészletben! Ha hibát jelez a NetBeans IDE, próbálja megtalálni a hibát!	Az új ismereteket megalapozó tények biztosítása Fogalomalkotás Szemléletesség Új ismeret közlése Figyelem felkeltése Fokozatosság Aktivizálás Figyelem felkeltése Alkalmazás Precizitás, pontosság Diagnosztizálás	Elbeszélés Szemléltetés Frontális munka Tábla Fejlesztői környezet Önálló munka

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK

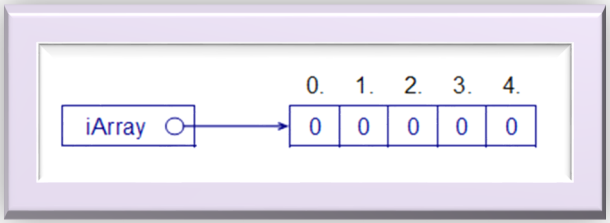
31 – 32’	A kivetített kódrészlet alapján ellenőrizze munkáját! Mit tapasztal a futtatás során?
	
	`// tömb első elemének kiíratása`
	`System.out.println("Az iArray tömb első eleme: " + iArray[0]);`
	
	Válasz: A tömb első eleme alapértelmezett értéket vett fel, ez a nulla.
	
	
	
	Mire következtetnek ebből?
	
	Válasz: A tömb többi eleme is az alapértelmezés szerinti nulla értéket veszi fel.
		Szintaktika követése
		Precizitás, pontosság
		Önellenőrzés
		
		Megfigyelés
		
		Információt gyűjt
		A jelenség elemzése
		
		Következtetés
		
		Logikus gondolkodás
			Szemléltetés
			Projektor
			Önálló munka
			
			Megbeszélés
			
			Tábla
			
			Kooperatív oktatási módszer

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
32 – 33’	Győződjünk meg róla, hogy sejtésünk igaz-e! Hány elemet kellene még kiíratnunk? *Válasz: Még négy elemet kellene kiíratnunk.* Hogyan tudnánk ezt az ismétlő tevékenységet egyszerűbben lekódolni? Gondoljon bele, ha 100 elemű lenne a tömb! *Válasz: Egyszerűbben iterációval, méghozzá, mivel ismerjük az ismétlések számát, ezért léptetéses ciklussal tudjuk lekódolni.*	Megfigyelés Figyelem felkeltése Visszacsatolás Ismétlés Fokozatosság Problémamegoldás	Kooperatív oktatási módszer Frontális munka
33 – 36’	Feladat: Írassa ki a tömb elemeit egymás mellé, tabulátorral elválasztva, használva a már megismert léptetéses ciklust! Készítsük el a kiíratás tervét, válasszuk ki a hozzá megfelelő alkalmazást! 	Eszközválasztás Alkalmazás Algoritmikus gondolkodás Struktogram készítése Precizitás, pontosság Alkotó képesség Struktogram készítése vagy Struktogram elemzése	Structorizer vagy Microsoft Office Excel Irányított, önálló munka Ha kevés az idő a struktogramot megkapják hálózaton, elemezzük, és az alapján kódolnak. Beszélgetés Szemléltetés Projektor

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
36 – 40’	A terv alapján kódoljon! Figyelje meg a futási eredményt!	Programkód írása Szintaktika követése Hibakeresés	Számítógép Fejlesztői környezet Egyéni munka
40 – 41’	A kivetített kódrészlet alapján ellenőrizze munkáját! Vesse össze a kétféle megoldást, nézze meg, mi a különbség a kettő között!	Szemléletesség	Szemléltetés Projektor
41 – 42’	Melyik megoldást célszerű alkalmazni?	Önellenőrzés Megfigyelés A megoldás sokoldalú elemzése Meggyőzőképesség	Projektor NetBeans IDE 7.3 Vita

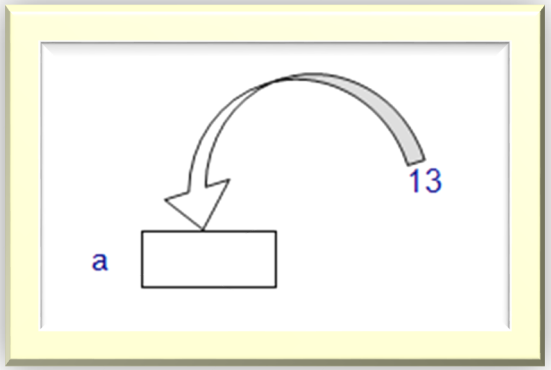
```
// tömb elemeinek kiíratása I.a)
System.out.println("A tömb elemei: ");
for (int i = 0; i <= 4; i++) {
    System.out.print(iArray[i] + "\t");
}
System.out.println();

// tömb elemeinek kiíratása I.b)
System.out.println("A tömb elemei: ");
for (int i = 0; i < 5; i++) {
    System.out.print(iArray[i] + "\t");
}
System.out.println();
```

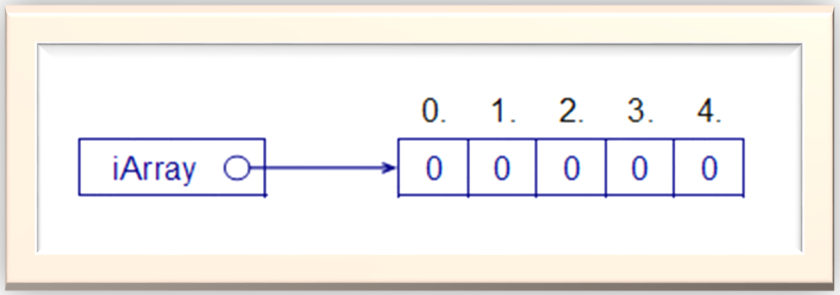
IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
	I.b) A tömb tulajdonságait figyelembe véve az 5 a tömb elemeinek a száma. Egy karakterrel kevesebbet kell gépelni ☺	Döntésképesség	A következő órán a length bevezetése után még jobban megerősítést kap és tudatosul választásunk oka.
42 – 43’	Mit állapíthatunk meg tehát a futási eredmény alapján? Az újonnan létrehozott tömb elemei alapértelmezés szerinti értékeket vesznek fel. 	Általánosítás	Szemléltetés Tábla A KÖVETKEZŐ ÓRA MEGTARTÁSÁRA KÖZVETLENÜL EZ UTÁN AZ ÓRA UTÁN KERÜL SOR, EZÉRT NINCS HÁZI FELADAT.
43 – 44’	Rendszerezés	Rendszerezés	Kooperatív oktatási módszer
44 – 45’	Tanulók munkájának értékelése	Értékelés	Megbeszélés

TÁBLAKÉP

```
int a = 13;
```




```
int[] iArray = new int[5];
```



Feladatok

1. feladat:

   Döntse el, melyik tömb a következők közül?

   ```
   int[] iArray1, iArray2;
   int iArray3[], iArray4;
   ```

2. feladat:

   Deklaráljon egy iArray nevű, int típusú elemekből álló tömböt, a kódot lássa el megjegyzéssel, mely tükrözi, hogy mi történik a kódrészletben!

   Figyeljen a kód strukturáltságára!

3. feladat:

   Hogyan tud létrehozni egy 5 elemű, int típusú tömböt? Fel tudja használni az előbbi deklarálást? Mindenki a fejlesztői környezetben próbálkozzon, ha hibát jelez a NetBeans IDE, próbálja megtalálni a hibát! A kódot lássa el megjegyzéssel, mely tükrözi, hogy mi történik a kódrészletben!

4. feladat:

   Írassa ki az iArray tömb első elemét a fejlesztői környezetben! A kódot lássa el megjegyzéssel, mely tükrözi, hogy mi történik a kódrészletben! Ha hibát jelez a NetBeans IDE, próbálja megtalálni a hibát!

5. feladat:

   Írassa ki a tömb elemeit egymás mellé, tabulátorral elválasztva, használva a már megismert léptetéses ciklust! Készítse el a kiíratás tervét, válassza ki a hozzá megfelelő alkalmazást! A terv alapján kódoljon! Figyelje meg a futási eredményt!

Reflexió

Új ismereteket feldolgozó óra.

A házi feladat nagyon jól előkészítette az új anyag bevezetését, a tömb adatszerkezet fogalmának kialakítását, és a figyelem felkeltését. Nagymértékben elősegítette a motiválást a CINEMA 4D programmal készített modell, melynek még az elkészítése felől is kérdezősködtek a diákok. Sikerült felkelteni érdeklődésüket új programok iránt, illetve így személyes példát mutattam az egész életen át tartó tanulás jelentőségére.

Nehéz elképzelni a tömb memóriafoglalását, ennek megértését ábrákkal tettem szemléletessé, de természetesen ennek rögzülése hosszú folyamat. Nagy hangsúlyt fektettem az elmondottak ismétlésére, mindezzel hozzájárulva az új fogalmak megerősítéséhez és rögzítéséhez.

Fontosnak tartom a dokumentáció folyamatos tanulmányozását, használatát és így egy tanári közlés helyett a diákok saját maguk tapasztalhatják meg a tényeket, ezzel elősegítve az olvasott angol nyelvű szakmai szöveg megértését.

A tömb kiíratását nem pusztán feladatként jelöltem meg, hanem ezáltal saját maguk észlelhették a diákok, hogy milyen alapértelmezett értékeket vesznek fel egy int típusú tömb elemei. Az észrevételüket folyamatosan rögzítettem a táblán.

A struktogramot sikerült elkészíteni, de lehet, hogy az óra végi összefoglalás így nem volt olyan teljeskörű, de ez nem probléma, mert a következő órát, ami közvetlenül ez után jött, ezzel kezdtem.

Többféle megoldási alternatívákra hívtam fel kódolásnál a figyelmüket, valamint arra, érdemes mindig átgondolni, hogy melyik a hatékonyabb megoldás. Az, hogy miért is esett a választás a b) megoldásra, azt a következő órán értették meg igazán a length konstans bevezetésével.

A folyamatos visszacsatolás és a fokozatosság elvének betartását követve a diákok szinte saját maguk jöttek rá és mondták ki az új ismereteket, így elmondhatom, hogy a logikus gondolkodás fejlesztése végig kísérte az órát. Természetesen az új ismereteket a tudományosság elvét követve mindig újra megfogalmaztam, mindezzel hozzájárulva a hallott szakmai szöveg megértésének fejlesztéséhez, a szakmai nyelvű beszédkészség kialakításához és az anyag megerősítéséhez

ÓRATERVEZET - A STRING OSZTÁLY ÉS A TÖMBÖK. TÚLINDEXELÉS.

A pedagógus neve: Kovács Henriette

A szakmacsoport száma és megnevezése: 7. Informatika

A szakképesítés azonosító száma: 54 213 05

A szakképesítés megnevezése: Szoftverfejlesztő

Iskolai rendszerű szakképzés (a szakképzési évfolyamok száma: 2)

Szakmai követelménymodul: 10817-12 Hálózatok, programozás és adatbázis-kezelés

Tantárgy: Adatbázis- és szoftverfejlesztés gyakorlat

Programozási nyelv: Java

Fejlesztői környezet: NetBeans IDE 7.3

Osztály - csoport: I/13. S - SZ2

Az óra tananyaga: A String osztály és a tömbök. Túlindexelés.

Az óra cél- és feladatrendszere

A fejlesztendő attitűd, készségek, képességek, a tanítandó ismeretek (fogalmak, szabályok stb.) és az elérendő fejlesztési szint, tudásszint megnevezése:

- A tanuló tartsa be a számítógépes munka szabályait, különös tekintettel a balesetek megelőzésére. A berendezésekkel fegyelmezetten, a használati utasításokat pontosan követve dolgozzon. Legyen tisztában a számítógépes környezet alapvető ergonómiai kérdéseivel, az egészségvédelem lehetőségeivel számítógépes munkakörnyezetben.
- Felkelteni és folyamatosan ébren tartani a tanulók érdeklődését a szoftverfejlesztés iránt.
- Korszerű alkalmazói készség kialakítása: legyenek képesek arra, hogy a szoftvereket célszerűen használják.
- Az algoritmikus gondolkodás fejlesztése: a tanulók önálló, rendszerezett, logikus gondolkodásának fejlesztése; a logikus gondolkodás lényeges a problémamegoldásban, és az algoritmusok elkészítésében, kódolásában.
- Legyen képes a NetBeans fejlesztői környezet kényelmi funkcióinak készségszintű használatára.
- Önálló munkára nevelés és felzárkóztatás.
- Alkotó munkára nevelés: a végeredmény egy új termék, egy program lesz.
- Az esztétikai készség fejlesztése: igény és készség a számítógépes produktum esztétikus formájának kialakítására, a program helyes strukturálására.
- *Tudjon tömb típusú változót deklarálni, létrehozni.*
- *Legyen képes a tömbök elemeit elérni.*
- *Tudjon tömböt feltölteni, tömb elemeit megjeleníteni.*
- *Lássa a hasonlóságot String és tömb indexelésében.*
- *Tudja mi a különbség a length() metódus és a length konstans között.*
- *Legyen képes felismerni, diagnosztizálni a túlindexelés jelenségét, ezt követően legyen képes azt javítani.*

Fejlesztendő készségek, képességek

- Alkalmazás, feladatértelmezés, fogalomalkotás, kommunikációs képesség, logikus gondolkodás, önellenőrzés, precizitás, pontosság, struktogram készítése, programkód írása, visszaemlékezés, diagnosztizálás, hibakeresés.

Fejlesztendő attitűd

- Legyen képes: számítógépet kezelni, szoftvereket használni.
- Tudja önállóan használni a hálózatot és annak alapszolgáltatásait.
- Legyen képes az adott probléma megoldásához kiválasztani az általa ismert módszerek, eszközök és alkalmazások közül a megfelelőt.
- Legyen képes a problémamegoldáshoz szükséges adatok és az eredmény kapcsolatának megtervezésére, értelmezésére.
- Adott feladat megoldásához legyen képes algoritmusokat megfogalmazni, tervezni és megvalósítani.
- Legyen képes programozási feladatot ellátni, egy szoftvert elkészíteni (kódolni), Java programnyelven implementálni.
- Legyen képes értelmezni és javítani a NetBeans által megjelenített hibaüzeneteket.
- Legyen képes programjait úgy megírni, hogy azt később Ő maga is, vagy tanulótársa is fel tudja használni.

Az óra didaktikai feladatai

- Aktivizálás
- Ismétlés
- Megerősítés
- Fokozatosság
- A tanulók informálása a tanulási célokról
- Szemléletesség
- Visszacsatolás
- Összehasonlítás
- Következtetés
- A tények sokoldalú elemzése
- Új ismeret közlése
- Rögzítés
- Ellenőrzés
- A futási eredmény elemzése
- Gyakorlás

Tantárgyi kapcsolatok: Adatbázis- és szoftverfejlesztés elmélet, Matematika, Hálózati ismeretek I., Informatikai alapismeretek

Felhasznált források (tankönyv, munkafüzet, feladat- és szöveggyűjtemény, digitális tananyag, online források, szakirodalom stb.):

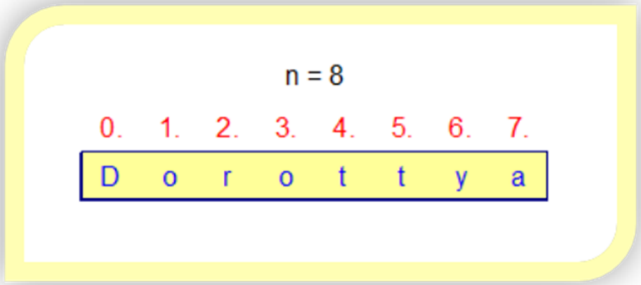
Szerző(k)	Cím	Kiadó	ISBN kód
Vég Csaba – dr. Juhász István	Java – start!	Logos 2000	963 03 9005 1
Vég Csaba	Instant Java/Java EE/SOA I-II	Logos 2000	978-963-06-2601-9 978-963-06-2602-6
Dirk Louis – Peter Müller:	Java 5 Belépés a programozás világába	Panem	963 545 454 6
Benkő Tiborné, Tóth Bertalan	Együtt könnyebb a programozás Java	Computerbooks	963 618 337 6
Rogers Cadenhead	Tanuljuk meg a Java programozási nyelvet 24 óra alatt	Kiskapu Kft.	963 9637 07 6
Angster Erzsébet	Objektumorientált tervezés és programozás Java 1 - 2	4KÖR Bt.	963 00 6262 3 963 00 6263 1
Peter J. DePasquale	Java zsebkönyv	Kiskapu Kft.	963 9637 10 6
Nyékyné Gaizler Judit, Frohner Ákos, Kozsik Tamás (Szerkesztők)	Java 2 útikalauz programozóknak	ELTE TTK Hallgatói Alapítvány	963 463 305 6
Kathy Sierra, Bert Bates	Agyhullám: Java	Kiskapu Kft.	978 963 9637 79 5

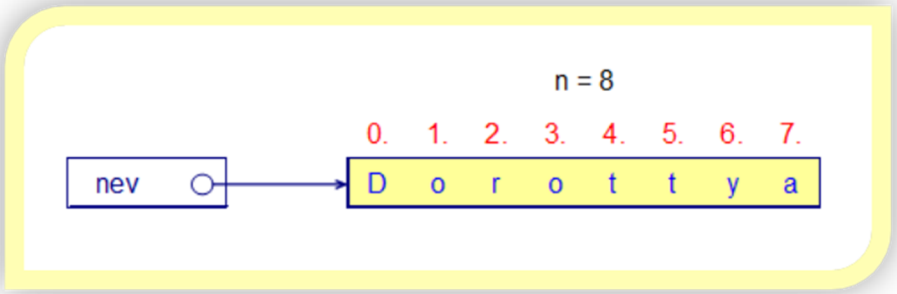
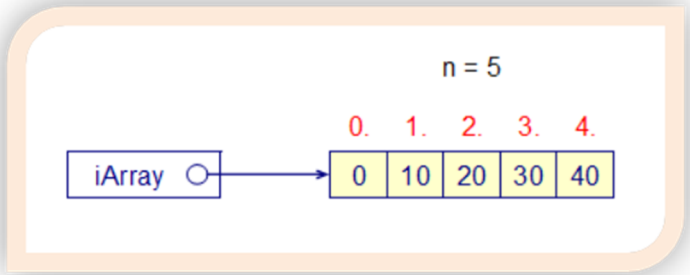
http://docs.oracle.com/javase/7/docs/api/

Dátum: éééé. hh. nn.

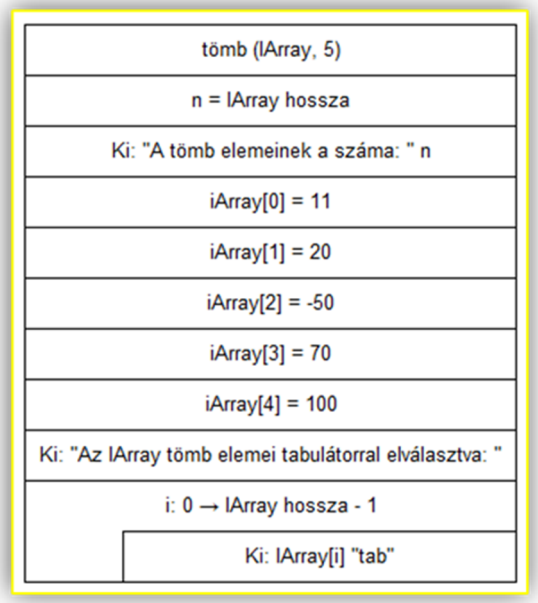
IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MÓDSZEREK TANULÓI MUNKAFORMÁK ESZKÖZÖK MEGJEGYZÉSEK
0 – 5’	Elmúlt órán tanultak felelevenítése: Mi tette szükségessé a tömb adatszerkezet bevezetését? Hogyan deklarálunk egy tömböt? A tömbképző operátort szintaktikailag hol kell elhelyezni? Deklaráció hatására jön létre objektum? Melyik operátorral hozunk létre tömböt? A tömb méretét mibe kell tenni? A memóriában hol helyezkednek el a tömb adatelemei? Hány eleme lehet egy tömbnek? Egy int típusú tömb elemei vesznek-e fel alapértelmezett értékeket? Miket? (Az elkövetkezőkben meg fogjuk vizsgálni azt is, hogy más elemtípusú tömb elemei milyen alapértelmezett értékeket vesznek fel.) Melyik vezérlési szerkezetet használtuk fel a tömb elemeinek kiíratására? Hogyan hivatkozunk a tömb elemeire?	Visszaemlékezés Kommunikációs képesség Aktivizálás Ismétlés Megerősítés Fokozatosság A tanulók informálása a tanulási célokról	Ez az óra közvetlenül kapcsolódik az előző órához. Beszélgetés Frontális munka Projektor

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK

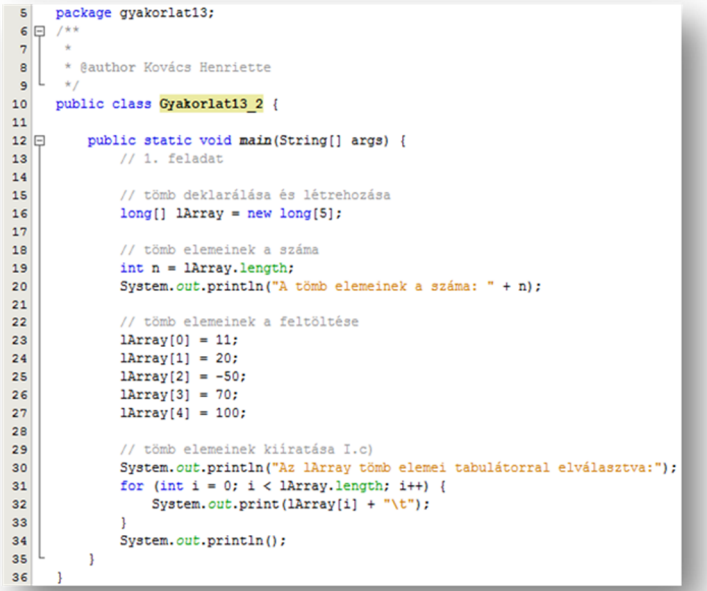
5 – 7’	Hol találkoztunk már az index fogalmával?
	
	A String objektum szövege unicode karakterek sorozata, a karakterek sorszámozottak: ha a szöveg n karaktert tartalmaz, vagyis a szöveg hossza n, akkor a legelső karakter sorszáma, indexe 0, az utolsóé pedig n-1. Tehát hasonlóan, mint a tömböknél.
	
	
	
	Hasonlítsuk össze egy String objektum és egy tömb objektum memóriafoglalását!
	
	Diktálják, milyen kódot írjak az ábrák fölé?
	
	`String nev = new String(”Dorottya”);`
	
	`int[] iArray = new int[5];`
		Visszaemlékezés
		
		Szemléletesség
		Visszacsatolás
		Megerősítés
		
		Logikus gondolkodás
		
		Összehasonlítás
			Kooperatív oktatási módszer
			Elbeszélés
			Frontális munka
			
			Tábla
			
			A diákok mondják, hogy mit írjak az ábrák fölé, melyeket felrajzolok a táblára.
			
			Tábla

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
7 – 9’	 	Szemléletesség Összehasonlítás	Frontális munka Kooperatív oktatási módszer
9 – 11’	Hogyan tudjuk lekérdezni egy szöveg hosszát? A String osztály length() metódusa visszaadja a szöveg hosszát. A visszatérési érték int típusú. Mit gondolnak vajon le tudom kérdezni a tömb hosszát is?	Visszaemlékezés Visszacsatolás Megerősítés Következtetés	Megbeszélés

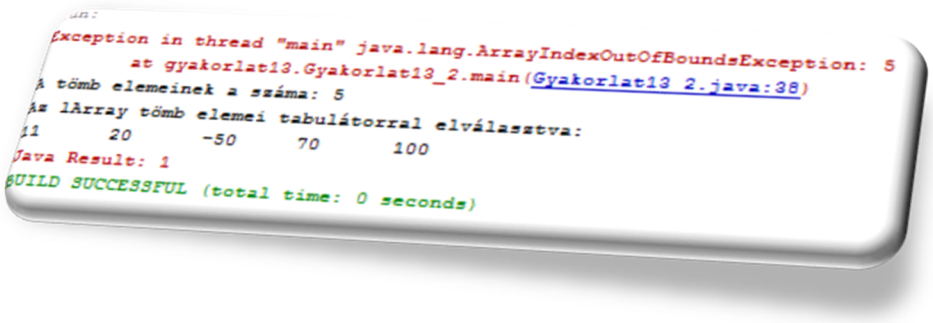
IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
	Minden tömb objektumnak van egy length konstansa, amely megadja a tömb hosszát. `<tömbAzonosító>.length` A length értéke a tömb létrehozásától állandó, nem lehet megváltoztatni. A tömb indexei 0 és length-1 közöttiek.	A tények sokoldalú elemzése → új ismeret közlése Fokozatosság	Elbeszélés
11 – 23’	Feladat: 1) Hozzon létre egy lArray nevű, long elemtípusú, 5 elemű tömböt! Készítsen a feladathoz struktogramot! a) Nézze meg, hogy a length konstans valóban visszaadja-e a tömb hosszát! b) Töltse fel a tömb elemeit a következő számokkal: 11, 20, -50, 70, 100! c) Írassa ki a tömb elemeit a length konstans felhasználásával, tabulátorokkal elválasztva! A feladatot a Gyakorlat13 projekt gyakorlat13 csomagjának, Gyakorlat13_2 néven létrehozott osztályába készítse el!	Feladatértelmezés Alkalmazás Precizitás, pontosság Struktogram készítése Programkód írása Aktivizálás Rögzítés Megerősítés Ellenőrzés Gyakorlás	Egyéni munka Számítógép Structorizer Java 1.5 NetBeans IDE 7.3 Közben segítek annak, aki igényli és követem a munkájukat. Lehetséges, hogy túlindexelési problémába futnak. → Következő feladat Magyarázat

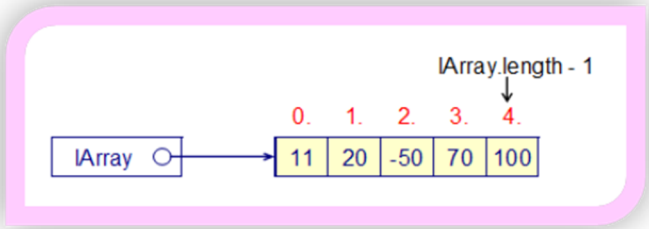
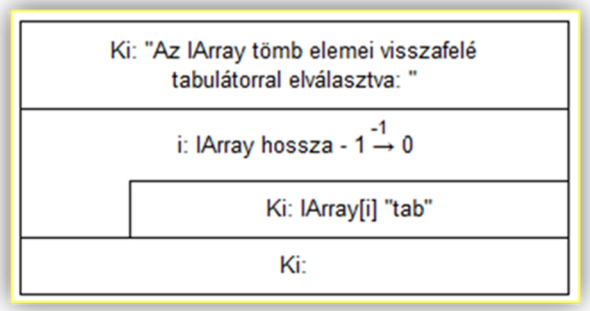
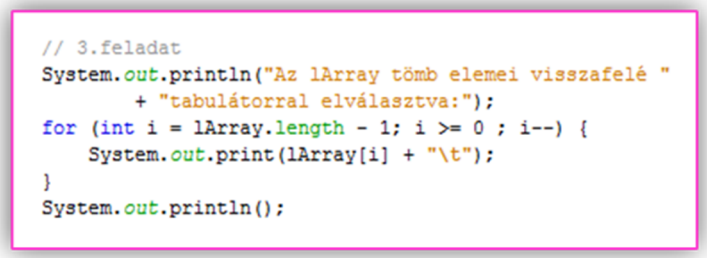
IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
23 – 25’	 A tömb utolsó elemének indexe a tömb hossza-1. A kódolásnál, ha csak a kisebb relációjelet használja, akkor utána a tömb hosszát kell csak beírnia, ellenben ha a kisebb-egyenlő relációjelet használja, akkor utána a tömb hossza-1-et kell beírnia, úgymint a struktogramban.	Önellenőrzés	Microsoft Office Excel 2007 Projektor Magyarázat

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK

25 – 27’	
	``
	package gyakorlat13;
	/**
	 *
	 * @author Kovács Henriette
	 */
	public class Gyakorlat13_2 {
	
	    public static void main(String[] args) {
	        // 1. feladat
	
	        // tömb deklarálása és létrehozása
	        long[] lArray = new long[5];
	
	        // tömb elemeinek a száma
	        int n = lArray.length;
	        System.out.println("A tömb elemeinek a száma: " + n);
	
	        // tömb elemeinek a feltöltése
	        lArray[0] = 11;
	        lArray[1] = 20;
	        lArray[2] = -50;
	        lArray[3] = 70;
	        lArray[4] = 100;
	
	        // tömb elemeinek kiíratása I.c)
	        System.out.println("Az lArray tömb elemei tabulátorral elválasztva:");
	        for (int i = 0; i < lArray.length; i++) {
	            System.out.print(lArray[i] + "\t");
	        }
	        System.out.println();
	    }
	}
	``
	
		Diagnosztizálás
		Hibakeresés
		
		Önellenőrzés
			Projektor

27 – 29’	Feladat: 2) Nézzük meg, mi történik, ha a tömb egy nem létező elemére hivatkozunk! Pl. az lArray tömb 5., nem létező elemét próbáljuk meg feltölteni -13-mal! Figyeljük meg kapunk-e és honnan erről valamilyen hibaüzenetet!	Feladatértelmezés	Egyéni munka Számítógép Java 1.5 NetBeans IDE 7.3

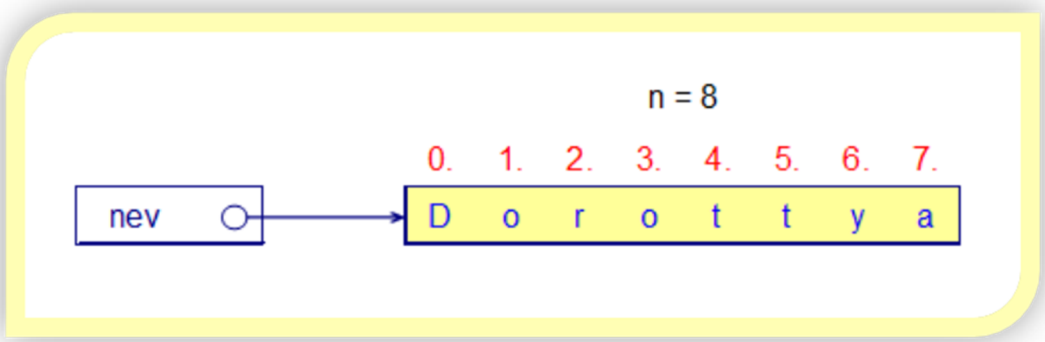
IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
29 – 31’	A NetBeans nem jelez semmilyen problémát! Futtatás során azonban hibába futunk!  A Java a programvégrehajtás alatt figyelmeztetést küld, ha egy hivatkozás egy tömb érvénytelen elemére mutat, és a Java-értelmező, az interpreter megszakítja működését az ArrayIndexOutOfBoundsException hibaüzenettel. Kommentezze ki a sort!	A futási eredmény elemzése Fogalomalkotás	Frontális munka Kooperatív oktatási módszer Magyarázat
31 – 37’	Feladat: 3.) Írassa ki az lArray tömb elemeit fordított sorrendben! Készítsen a kiíratáshoz struktogramot! Ha nem megy, segítségül a hálózaton található egy kép Visszafele néven.	Feladatértelmezés Logikus gondolkodás Alkalmazás Gyakorlás	Egyéni munka

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
37 – 38’		Szemléletesség Rögzítés	Szemléltetés Visszafele.png
38 – 39’		Ellenőrzés Önellenőrzés	Microsoft Office Excel 2007 Projektor
39 – 40’		Ellenőrzés Diagnosztizálás Hibakeresés Önellenőrzés	Projektor

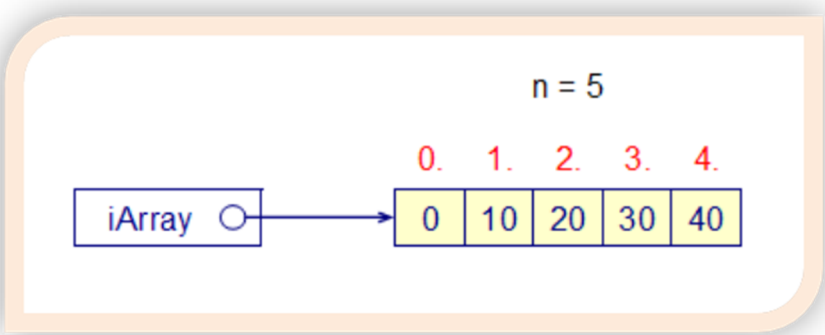
IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
40 – 44’	Rendszerezés: 1. Indexelés 2. Memóriafoglalás 3. length 4. ArrayIndexOutOfBoundsException	Rendszerezés	Kooperatív oktatási módszer A KÖVETKEZŐ ÓRA MEGTARTÁSÁRA KÖZVETLENÜL EZ UTÁN AZ ÓRA UTÁN KERÜL SOR, EZÉRT NINCS HÁZI FELADAT.
44 – 45’	Tanulók munkájának értékelése	Értékelés	Megbeszélés

TÁBLAKÉP

```
String nev = new String(”Dorottya”);
```




```
int[] iArray = new int[5];
```



Feladatok

1.) Hozzon létre egy lArray nevű, long elemtípusú, 5 elemű tömböt! Készítsen a feladathoz struktogramot!
   a) Nézze meg, hogy a length konstans valóban visszaadja-e a tömb hosszát!
   b) Töltse fel a tömb elemeit a következő számokkal: 11, 20, -50, 70, 100!
   c) Írassa ki a tömb elemeit a length konstans felhasználásával, tabulátorral elválasztva!

2.) Nézzük meg, mi történik, ha a tömb egy nem létező elemére hivatkozunk! Pl. az lArray tömb 5., nem létező elemét próbáljuk meg feltölteni -13-mal! Figyeljük meg kapunk-e és honnan erről valamilyen hibaüzenetet!

3.) Írassa ki az lArray tömb elemeit fordított sorrendben! Készítsen a kiíratáshoz struktogramot!
Ha nem megy, segítségül a hálózaton található egy kép Visszafele néven.

Reflexió

Új ismereteket feldolgozó óra

Úgy érzem nagyon hasznos volt az a választás, hogy összehasonlítottam több szempontból is (indexelés, memóriafoglalás, hossz) a tömböket és a String osztály objektumait. Ezzel elősegítettem a Stringről tanultak elmélyítését és a tömb ismeretanyagának rögzítését. A tények sokoldalú elemzésével rávilágítottam a hasonlóságokra és a különbségekre, mely a fogalmak rendszerezését nagyban elősegítette.

A for ciklusban a length konstans használatával a precizitás és pontosság kompetenciákat fejlesztettem, de ennek elmélyítését a továbbiakban fontos feladatomnak tartom, mivel nehézséget okozott az i <= lArray.length -1 és az i < lArray.length azonosságának megértése.

A struktogram elkészítésénél az új jelölések készségszinten természetesen még nem fejlődtek ki, ebben sokat segítettem, sőt a struktogramokat részben közösen készítettük el.

Az ArrayIndexOfBoundsException hibaüzenet megjelenése meglepetésemre nem okozott nehézséget senkinek, sőt a kifejezést még meg is mosolyogták, amikor kimondtam. Örömmel fogadták, hogy ha esetleg ilyen típusú hibát ejtenek, az interpreter jelez nekik. Megértették, hogy azért kell ismerniük az Exception-öket, mert így érteni fogják, hogy mi a hiba oka.

A tömb elemeinek fordított sorrendben történő kiíratása két fontos célt is szolgált, egyrészt a léptetéses ciklus megerősítését, valamint az indexek helyes használatát. Sejtésem beigazolódott, egy diáknak problémát okozott és hasznos volt a hálózatra feltett kép, mely a szemléltetés és vizualitás révén segítséget nyújtott neki.

ÓRATERVEZET - TÖMB ELEMEINEK ALAPÉRTELMEZETT ÉRTÉKEI. TÖMB FELTÖLTÉSE.

A pedagógus neve: Kovács Henriette

A szakmacsoport száma és megnevezése: 7. Informatika

A szakképesítés azonosító száma: 54 213 05

A szakképesítés megnevezése: Szoftverfejlesztő

Iskolai rendszerű szakképzés (a szakképzési évfolyamok száma: 2)

Szakmai követelménymodul: 10817-12 Hálózatok, programozás és adatbázis-kezelés

Tantárgy: Adatbázis- és szoftverfejlesztés gyakorlat

Programozási nyelv: Java

Fejlesztői környezet: NetBeans IDE 7.3

Osztály - csoport: I/13. S - SZ2

Az óra tananyaga: Tömb elemeinek alapértelmezett értékei. Tömb feltöltése.

Az óra cél- és feladatrendszere

A fejlesztendő attitűd, készségek, képességek, a tanítandó ismeretek (fogalmak, szabályok stb.) és az elérendő fejlesztési szint, tudásszint megnevezése:

- A tanuló tartsa be a számítógépes munka szabályait, különös tekintettel a balesetek megelőzésére. A berendezésekkel fegyelmezetten, a használati utasításokat pontosan követve dolgozzon. Legyen tisztában a számítógépes környezet alapvető ergonómiai kérdéseivel, az egészségvédelem lehetőségeivel számítógépes munkakörnyezetben.
- Felkelteni és folyamatosan ébren tartani a tanulók érdeklődését a szoftverfejlesztés iránt.
- Korszerű alkalmazói készség kialakítása: legyenek képesek arra, hogy mindezeket célszerűen használják.
- Az algoritmikus gondolkodás fejlesztése: a tanulók önálló, rendszerezett, logikus gondolkodásának fejlesztése; a logikus gondolkodás lényeges a problémamegoldásban, és az algoritmusok elkészítésében, kódolásában.
- Legyen képes a NetBeans fejlesztői környezet kényelmi funkcióinak készségszintű használatára.
- Önálló munkára nevelés és felzárkóztatás.
- Alkotó munkára nevelés: a végeredmény egy új termék, egy program lesz.
- Az esztétikai készség fejlesztése: igény és készség a számítógépes produktum esztétikus formájának kialakítására, a program helyes strukturálására.
- *Tudjon tömb típusú változót deklarálni, létrehozni.*
- *Legyen képes a tömbök elemeit elérni.*
- *Tudjon tömböt szabály alapján léptetéses ciklussal feltölteni.*
- *Tudjon tömböt megjeleníteni, elemeit szóközzel vagy tabulátorral elkülöníteni.*
- *Tudja primitív és összetett elemtípusú tömbök elemeinek alapértelmezett értékeit meghatározni.*

Fejlesztendő készségek, képességek

– Algoritmus megvalósítása, algoritmus módosítása, diagnosztizálás, hibakeresés, feladatértelmezés, logikus gondolkodás, önállóság, önellenőrzés, precizitás, pontosság, programkód írása, programkód átalakítása, javítása, programkód módosítása, szintaktika követése, visszaemlékezés, kommunikációs képesség, olvasott szakmai szöveg megértése, struktogram átalakítása, struktogram készítése, összefüggések felismerése.

Fejlesztendő attitűd

– Legyen képes: számítógépet kezelni, szoftvereket használni.
– Tudja önállóan használni a hálózatot és annak alapszolgáltatásait.
– Legyen képes az adott probléma megoldásához kiválasztani az általa ismert módszerek, eszközök és alkalmazások közül a megfelelőt.
– Legyen képes elemi és összetett adatok megkülönböztetésére, kezelésére, használatára.
– Legyen képes a problémamegoldáshoz szükséges adatok és az eredmény kapcsolatának megtervezésére, értelmezésére.
– Adott feladat megoldásához legyen képes algoritmusokat megfogalmazni, tervezni és megvalósítani.
– Legyen képes programozási feladatot ellátni, egy szoftvert elkészíteni (kódolni), Java programnyelven implementálni.
– Legyen képes értelmezni és javítani a NetBeans által megjelenített hibaüzeneteket.
– Legyen képes programjait úgy megírni, hogy azt később Ő maga is, vagy tanulótársa is fel tudja használni.

Az óra didaktikai feladatai

- A figyelem felkeltése
- A szükséges előzetes ismeretek felidézése
- A tanulók informálása a tanulási célokról
- A tények elemzése
- Absztrakciók
- Aktivizálás
- Általánosítások
- Az új ismerteket megalapozó tények biztosítása
- Ellenőrzés
- Értékelés
- Fogalomalkotás
- Fokozatosság
- Gyakorlás
- Információ keresése, elemzése.
- Ismétlés
- Következtetés
- Megerősítés
- Rendszerezés
- Új ismeret közlése

Tantárgyi kapcsolatok: Adatbázis- és szoftverfejlesztés elmélet, Matematika, Angol, Hálózati ismeretek I., Informatikai alapismeretek, Biológia

Felhasznált források (tankönyv, munkafüzet, feladat- és szöveggyűjtemény, digitális tananyag, online források, szakirodalom stb.):

Szerző(k)	Cím	Kiadó	ISBN kód
Vég Csaba – dr. Juhász István	Java – start!	Logos 2000	963 03 9005 1
Vég Csaba	Instant Java/Java EE/SOA I-II	Logos 2000	978-963-06-2601-9 978-963-06-2602-6
Dirk Louis – Peter Müller:	Java 5 Belépés a programozás világába	Panem	963 545 454 6
Benkő Tiborné, Tóth Bertalan	Együtt könnyebb a programozás Java	Computerbooks	963 618 337 6
Rogers Cadenhead	Tanuljuk meg a Java programozási nyelvet 24 óra alatt	Kiskapu Kft.	963 9637 07 6
Angster Erzsébet	Objektumorientált tervezés és programozás Java 1 - 2	4KÖR Bt.	963 00 6262 3 963 00 6263 1
Peter J. DePasquale	Java zsebkönyv	Kiskapu Kft.	963 9637 10 6
Nyékyné Gaizler Judit, Frohner Ákos, Kozsik Tamás (Szerkesztők)	Java 2 útikalauz programozóknak	ELTE TTK Hallgatói Alapítvány	963 463 305 6
Kathy Sierra, Bert Bates	Agyhullám: Java	Kiskapu Kft.	978 963 9637 79 5

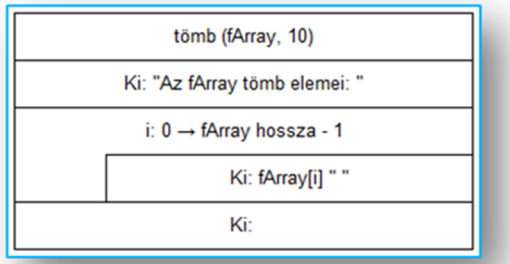
http://docs.oracle.com/javase/7/docs/api/

http://unicode-table.com/en/

Dátum: éééé. hh. nn.

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MÓDSZEREK TANULÓI MUNKAFORMÁK ESZKÖZÖK MEGJEGYZÉSEK
0 – 5’	Tanórát előkészítő kérdések: – A length szó kapcsán mi jut eszébe? – Tömbök esetében hol célszerű használni a length konstanst? – Milyen alapértelmezett értékeket vesznek fel az int, long elemtípusú tömb elemei? – Tudna-e olyan elemtípust mondani, ahol ugyanez az érték jelenne meg? – Milyen primitív típusok vannak még a Java-ban? – A mai órán megtapasztaljuk, hogy valós, logikai, karakter és összetett típus esetén mik az alapértelmezett értékei a tömb elemeinek. – Hogyan töltöttük fel a tömb elemeit? *Válasz: egyesével.* – A mai órán azt is megtanuljuk, hogyan tudnánk sokkal hatékonyabban feltölteni a tömböket. Gyakorlat13/gyakorlat13: Gyakorlat13_3 osztály létrehozása	Visszaemlékezés Kommunikációs képesség Aktivizálás Ismétlés Megerősítés Fokozatosság A tanulók informálása a tanulási célokról	Ez az óra közvetlenül kapcsolódik az előző órához. Beszélgetés Frontális munka Projektor Java 1.5 NetBeans IDE 7.3

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
5 – 8’	A feladat előkészítése: ✓ Felhívom a diákok figyelmét arra, hogy ha a tömb elemei és az indexek között valamilyen összefüggést, szabályt találnak, akkor azt alkalmazva egy léptetéses ciklus segítségével fel tudják tölteni a tömböt. Ennek megértésére feltettem egy képet *Feltoltes_ciklussal* néven a hálózatra. ✓ Átismétlem, hogy mi a literál és milyen típusú az egész és valós literál. ✓ Az ismétlődő kódrészleteket a megfelelő változtatással másolják!	Visszaemlékezés Figyelem felkeltése Aktivizálás Szükséges előzetes ismeretek felidézése	Együttműködő oktatási módszer Szemléltetés Frontálismunka Projektor Hálózat
8 – 9’	Feladat: Minden feladathoz készítsen struktogramot! 1.) Hozzon létre egy fArray nevű, float elemtípusú tömböt, melynek 10 eleme van! a) Írassa ki a tömb elemeinek értékét egymás mellé szóközökkel elválasztva! Milyen következtetéseket von le belőle? Írja be megjegyzésbe! b) Töltse fel a tömb elemeit az alábbi értékekkel! 0.0 2.5 5.0 7.5 10.0 12.5 15.0 17.5 20.0 22.5 c) Írassa ki a tömb elemeit tabulátorral elválasztva! Ha a fejlesztői környezet felhívja figyelmét egy lehetséges pontosságvesztésre, gondoljon arra, hogy egy valós literál milyen típusú!	Feladatértelmezés Logikus gondolkodás Összefüggések felismerése	Projektor Megbeszélés

<table>
<tr><th>IDŐKERET</th><th>AZ ÓRA MENETE</th><th>CÉLOK ÉS FELADATOK</th><th>NEVELÉSI-OKTATÁSI STRATÉGIA
MEGJEGYZÉSEK</th></tr>
<tr>
<td>9 – 13</td>
<td>1.a) feladat tervezése

</td>
<td>Algoritmus tervezése
Struktogram készítése
Visszaemlékezés

Visszacsatolás
Gyakorlás</td>
<td>Egyéni munka

Figyelemmel kísérem a diákok munkáját, ha szükséges, vagy igénylik segítséget nyújtok nekik.

Számítógép
Structorizer</td>
</tr>
<tr>
<td>13 – 17’</td>
<td>1.a) feladat kódolása

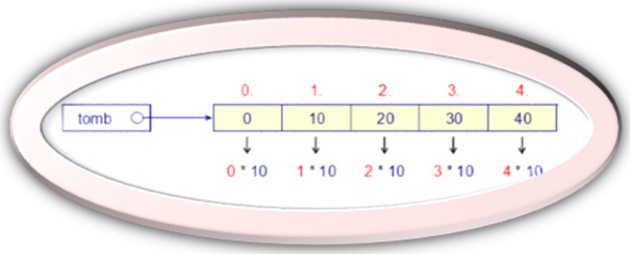
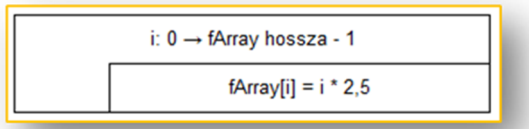
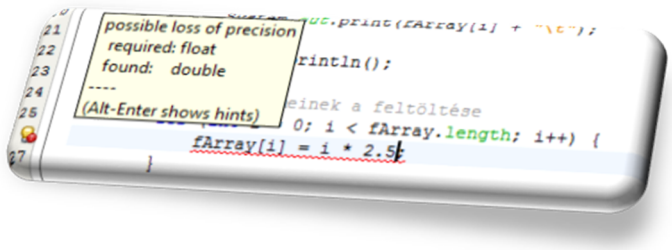
```
5  package gyakorlat13;
6
7  /**
8   *
9   * @author Kovács Henriette
10  */
11 public class Gyakorlat13_3 {
12     public static void main(String[] args) {
13         // 1. feladat
14
15         float[] fArray = new float[10];
16
17         // tömb elemeinek kiíratása
18         System.out.println("Az fArray tömb elemei: ");
19         for (int i = 0; i < fArray.length; i++) {
20             System.out.print(fArray[i] + " ");
21         }
22         System.out.println();
```

A float elemtípusú tömb elemeinek alapértelmezett értéke: 0,0.
Ez a double elemtípusú tömb elemeire is igaz, tehát megállapíthatjuk, hogy a valós elemtípusú tömb elemeinek alapértelmezett értéke: 0,0.</td>
<td>Precizitás, pontosság
Szintaktika követése
Programkód írása
Algoritmus megvalósí-tása

Gyakorlás
Ellenőrzés
Következtetés
Általánosítás
Új ismeret közlése</td>
<td>Egyéni munka

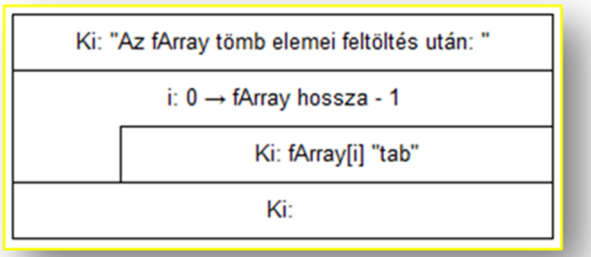
Számítógép
Java 1.5
NetBeans IDE 7.3

Megbeszélés</td>
</tr>
</table>

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
17 – 19’	Feltoltes_ciklussal.png – Szabály felismerése az index és a tömbelem értéke között: 	Szabály felismerése	Feltoltes_ciklussal.png Szemléltetés Megbeszélés
19 – 21’	1.b) feladat tervezése 	Struktogram átalakítása, készítése Önállóság	Egyéni munka Structorizer
21 – 23’	NetBeaans figyelmeztet a lehetséges pontosságvesztésre: possible loss of precision 	Olvasott szakmai angol szöveg értése Diagnosztizálás, hibakeresés	Együttműködő oktatási módszer Projektor

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK

23 – 24’	1.b) feladat kódolása
	`// tömb elemeinek a feltöltése`
	`for (int i = 0; i < fArray.length; i++) {`
	`fArray[i] = i * 2.5F;`
	`}`
		Programkód átalakítása, javítása
		Szintaktika követése
			Egyéni munka
			Számítógép
			Java 1.5
			NetBeans IDE 7.3

24 – 27’	1.c) feladat tervezése 	Algoritmus módosítása	Egyéni munka Structorizer

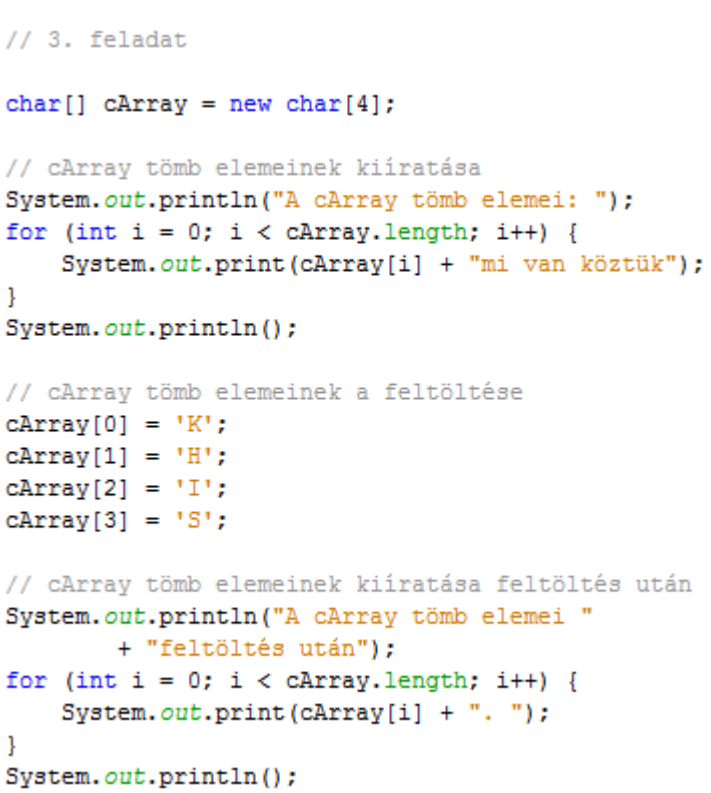
27 – 29’	1.c) feladat kódolása
	`// tömb elemeinek kiíratása feltöltés után`
	`System.out.println("Az fArray tömb elemei "`
	`+ "feltöltés után: ");`
	`for (int i = 0; i < fArray.length; i++) {`
	`System.out.print(fArray[i] + "\t");`
	`}`
	`System.out.println();`
	`}`
	`}`
		Programkód módosítása
		Precizitás, pontosság
		Szintaktika követése
		
		Önellenőrzés
			Egyéni munka
			Számítógép
			Java 1.5
			NetBeans IDE 7.3
			Projektor

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
29 – 30’	A 2., 3., 4. feladat nem véletlenszerűen, tudatosan felosztásra kerül a 3 diák között.	Fokozatosság Precizitás, pontosság	Differenciálás

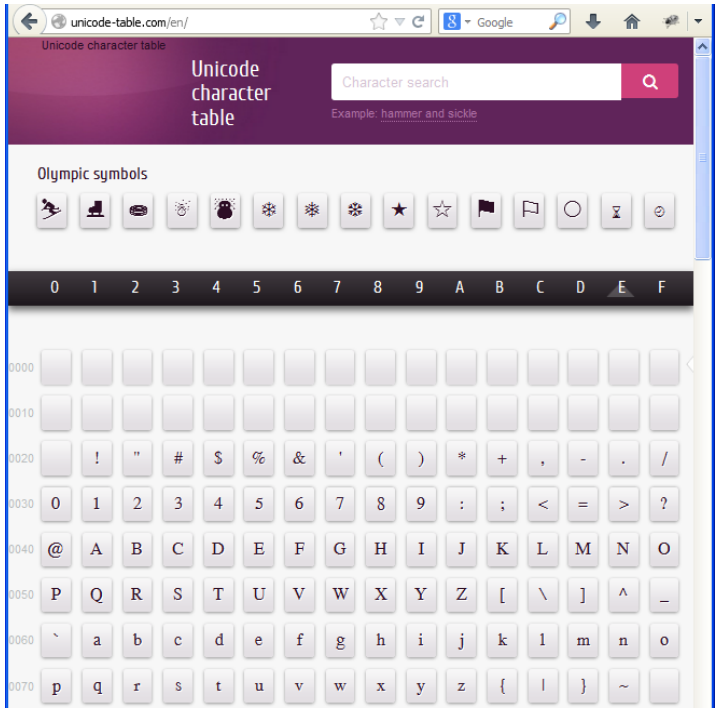
30 – 35’	`// 2. feladat`
	
	`boolean[] bArray = new boolean[4];`
	
	`// bArray tömb elemeinek kiíratása`
	`System.out.println("A bArray tömb elemei: ");`
	`for (int i = 0; i < bArray.length; i++) {`
	`  System.out.print(bArray[i] + "\t");`
	`}`
	`System.out.println();`
	
	`// bArray tömb elemeinek a feltöltése`
	`bArray[0] = true;`
	`bArray[1] = false;`
	`bArray[2] = true;`
	`bArray[3] = false;`
	
	`// bArray tömb elemeinek kiíratása feltöltés után`
	`System.out.println("A bArray tömb elemei "`
	`  + "feltöltés után:");`
	`for (int i = 0; i < bArray.length; i++) {`
	`  System.out.print(bArray[i] + "\t");`
	`}`
	`System.out.println();`
		Programkód írása
		
		Az új ismerteket megalapozó tények biztosítása
		A tények elemzése
		
		Ellenőrzés
			Struktogramot nem kérek – eddigi struktogramok és kódok felhasználása
			
			Egyéni munka
			NetBeans IDE 7.3
			A megoldást a hálózatra a bead mappába rakatom fel.
			Kivetítem
			projektor
			segítségével.

35 – 36’	Egy boolean elemtípusú tömb elemeinek alapértelmezés szerinti értéke: false	Fogalomalkotás, következtetés, absztrakció, általánosítás.	Tanulói elbeszélés

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK

30 – 35’	
	// 3. feladat
	
	char[] cArray = new char[4];
	
	// cArray tömb elemeinek kiíratása
	System.out.println("A cArray tömb elemei: ");
	for (int i = 0; i < cArray.length; i++) {
	System.out.print(cArray[i] + "mi van köztük");
	}
	System.out.println();
	
	// cArray tömb elemeinek a feltöltése
	cArray[0] = 'K';
	cArray[1] = 'H';
	cArray[2] = 'I';
	cArray[3] = 'S';
	
	// cArray tömb elemeinek kiíratása feltöltés után
	System.out.println("A cArray tömb elemei "
	+ "feltöltés után");
	for (int i = 0; i < cArray.length; i++) {
	System.out.print(cArray[i] + ". ");
	}
	System.out.println();
		Programkód írása
		
		Az új ismerteket megalapozó tények biztosítása
		A tények elemzése
		
		Ellenőrzés
			Egyéni munka
			NetBeans IDE 7.3
			A megoldást a hálózatra a bead mappába rakatom fel.
			Kivetítem
			projektor
			segítségével.
			A helyes megoldások felkerülnek a hálózatra, így mindenkinek meglesz a „másik munkája”, melyeket be tudnak másolni a saját kódjukba.

36 – 37’	Egy char elemtípusú tömb elemeinek alapértelmezés szerinti értéke: \u0000 Nézzünk utána a böngészőben, hogy melyik kódú unicode karakter ez. PL.: http://unicode-table.com/en/	Fogalomalkotás, következtetés, absztrakció, általánosítás.	Tanulói elbeszélés Internet

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
37 – 39’	 A Unicode a különböző írásrendszerek egységes kódolását leíró nemzetközi szabvány. A \u0000 a 0-s kódú unicode karakter.	Információ keresése, elemzése Megerősítés Új ismeret közlése	Egyéni munka Megbeszélés Magyarázat

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK

30 – 35’	`// 4. feladat`
	
	`String[] sArray = new String[4];`
	
	`// sArray tömb elemeinek kiíratása`
	`System.out.println("Az sArray tömb elemei: ");`
	`for (int i = 0; i < sArray.length; i++) {`
	`  System.out.print(sArray[i] + "\t");`
	`}`
	`System.out.println();`
	
	`sArray[0] = "alma";`
	`sArray[1] = "körte";`
	`sArray[2] = "szilva";`
	`sArray[3] = "eper";`
	
	`// sArray tömb elemeinek kiíratása feltöltés után`
	`System.out.println("Az sArray tömb elemei "`
	`  + "feltöltés után: ");`
	`for (int i = 0; i < sArray.length; i++) {`
	`  System.out.print(sArray[i] + "\t");`
	`}`
	`System.out.println();`
		Programkód írása
		
		Az új ismerteket megalapozó tények biztosítása
		A tények elemzése
		
		Ellenőrzés
			Egyéni munka
			NetBeans IDE 7.3
			A megoldást a hálózatra a bead mappába rakatom fel.
			Kivetítem
			Projektor
			segítségével.
			Egészséges táplálkozás fontosságának hangsúlyozása.
			Ha egy diák sokkal hamarabb végez, mint a másik, akkor a másik két feladatból választ, hogy melyiket oldja meg.

39 – 41’	Egy String elemtípusú tömb elemeinek alapértelmezés szerinti értéke: null Referenciák esetén ez az érték null. A következő órán megmutatom, mi történik a memóriában, ha a tömb összetett elemtípusú.	Fogalomalkotás, következtetés, absztrakció, általánosítás. Figyelem felkeltése	Tanulói elbeszélés

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
41 – 44’	Rendszerezés: 1. Egyszerű elemtípusú tömbök elemeinek alapértelmezett értékei 2. Összetett elemtípusú tömbök elemeinek alapértelmezett értékei 3. Tömbök feltöltési lehetőségei: kódban egyesével szabály alapján Mit gondolnak, hogy lehetne még feltölteni a tömböket? felhasználó által (a következő órán megnézzük) 4. Tömbök kiíratása szóközzel, tabulátorral elválasztva Hogy lenne a tömbök kiíratása még szebb? vesszővel elválasztva	Rendszerezés Figyelem felkeltése	Együttműködő oktatási módszer A KÖVETKEZŐ ÓRA MEGTARTÁSÁRA KÖZVETLENÜL EZ UTÁN AZ ÓRA UTÁN KERÜL SOR, EZÉRT NINCS HÁZI FELADAT.
44 – 45’	Tanulók munkájának értékelése	Értékelés	Megbeszélés Egész órát végig kíséri az értékelés.

Feladatok

1.) Hozzon létre egy fArray nevű, float elemtípusú tömböt, melynek 10 eleme van! Minden feladathoz készítsen struktogramot!
   a) Írassa ki a tömb elemeinek értékét egymás mellé szóközökkel elválasztva! Milyen következtetéseket von le belőle? Írja be megjegyzésbe!
   b) Töltse fel a tömb elemeit az alábbi értékekkel!
      0.0 2.5 5.0 7.5 10.0 12.5 15.0 17.5 20.0 22.5
   c) Írassa ki a tömb elemeit tabulátorral elválasztva!

   Ha a fejlesztői környezet felhívja figyelmét egy lehetséges pontosságvesztésre, gondoljon arra, hogy egy valós literál milyen típusú!
2.) Hozzon létre egy bArray nevű, boolean elemtípusú tömböt, melynek 4 eleme van!
   a) Írassa ki a tömb elemeinek értékét egymás mellé szóközökkel elválasztva! Milyen következtetéseket von le belőle? Írja be megjegyzésbe!
   b) Töltse fel a tömb elemeit tetszőleges logikai értékekkel!
   c) Írassa ki a tömb elemeit tabulátorral elválasztva!
3.) Hozzon létre egy cArray nevű, char elemtípusú tömböt, melynek 4 eleme van!
   a) Írassa ki a tömb elemeinek értékét egymás mellé szóközökkel elválasztva! Milyen következtetéseket von le belőle? Írja be megjegyzésbe!
   b) Töltse fel a tömb elemeit tetszőleges karakterekkel!
   c) Írassa ki a tömb elemeit tabulátorral elválasztva!
4.) Hozzon létre egy sArray nevű, String elemtípusú tömböt, melynek 4 eleme van!
   a) Írassa ki a tömb elemeinek értékét egymás mellé szóközökkel elválasztva! Milyen következtetéseket von le belőle? Írja be megjegyzésbe!
   b) Töltse fel a tömb elemeit egészséges élelmiszerekkel!
   c) Írassa ki a tömb elemeit tabulátorral elválasztva

Reflexió

Ezen óra elemzése során a feladatsor mentén fogok haladni, mely formailag szinte ugyanazon feladatokat tartalmazza. Ez azonban csak a látszat. Minden feladat hordoz új ismeret megszerzésére irányuló törekvéseket, túlhaladva az informatikán, keresvén a kapcsolatot más tantárgyakkal, ami ilyen elvont, absztrakt ismeret megalapozása során nagyon fontos feladat, de ugyanakkor kihívás.

Beigazolódott, hogy nagyon jó választásnak bizonyult az 1. feladat. Összeállításánál több gondolat is vezérelt. Ezek a következők voltak: át szerettem volna ismételni gyakorlati feladaton keresztül a tömb létrehozását, feltöltését és megjelenítését; meg szerettem volna mutatni, hogy tömb elemeinek mik az alapértelmezett értékei valós elemtípus esetén; illetve azt, hogy egy tömb feltöltése nemcsak szekvenciákkal lehetséges. Először double elemtípusú tömbben gondolkoztam, aztán átírtam a feladatot float elemtípusúra, egyrészt azért, mert így lehetőségem volt átismételni literálok adattípusait, másrészt belefutottak a possible loss of precision üzenetbe. Valóban segítségül szolgált a hálózatra feltett Feltoltes_ciklussal.png kép, főleg az egyik diáknak, aki a matematikával nincs olyan jó barátságban. ☺ A második kiíratásnál - 1.c) feladat - azért változtattam a fehér szóközökön, hogy megerősítsem bennük azt a tényt, ha átvesznek valamilyen kódrészletet, mindig győződjenek meg arról, hogy az megfelel-e a specifikációnak.

A következtetés leírása gondot okozott, a megfogalmazás maga nem volt probléma, de ezt megjegyzésbe beírni nehezükre esett. Vagyis a kommentek írását a továbbiakban is következetesen kérni fogom.

A 2-4. feladat alkalmat nyújtott a differenciálásra és nagyon kézenfekvő volt. Láttam világosan, hogy arra nem lesz idő, hogy megtapasztalják azt, hogy logikai, karakter és összetett elmtípus esetén mik az alapértelmezett értékek és mivel hárman voltak így nehézségi sorrendben kiosztottam a feladatokat. Lehet, hogy megkérdezhettem volna őket, ki melyiket szeretné, és ugyanazt választják, amit én szántam volna nekik. Így még motiváltabbak lettek volna, de ez magában foglalta volna azt a lehetőséget, hogy nem tudták volna feltölteni a tömböt és nem lesz sikerélményük.

A tetszőleges elemekkel való feltöltés szintén nagyon tudatos volt részemről, mert nem igazán kreatívak, így ezt a képességet mindenképpen fejleszteni szerettem volna. Az utolsó feladatban egy kis megkötés azért volt, az egészséges élelmiszerekből lehetett csak választani. Ez irányban nagyon fogékony ez a csoport is, sokat szoktam beszélgetni velük erről és úgy érzem próbálják betartani amiket elmagyarázok nekik ezzel kapcsolatban.

Feladataim a fokozatosság elvét követve jó szolgálatot tettek mind az új ismeretek elsajátításában, mind a gyakorlásban, rögzítésben.

ÓRATERVEZET - TÖMB ELEMEINEK FELTÖLTÉSE BILLENTYŰZETRŐL, KIÍRATÁSA VESSZŐVEL ELVÁLASZTVA.

A pedagógus neve: Kovács Henriette

A szakmacsoport száma és megnevezése: 7. Informatika

A szakképesítés azonosító száma: 54 213 05

A szakképesítés megnevezése: Szoftverfejlesztő

Iskolai rendszerű szakképzés (a szakképzési évfolyamok száma: 2)

Szakmai követelménymodul: 10817-12 Hálózatok, programozás és adatbázis-kezelés

Tantárgy: Adatbázis- és szoftverfejlesztés gyakorlat

Programozási nyelv: Java

Fejlesztői környezet: NetBeans IDE 7.3

Osztály - csoport: I/13. S - SZ2

Az óra tananyaga: Tömb elemeinek feltöltése billentyűzetről, kiíratása vesszővel elválasztva.

Az óra cél- és feladatrendszere

A fejlesztendő attitűd, készségek, képességek, a tanítandó ismeretek (fogalmak, szabályok stb.) és az elérendő fejlesztési szint, tudásszint megnevezése:

- A tanuló tartsa be a számítógépes munka szabályait, különös tekintettel a balesetek megelőzésére. A berendezésekkel fegyelmezetten, a használati utasításokat pontosan követve dolgozzon. Legyen tisztában a számítógépes környezet alapvető ergonómiai kérdéseivel, az egészségvédelem lehetőségeivel számítógépes munkakörnyezetben.
- Felkelteni és folyamatosan ébren tartani a tanulók érdeklődését a szoftverfejlesztés iránt. Kialakítani a tanulókban az informatikai ismereteik folyamatos bővítésének és megújításának igényét.
- Korszerű alkalmazói készség kialakítása: meglévő, - Internet segítségével - új alkalmazások kiválasztása a feladat tervezéséhez.
- Az algoritmikus gondolkodás fejlesztése: a tanulók önálló, rendszerezett, logikus gondolkodásának fejlesztése; a logikus gondolkodás lényeges a problémamegoldásban, és az algoritmusok elkészítésében, kódolásában.
- Legyen képes a NetBeans fejlesztői környezet kényelmi funkcióinak készségszintű használatára.
- Önálló munkára nevelés és felzárkóztatás.
- Alkotó munkára nevelés.
- Az esztétikai készség fejlesztése: igény és készség a számítógépes produktum esztétikus formájának kialakítására, a program helyes strukturálására.
- *Tudjon tömb típusú változót deklarálni, létrehozni.*
- *Legyen képes a tömbök elemeit absztrakt módon elérni.*
- *Tudjon tömböt billentyűzetről feltölteni.*
- *Tudjon tömbelemeket vesszővel elválasztva megjeleníteni.*
- *Legyen képes algoritmust alkotni, megvalósítani, struktogramot készíteni, programkódot írni.*
- *Legyen képes a diagnosztizálásra, hibakeresésre.*

Fejlesztendő készségek, képességek

- Visszaemlékezés, kommunikációs képesség, feladatértelmezés, alkalmazásválasztás, algoritmus alkotása, algoritmus tervezése, algoritmus készítése, struktogram készítése, precizitás, pontosság, önellenőrzés, diagnosztizálás, programkód írása, szintaktika követése, absztrakt gondolkodás, logikus gondolkodás, algoritmikus gondolkodás, rendszerezés, figyelem felkeltése.

Fejlesztendő attitűd

- Legyen képes: számítógépet kezelni, szoftvereket használni.
- Tudja önállóan használni a hálózatot és annak alapszolgáltatásait.

- Legyen képes az adott probléma megoldásához kiválasztani az általa ismert módszerek, eszközök és alkalmazások közül a megfelelőt.
- Ismerje fel és legyen képes különféle formákban megfogalmazni a környezetében előforduló tevékenységek algoritmizálható részleteit.
- Adott feladat megoldásához legyen képes algoritmusokat megfogalmazni, tervezni és megvalósítani.
- Legyen képes programozási feladatot ellátni, egy szoftvert elkészíteni (kódolni), Java programnyelven implementálni.
- Legyen képes értelmezni és javítani a NetBeans által megjelenített hibaüzeneteket.
- Legyen képes programjait úgy megírni, hogy azt később Ő maga is, vagy tanulótársa is fel tudja használni.

Az óra didaktikai feladatai

- Motiválás
- A figyelem felkeltése
- A tanulók informálása a tanulási célokról.
- Aktivizálás
- Ismétlés
- A szükséges előzetes ismeretek felidézése a feladat megoldásához.
- Fokozatosság
- Figyelem felhívása
- Megerősítés
- Feladat kitűzése
- Gyakorlás
- Visszacsatolás
- Általánosítás
- Szemléletesség
- Rögzítés

Tantárgyi kapcsolatok: Adatbázis- és szoftverfejlesztés elmélet, Angol, Hálózati ismeretek I., Informatikai alapismeretek

Felhasznált források (tankönyv, munkafüzet, feladat- és szöveggyűjtemény, digitális tananyag, online források, szakirodalom stb.):

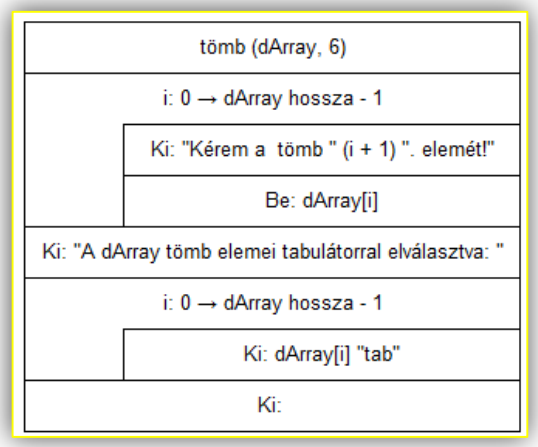
Szerző(k)	Cím	Kiadó	ISBN kód
Vég Csaba – dr. Juhász István	Java – start!	Logos 2000	963 03 9005 1
Vég Csaba	Instant Java/Java EE/SOA I-II	Logos 2000	978-963-06-2601-9 978-963-06-2602-6
Dirk Louis – Peter Müller:	Java 5 Belépés a programozás világába	Panem	963 545 454 6
Benkő Tiborné, Tóth Bertalan	Együtt könnyebb a programozás Java	Computerbooks	963 618 337 6
Rogers Cadenhead	Tanuljuk meg a Java programozási nyelvet 24 óra alatt	Kiskapu Kft.	963 9637 07 6
Angster Erzsébet	Objektumorientált tervezés és programozás Java 1 - 2	4KÖR Bt.	963 00 6262 3 963 00 6263 1
Peter J. DePasquale	Java zsebkönyv	Kiskapu Kft.	963 9637 10 6
Nyékyné Gaizler Judit, Frohner Ákos, Kozsik Tamás (Szerkesztők)	Java 2 útikalauz programozóknak	ELTE TTK Hallgatói Alapítvány	963 463 305 6
Kathy Sierra, Bert Bates	Agyhullám: Java	Kiskapu Kft.	978 963 9637 79 5

http://docs.oracle.com/javase/7/docs/api/

Dátum: éééé. hh. nn.

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MÓDSZEREK TANULÓI MUNKAFORMÁK ESZKÖZÖK MEGJEGYZÉSEK
0 – 4’	Az órát előkészítő kérdések: 1. Hogyan tudunk létrehozni tömböket? new operátorral *meg fogunk ismerkedni még egy lehetőséggel, az úgynevezett inicializáló blokkal* 2. Hogyan tudunk feltölteni tömböket? egyesével, szekvenciával szabály alapján, vezérlési szerkezettel, léptetéses ciklussal *felhasználó által (ezen az órán)* *véletlenszerűen* 3. Hogyan végeztük tömbök kiíratását? szóközzel, tabulátorral elválasztva *vesszővel elválasztva (ezen az órán)* Hozzanak létre a Gyakorlat13 projekt gyakorlat13 csomagjában egy Gyakorlat13_4 osztályt!	Visszaemlékezés Kommunikációs képesség Motiválás A figyelem felkeltése A tanulók informálása a tanulási célokról. Aktivizálás Ismétlés Fokozatosság	Ez az óra közvetlenül kapcsolódik az előző órához. Beszélgetés Frontális munka Projektor NetBeans IDE 7.3 Java 1.5

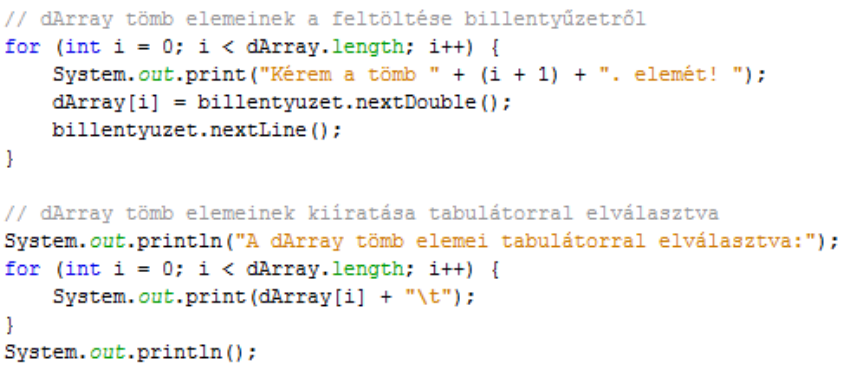
IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
4 – 5’	Feladat kiadása: Olvassa el figyelmesen a feladatot! 1.) A feladathoz készítsen struktogramot! Hozzon létre egy dArray nevű, double elemtípusú tömböt, melynek 6 eleme van! a) Töltse fel a tömböt a felhasználó által megadott értékekkel! b) Írassa ki a tömb elemeinek értékét egymás mellé tabulátorral elválasztva!	Feladatértelmezés	Projektor Hálózat Egyéni munka
5 – 6’	Ne feledkezzen meg kiíratni a felhasználónak, hogy írjon be számot! Azt is közölje mindig a felhasználóval, hogy hányadik elemet kéri be! A felhasználó számára első elem a programban milyen indexű? *Válasz: 0* Hogy hivatkozunk a tömb egy elemére? *Válasz: A tömb neve és szögletes zárójelben az elem indexe.*	Figyelem felhívása Megerősítés	Frontális munka Elbeszélés Megbeszélés
6 – 13’	Készítse el a struktogramot, használja hozzá a legoptimálisabb programot!	Feladat kitűzése Alkalmazásválasztás	Microsoft Office Excel 2007 Structorizer

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
6 – 13’		Algoritmus alkotása Struktogram készítés Precizitás, pontosság Aktivizálás Gyakorlás	Egyéni munka Figyelemmel követem munkájukat.
13 – 15’	Kérem, ellenőrizzék a kivetített struktogram alapján, hogy jól dolgoztak-e! Ha nem tudják eldönteni jó-e amit elkészítettek, szóljanak, segítek. Az esetleges hibákat javítsák!	Önellenőrzés Diagnosztizálás	Projektor
15 – 16’	Kódolás előkészítése: Mit kell tennünk, hogy a felhasználó adatokat tudjon bevinni a programba a billentyűzeten keresztül? *Válasz: Billentyűzetadatok beolvasására a Scanner osztály szolgál. Létre kell hoznunk egy Scanner típusú objektumot. Az osztálynak tudnia kell, hogy honnan jönnek az adatok. Billentyűzet esetén ez a System.in.*	Ismétlés	Megbeszélés

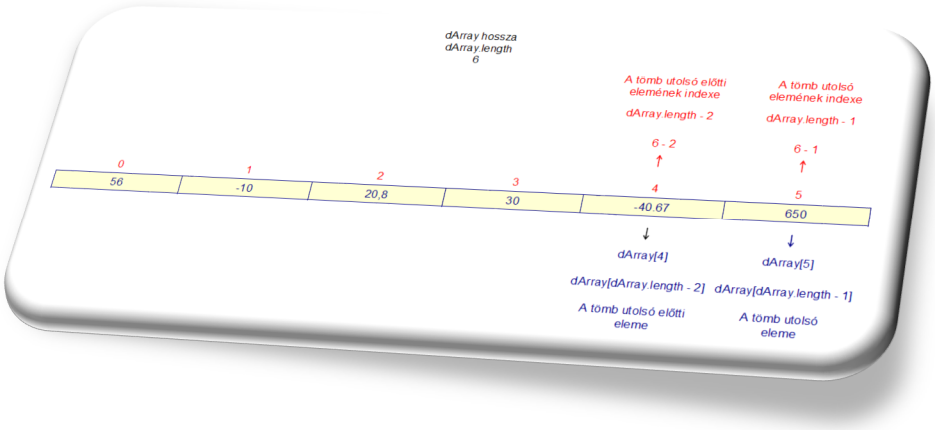
IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
15 – 16’	Mi a Java alapértelmezett csomagja? Melyik csomagban van a Scanner osztály? *Válasz: A lang az alapértelmezett csomag, Scanner osztály a util csomagban van, ezért importálni kell.* Melyik metódus szolgál double típusú számok beolvasására? *Válasz: A nextDouble() metódus szolgál double típusú számok beolvasására.* Számok beolvasása esetén mire kell figyelnünk? *Válasz: A megfelelő metódus nem olvassa ki az Entert, ezért a nextLine() metódus segítségével erről gondoskodnunk kell.*	A szükséges előzetes ismeretek felidézése a feladat megoldásához.	Beszélgetés

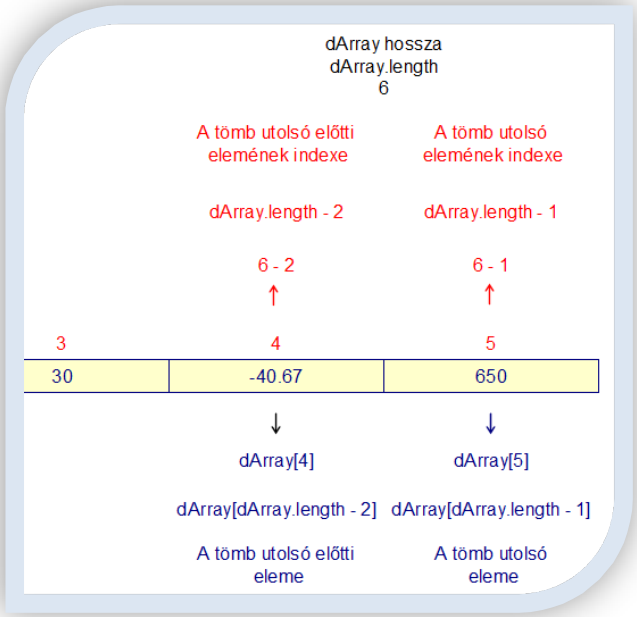
16 – 17’	Kódoljon a struktogram alapján!
	`package gyakorlat13;`
	`import java.util.Scanner;`
	`/**`
	` *`
	` * @author Kovács Henriette`
	` */`
	`public class Gyakorlat13_4 {`
	`  public static void main(String[] args) {`
	`  // 1. feladat`
	`  double[] dArray = new double[6];`
	`  Scanner billentyuzet = new Scanner(System.in);`
		Programkód írása
		Precizitás, pontosság
			Egyéni munka
			Számítógép
			Java 1.5
			NetBeans IDE 7.3

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK

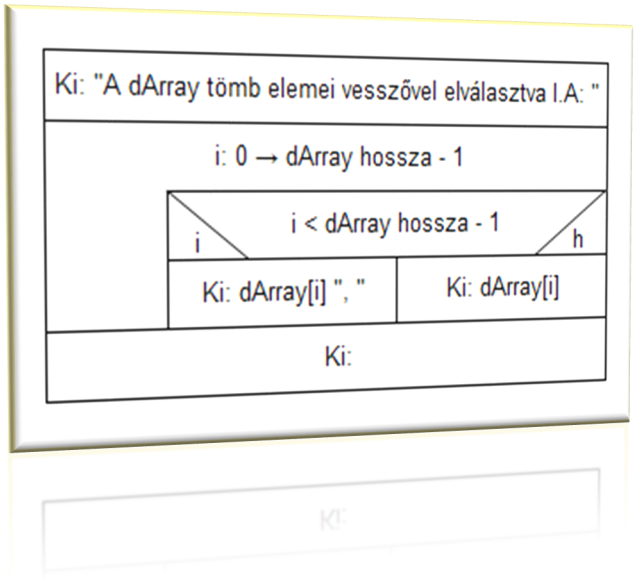
17 – 22’	
	``
	// dArray tömb elemeinek a feltöltése billentyűzetről
	for (int i = 0; i < dArray.length; i++) {
	    System.out.print("Kérem a tömb " + (i + 1) + ". elemét! ");
	    dArray[i] = billentyuzet.nextDouble();
	    billentyuzet.nextLine();
	}
	
	// dArray tömb elemeinek kiíratása tabulátorral elválasztva
	System.out.println("A dArray tömb elemei tabulátorral elválasztva:");
	for (int i = 0; i < dArray.length; i++) {
	    System.out.print(dArray[i] + "\t");
	}
	System.out.println();
	``
		Szintaktika követése	Egyéni munka
			
			Számítógép
			Java 1.5
			NetBeans IDE 7.3

22 – 23’	Kérem, ellenőrizzék a kivetített kód alapján, hogy jól dolgoztak-e! Az esetleges hibákat javítsák!	Önellenőrzés Diagnosztizálás	Projektor
23 – 24’	2.) Feladat előkészítése I. Gondoljuk át a tömb elemeinek indexelését a length konstans felhasználásával! Példaként vegyünk egy 6 elemű tömböt. Mi a tömb utolsó elemének indexe? *Válasz: A tömb utolsó elemének indexe 5.* Utolsó előtti elemének indexe? 4 Hogyan hivatkozunk ezen indexű elemekre, ha a tömb neve mondjuk dArray? *Válasz: Utolsó elemre: dArray[5]* *Utolsó előtti elemre: dArray[4]*	Visszaemlékezés Visszacsatolás	Megbeszélés Frontális munka

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
24 – 28’	Hogyan határozhatjuk meg a length konstanssal ezen tömb hosszát? *Válasz: dArray.length* Nézzük meg, ha általánosságban szeretnénk megfogalmazni, nem konkrét értékekkel, akkor mit mondhatunk az indexekről? *Válasz: A tömb utolsó elemének indexe dArray.length – 1* *A tömb utolsó előtti elemének indexe dArray.length – 2* Hogyan hivatkozunk ezen indexű elemekre, ha a tömb neve mondjuk dArray? *Válasz: Utolsó elemre: dArray[dArray.length – 1]* *Utolsó előtti elemre: dArray[dArray.length – 2]*	Absztrakt gondolkodás Általánosítás	Megbeszélés Frontális munka
	Segít a megértésben a hálózatra feltett kép, elemezzük! 	Szemléletesség	TombHossza_Index.png Szemléltetés Magyarázat Frontális munka Hálózat Projektor

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
	A kép egy részletét, ami kinagyítja a lényeget, szintén feltettem a hálózatra, le tudják tölteni!  	Rögzítés Megerősítés	TombHossza_Index_resze.png Elbeszélés Frontális munka Hálózat Projektor
28 – 29’	Feladat kiadása: Olvassa el figyelmesen a feladatot! 2.) A feladathoz készítsen struktogramot! Írassa ki a dArray tömb elemeit vesszővel elválasztva úgy, hogy az utolsó elem után ne legyen vessző! Keressen többféle megoldást!	Feladatértelmezés	Projektor Hálózat Egyéni munka

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
29 – 32’	2.) Feladat további előkészítése II. Próbálja megfogalmazni lépésről lépésre, hogy mit tesz! *Válasz: Kiíratom az első elemet, ha nem az utolsó elem vesszőt teszek, kiíratom a következő elemet, ha nem az utolsó elem, vesszőt teszek, … és így tovább … kiíratom a következő elemet, ez az utolsó elem, nem teszek vesszőt.* Mikor teszek vesszőt? *Válasz: Mindig teszek vesszőt, ha nem az utolsó elemnél vagyok.* Milyen elemindexnél teszek vesszőt? *Válasz: Akkor teszek vesszőt, ha az elemindex kisebb, mint az utolsó elem indexe.* Mitől függ, hogy teszek-e vesszőt? *Válasz: Egy feltételtől függ, méghozzá attól, hogy az elemindex kisebb-e, mint az utolsó elem indexe.* Milyen vezérlési szerkezetet használok ennek megvalósítására? *Válasz: Elágazást.* Hányszor kell megvizsgálnom ezt a feltételt? *Válasz: Ahány eleme van a tömbnek, annyiszor kell megvizsgálnom a feltételt.* Milyen vezérlési szerkezetet használok ennek megvalósítására? *Válasz: Léptetéses ciklust fogok használni.*	Logikus gondolkodás Fokozatosság Algoritmikus gondol-kodás Algoritmus tervezése Algoritmus alkotása	Megbeszélés Frontális munka

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
32 – 35’	Készítsék el a struktogramot, használják hozzá a legoptimálisabb programot!	Struktogram készítés Precizitás, pontosság Aktivizálás	Microsoft Office Excel 2007 Structorizer
		Algoritmus készítése Struktogram készítése	Egyéni munka
35 – 36’	Kérem, ellenőrizzék a kivetített struktogram alapján, hogy jól dolgoztak-e! Az esetleges hibákat javítsák!	Önellenőrzés Diagnosztizálás	Projektor
	A struktogram alapján kódoljanak!	Feladat kitűzése	

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
36 – 41’		Programkód írása Precizitás, pontosság Szintaktika követése	Számítógép Java 1.5 NetBeans IDE 7.3
41 – 42’	Kérem, ellenőrizzék a kivetített kód alapján, hogy jól dolgoztak-e! Az esetleges hibákat javítsák!	Önellenőrzés Diagnosztizálás	Projektor
42 – 44’	Mivel gyarapítottuk a tudásunkat a mai órán ☺? Válasz: Fel tudunk tölteni tömböket a billentyűzetről és ki tudjuk íratni a tömb elemeit vesszővel elválasztva. Ezzel az utolsó feladattal fogjuk folytatni a következő órát!	Rendszerezés Figyelem felkeltése	Együttműködő oktatási módszer A KÖVETKEZŐ ÓRA MEGTARTÁSÁRA KÖZVETLENÜL EZ UTÁN AZ ÓRA UTÁN KERÜL SOR, EZÉRT NINCS HÁZI FELADAT.

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
44 – 45’	Tanulók munkájának értékelése	Értékelés	Megbeszélés Egész órát végig kíséri az értékelés.

Feladatok

1.) A feladathoz készítsen struktogramot!
Hozzon létre egy dArray nevű, double elemtípusú tömböt, melynek 6 eleme van!
   a) Töltse fel a tömböt a felhasználó által megadott értékekkel!
   b) Írassa ki a tömb elemeinek értékét egymás mellé tabulátorral elválasztva!

2.) A feladathoz készítsen struktogramot!
Írassa ki a dArray tömb elemeit vesszővel elválasztva úgy, hogy az utolsó elem után ne legyen vessző!
Keressen többféle megoldást!

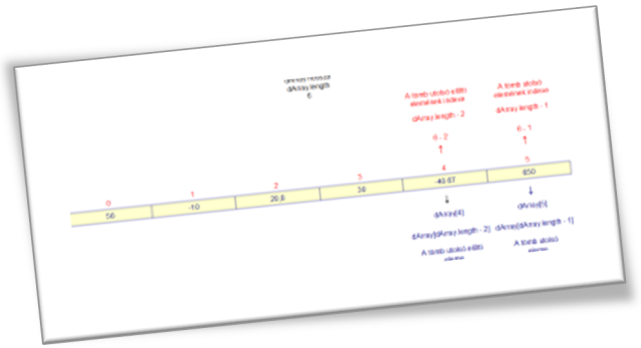
Reflexió

A diákok mind a két feladathoz kellően motiváltak voltak, anélkül, hogy bármilyen erőfeszítést tettem volna ennek érdekében, bár megjegyzem, hogy szünetben való beszélgetésünk alkalmával felmerült már ennek igénye részükről. Egyszóval mind a billentyűzetről való feltöltés, mind a tömbelemek vesszővel elválasztva történő megjelenése érdekelte őket. Még az algoritmikus gondolkodásuk is szárnyakat kapott, holott ez azért nehézségekbe szokott ütközni. Tulajdonképpen rájöttek az egyik lehetséges algoritmusra és pont arra, amelyiket erre az órára beterveztem, de ha nem így lett volna, akkor pusztán felcserélődtek volna a lehetséges sorrendek. A pontos megfogalmazásban segítenem kellett, arra mindig különös gondot fordítok, hogy világosan, precízen és pontosan fogalmazzanak (egyik diáknál ennek organikus oka van, vele a kommunikáció nagyon nehézkes élő szó formájában), a lényeg, hogy erre törekszem, ezáltal bennük is próbálom kialakítani az ez iránti igényt.

Nagyon fontos a programozás oktatása során az absztrakt gondolkodás fejlesztése, ezt a célt szolgálta a tömbelemek indexeinek, illetve a tömbelemek elérésének általános formában való megjelenítése, ami szükséges a következő megoldás során, úgymint:

dArray.length – 1	dArray.length – 2	dArray[dArray.length – 1]	dArray[dArray.length – 2]

Ez nehézségekbe ütközött és ezért is örültem nagyon, hogy készítettem egy képet, ami mindezt szemléletesen megmutatta nekik és ez sokat is segített.



TombHossza_Index.png

Mindezzel a következő órát is előkészítettem. Ennek megértése nagyon fontos a tömbökkel, illetve az indexekkel való foglalkozás során.

ÓRATERVEZET - TÖMB ELEMEINEK KIÍRATÁSA VESSZŐVEL ELVÁLASZTVA. INICIALIZÁLÓ BLOKK.

A pedagógus neve: Kovács Henriette

A szakmacsoport száma és megnevezése: 7. Informatika

A szakképesítés azonosító száma: 54 213 05

A szakképesítés megnevezése: Szoftverfejlesztő

Iskolai rendszerű szakképzés (a szakképzési évfolyamok száma: 2)

Szakmai követelménymodul: 10817-12 Hálózatok, programozás és adatbázis-kezelés

Tantárgy: Adatbázis- és szoftverfejlesztés gyakorlat

Programozási nyelv: Java

Fejlesztői környezet: NetBeans IDE 7.3

Osztály - csoport: I/13. S - SZ2

Az óra tananyaga: Tömb elemeinek kiíratása vesszővel elválasztva. Inicializáló blokk.

Az óra cél- és feladatrendszere

A fejlesztendő attitűd, készségek, képességek, a tanítandó ismeretek (fogalmak, szabályok stb.) és az elérendő fejlesztési szint, tudásszint megnevezése:

- A tanuló tartsa be a számítógépes munka szabályait, különös tekintettel a balesetek megelőzésére. A berendezésekkel fegyelmezetten, a használati utasításokat pontosan követve dolgozzon. Legyen tisztában a számítógépes környezet alapvető ergonómiai kérdéseivel, az egészségvédelem lehetőségeivel számítógépes munkakörnyezetben.
- Felkelteni és folyamatosan ébren tartani a tanulók érdeklődését a szoftverfejlesztés iránt. Kialakítani a tanulókban az informatikai ismereteik folyamatos bővítésének és megújításának igényét.
- Korszerű alkalmazói készség kialakítása: meglévő, - Internet segítségével - új alkalmazások kiválasztása a feladat tervezéséhez és legyenek képesek arra, hogy mindezeket célszerűen használják.
- Az algoritmikus gondolkodás fejlesztése: a tanulók önálló, rendszerezett, logikus gondolkodásának fejlesztése; a logikus gondolkodás lényeges a problémamegoldásban, és az algoritmusok elkészítésében, kódolásában.
- Legyen képes a NetBeans fejlesztői környezet kényelmi funkcióinak készségszintű használatára.
- Önálló munkára nevelés és felzárkóztatás.
- Alkotó munkára nevelés.
- Az esztétikai készség fejlesztése: igény és készség a számítógépes produktum esztétikus formájának kialakítására, a program helyes strukturálására.
- *Legyen képes az algoritmust megérteni, felismerni, elemezni, végrehajtani, alkotni, megvalósítani, módosítani, átalakítani, tervezni.*
- *Tudjon tömb típusú változót deklarálni, létrehozni, feltölteni.*
- *Legyen képes a tömbök elemeit elérni.*
- *Tudjon tömböt többféle módon és többféle algoritmussal megjeleníteni.*
- *Tudjon tömböt létrehozni inicializáló blokk segítségével.*
- *Ismerje a hatékonyság-vizsgálat jelentőségét.*
- *Tudja a feltételvizsgálatok, létrejött objektumok számát meghatározni.*

Fejlesztendő készségek, képességek

– Visszaemlékezés, kommunikációs képesség, programkód átalakítása, módosítása, önellenőrzés, struktogram olvasása, megértése, struktogram elemzése, algoritmus megfogalmazása, algoritmikus gondolkodás, programkód írása, precizitás, pontosság, szintaktika követése, diagnosztizálás, hibakeresés, programkód olvasása, programkód megértése, analizálása, állománykezelés, döntésképesség, hatékonyság-vizsgálat.

Fejlesztendő attitűd

– Legyen képes: számítógépet kezelni, szoftvereket használni.
– Tudja önállóan használni a hálózatot és annak alapszolgáltatásait.
– Legyen képes az adott probléma megoldásához kiválasztani az általa ismert módszerek, eszközök és alkalmazások közül a megfelelőt.
– Ismerje fel és legyen képes különféle formákban megfogalmazni a tevékenységek algoritmizálható részleteit.
– Adott feladat megoldásához legyen képes algoritmusokat megfogalmazni, tervezni és megvalósítani.
– Legyen képes programozási feladatot ellátni, egy szoftvert elkészíteni (kódolni), Java programnyelven implementálni.
– Legyen képes programjait úgy megírni, hogy azt később Ő maga is, vagy tanulótársa is fel tudja használni.

Az óra didaktikai feladatai

- Motiválás
- A figyelem felkeltése
- A tanulók informálása a tanulási célokról.
- Aktivizálás
- Ismétlés
- Fokozatosság
- Feladat továbbfejlesztése
- Visszacsatolás
- Ismétlés
- Alkalmazás
- Megerősítés
- Feladatkitűzés
- Folyamat elemzése
- Következtetés, általánosítás
- A tények elemzése
- Új ismeret közlése
- Rögzítés
- Gyakorlás
- Rendszerezés

Tantárgyi kapcsolatok: Adatbázis- és szoftverfejlesztés elmélet, Matematika, Angol, Hálózati ismeretek I., Informatikai alapismeretek

Felhasznált források (tankönyv, munkafüzet, feladat- és szöveggyűjtemény, digitális tananyag, online források, szakirodalom stb.):

Szerző(k)	Cím	Kiadó	ISBN kód
Vég Csaba – dr. Juhász István	Java – start!	Logos 2000	963 03 9005 1
Vég Csaba	Instant Java/Java EE/SOA I-II	Logos 2000	978-963-06-2601-9 978-963-06-2602-6
Dirk Louis – Peter Müller:	Java 5 Belépés a programozás világába	Panem	963 545 454 6
Benkő Tiborné, Tóth Bertalan	Együtt könnyebb a programozás Java	Computerbooks	963 618 337 6
Rogers Cadenhead	Tanuljuk meg a Java programozási nyelvet 24 óra alatt	Kiskapu Kft.	963 9637 07 6
Angster Erzsébet	Objektumorientált tervezés és programozás Java 1 - 2	4KÖR Bt.	963 00 6262 3 963 00 6263 1
Peter J. DePasquale	Java zsebkönyv	Kiskapu Kft.	963 9637 10 6
Nyékyné Gaizler Judit, Frohner Ákos, Kozsik Tamás (Szerkesztők)	Java 2 útikalauz programozóknak	ELTE TTK Hallgatói Alapítvány	963 463 305 6
Kathy Sierra, Bert Bates	Agyhullám: Java	Kiskapu Kft.	978 963 9637 79 5

http://docs.oracle.com/javase/7/docs/api/

Dátum: éééé. hh. nn.

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MÓDSZEREK TANULÓI MUNKAFORMÁK ESZKÖZÖK MEGJEGYZÉSEK
0 – 3’	Az órát bevezető kérdések: 1. Hogyan tudunk létrehozni tömböket? new operátorral *meg fogunk ismerkedni még egy lehetőséggel, az úgynevezett inicializáló blokkal (mai órán)* 2. Hogyan tudunk feltölteni tömböket? egyesével, szekvenciával szabály alapján, vezérlési szerkezettel, léptetéses ciklussal felhasználó által *véletlenszerűen* 3. Hogyan végeztük tömbök kiíratását? szóközzel, tabulátorral elválasztva vesszővel elválasztva *(folytatjuk)* A Gyakorlat13 projekt gyakorlat13 csomagjában létrehozott Gyakorlat13_4 osztályban folytatjuk a kódolást!	Visszaemlékezés Kommunikációs képesség Motiválás A figyelem felkeltése A tanulók informálása a tanulási célokról. Aktivizálás Ismétlés Fokozatosság	Ez az óra közvetlenül kapcsolódik az előző órához. Az utolsó feladatot folytatjuk. Beszélgetés Frontális munka Projektor Számítógép NetBeans IDE 7.3 Java 1.5

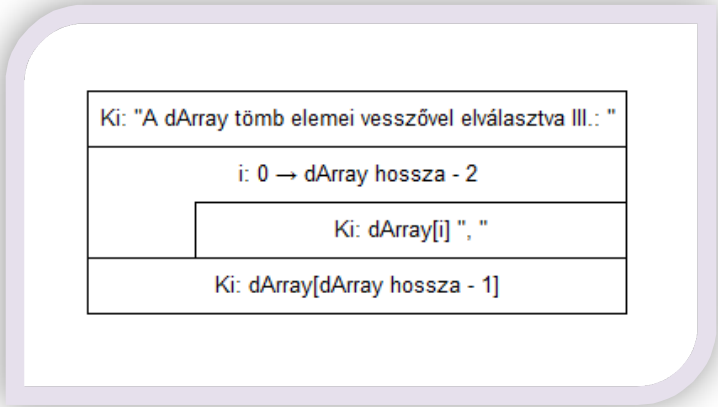
IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
3 – 4’	3.) Előző órai feladat folytatása. A feladathoz készítsen struktogramot! Írassa ki a dArray tömb elemeit vesszővel elválasztva úgy, hogy az utolsó elem után ne legyen vessző! Keressen többféle megoldást!	Visszacsatolás Feladat továbbfejlesztése	Projektor Hálózat
4 – 5’	Kódoljuk le egyszerűbben az elágazást! A kétágú elágazást mivel tudnánk helyettesíteni? *Válasz: A feltételes kiértékelés segítségével egyszerűbb lenne a kód.* ☺ Mi a szintaktikája? *Válasz:* `feltétel ? kif1 : kif2` Ha a feltétel igaz, akkor kifejezés1, egyébként kifejezés2 kerül kiértékelésre.	Ismétlés Alkalmazás	Tábla Megbeszélés Frontális munka

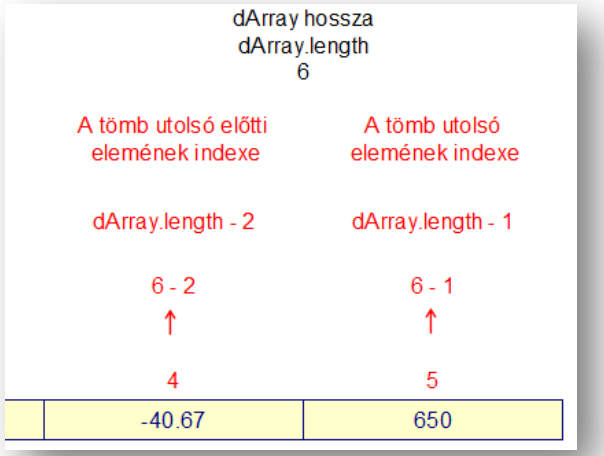
5 – 9’	`// dArray tömb elemeinek kiíratása vesszővel elválasztva I.B`
	`System.out.println("A dArray tömb elemei vesszővel elválasztva I.B:");`
	`for (int i = 0; i < dArray.length; i++) {`
	`  System.out.print(i < dArray.length - 1`
	`  ? dArray[i] + ", " : dArray[i]);`
	`}`
	`System.out.println();`
		Programkód átalakítása, módosítása
		Megerősítés
			Egyéni munka
			Számítógép
			NetBeans IDE 7.3
			Java 1.5

9 – 10’	A kivetített kód alapján ellenőrizze munkáját, ha szükséges javítsa a hibákat! .	Önellenőrzés Hibakeresés	Projektor
10 – 13’	Hány feltételvizsgálat történik a kódban?		

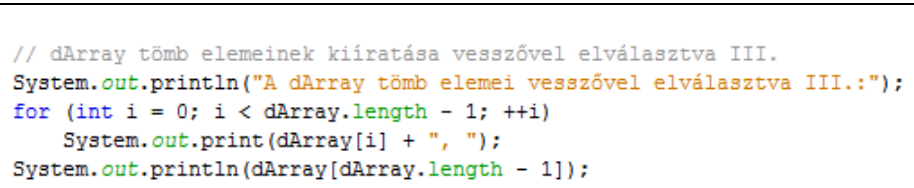
IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
	6 elemű tömb esetén: 0 < 6 0 < 5 1 < 6 1 < 5 2 < 6 2 < 5 3 < 6 3 < 5 4 < 6 4 < 5 5 < 6 5 < 5 6 < 6 } 6 * 2 + 1 *Válasz: dArray.length * 2 + 1*	Hatékonyság-vizsgálat Folyamat elemzése Következtetés, általánosí-tás	Magyarázat Szemléltetés Tábla Frontális munka

13 – 15’	Olvassa el a következő kódrészletet, elemezze, értelmezze!
	`// dArray tömb elemeinek kiíratása vesszővel elválasztva II.`
	`System.out.println("A dArray tömb elemei vesszővel elválasztva II.:");`
	`String szoveg = "";`
	`for(int i = 0; i < dArray.length; i++){`
	`  szoveg += (i < dArray.length - 1) ? dArray[i] + ", " : dArray[i];`
	`}`
	`System.out.println(szoveg);`
	Hány feltételvizsgálat történik, hány String objektum keletkezik?
		Programkód olvasása
		Programkód megértése, analizálása
		Hatékonyság-vizsgálat
			Projektor
			Gyakorlat13_4_2_II.png
			Egyéni munka

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
15 – 16’	Mi történik a kódban? *Válasz: Egy üres Stringhez mindig hozzáfűzöm a tömb egy elemét és egy vesszőt, az utolsó elemnél csak az elemet.* *dArray.length * 2 + 1 feltételvizsgálat történik.* *dArray.length + 1 String objektum keletkezik.*	Kommunikációs képes-ség	Tanulói elbeszélés
16 – 19’	Kódolja le String objektum felhasználásával a tömb vesszővel történő kiíratását!	Visszaemlékezés	Egyéni munka
19 – 20	Elemezzék a következő struktogramot, és fogalmazzák meg szavakkal az algoritmust, hogyan kerülnek a tömb elemei közé a vesszők!  Hány feltételvizsgálat történik?	Struktogram olvasása, megértése Struktogram elemzése Hatékonyság-vizsgálat	Projektor Hálózat Gyakorlat13_4_2_III_ Struk-togram. png Frontális munka

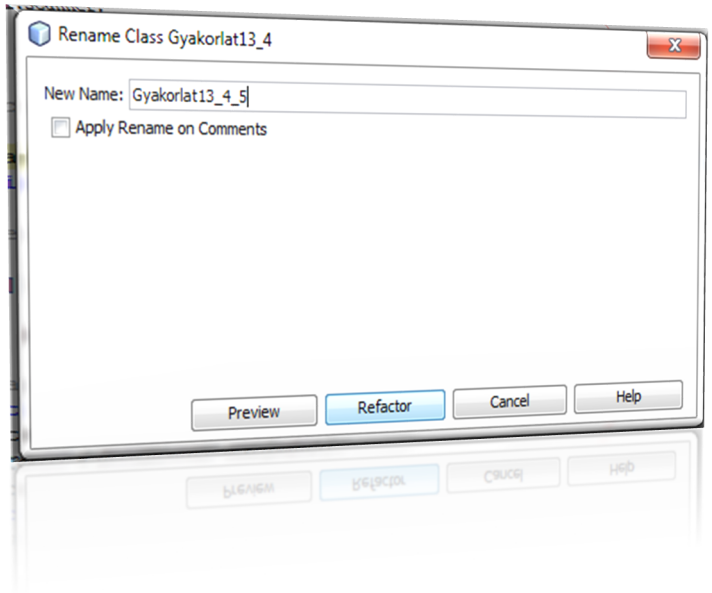
IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
20 – 21’	Emlékezzünk vissza, melyik elem indexe a dArray.length – 2?  *Válasz: A tömb utolsó előtti elemének indexe.*	Visszacsatolás	Szemléltetés Projektor Beszélgetés
21 – 23’	Struktogram elemzése *Válasz: Algoritmus: A tömb első elemétől kezdve végigmegyek a tömb elemein, de csak az utolsó előtti elemig. Mindig azt csinálom, hogy kiíratom az elemet és egy vesszőt.* *Aztán kiíratom a tömb utolsó elemét!* *Válasz: dArray.length feltételvizsgálat történik.*	Algoritmikus gondolkodás Algoritmus megfogalmazása	Tanulói elbeszélés
	A struktogram alapján kódoljanak!	Feladatkitűzés	

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK

23 – 26’	
	// dArray tömb elemeinek kiíratása vesszővel elválasztva III.
	System.out.println("A dArray tömb elemei vesszővel elválasztva III.:");
	for (int i = 0; i < dArray.length - 1; ++i)
	System.out.print(dArray[i] + ", ");
	System.out.println(dArray[dArray.length - 1]);
		Programkód írása
		Precizitás, pontosság
		Szintaktika követése
			Egyéni munka
			Számítógép
			NetBeans IDE 7.3
			Java 1.5

26 – 27’	Ellenőrizze munkáját!	Önellenőrzés Diagnosztizálás, hibakeresés	Projektor
27 – 28’	A 3 kiíratás közül melyikben történik a legkevesebb feltételvizsgálat? *Válasz: III-ban* Melyik kiíratást tartja a legkönnyebnek? ☺ Hatékonyság problémája.	A tények elemzése	Frontális munka Megbeszélés
28 – 29’	Óra elején már említettem, hogy a new operátor nélkül is létre lehet hozni méghozzá sokkal egyszerűbben tömböket. Nézzük meg, hogyan!	A figyelem felkeltése	

29 – 30’	Deklaráláskor a tömb elemeinek kezdeti értéket adhatunk úgynevezett inicializáló blokkal. Szintaktikája:
	`<elemtípus>[] <tömbAzonosító> = {<érték0>, <érték1>,…};`
	A kapcsos zárójelek által határolt blokkot inicializáló blokknak nevezzük. Ebben az esetben a tömb mérete pontosan akkora lesz, amennyi a felsorolt értékek száma.
	`int[] iArray = {0, 20, 40, 60, 100};`
		Új ismeret közlése	Elbeszélés
			Tábla

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
30 – 31’	Gyakorlat13_4 .java osztályt nevezze át Gyakorlat13_4_5.java osztályra!  Hozzon létre egy Gyakorlat13_5.java osztályt és a következő feladatot ebben végezze el!	Állománykezelés	Irányított, önálló munka Projektor NetBeans IDE 7.3 Számítógép
30 – 31’	Feladat: Hozzuk létre az iArray nevű int elemtípusú tömböt a táblára felírt elemekkel: 0, 20, 40, 60, 100! Írassa ki ez elemeket egymás mellé, vesszővel elválasztva!	Feladatkitűzés	Elbeszélés

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK

31 – 34’	`package gyakorlat13;`
	
	`/**`
	` *`
	` * @author Kovács Henriette`
	` */`
	`public class Gyakorlat13_5 {`
	`  public static void main(String[] args) {`
	
	`  // 1. feladat`
	`  // iArray tömb létrehozása inicializálló blokkal`
	`  int[] iArray = {0, 20, 40, 60, 100};`
	
	`  // az iArray tömb elemeinek kiíratása`
	`  System.out.println("Az iArray tömb elemei vesszővel elválasztva:");`
	`  for (int i = 0; i < iArray.length - 1; ++i)`
	`  System.out.print(iArray[i] + ", ");`
	`  System.out.println(iArray[iArray.length - 1]);`
	`  }`
	`}`
		Rögzítés
		
		Döntésképesség
			Projektor
			
			NetBeans IDE 7.3
			Magyarázat
			Frontális munka
			
			Egyéni munka

34 – 38’	Feladat 2.) Az inicializáló blokk felhasználásával hozzon létre a) egy mgh nevű tömböt, mely az angol ábécé magánhangzóit tartalmazza! Írassa ki a tömb elemeit vesszővel elválasztva! b) egy szep nevű, logikai elemtípusú tömböt, mely a lehetséges logikai értékeket tartalmazza! Írassa ki a tömb elemeit vesszővel elválasztva!	Gyakorlás	Differenciálás ketten az a), egy diák a b) feladatot készíti el Egyéni munka NetBeans IDE 7.3

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK

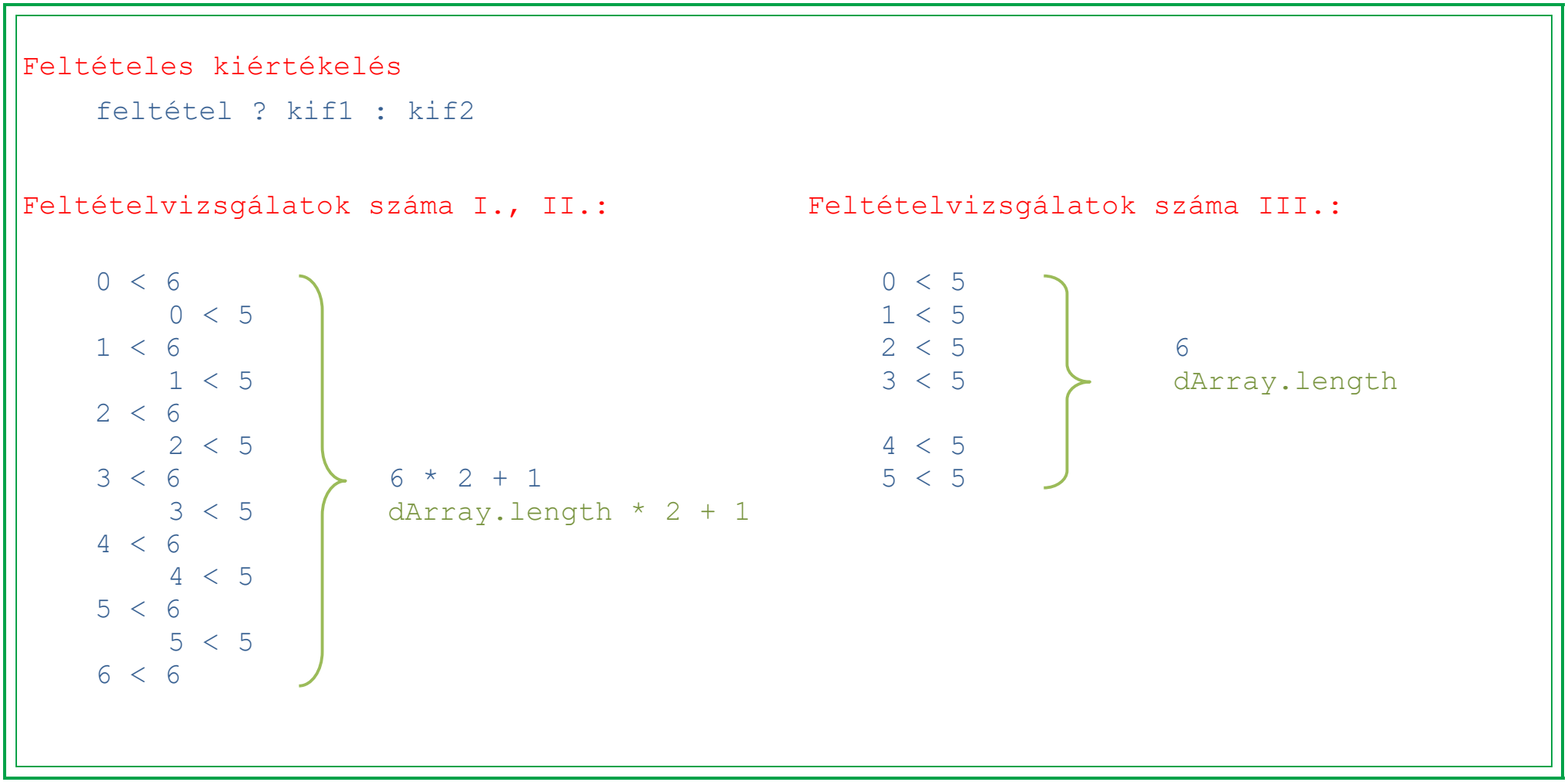
38 – 39’	Ellenőrizzék
	
	`char[] mgh = {'a', 'e', 'i', 'o', 'u'};`
	
	`// az mgh tömb elemeinek kiíratása`
	`System.out.println("Az mgh tömb elemei vesszővel elválasztva:");`
	`for (int i = 0; i < mgh.length - 1; ++i)`
	`System.out.print(mgh[i] + ", ");`
	`System.out.println(mgh[mgh.length - 1]);`
		Önellenőrzés	Egyéni munka
			Angol ábécé ismerete szükséges
			Projektor

38 – 39’	`boolean[] szep = {true, false};`
	
	`// a szep tömb elemeinek kiíratása`
	`System.out.println("A szep tömb elemei vesszővel elválasztva:");`
	`for (int i = 0; i < szep.length - 1; ++i)`
	`System.out.print(szep[i] + ", ");`
	`System.out.println(szep[szep.length - 1]);`
		Önellenőrzés	Egyéni munka
			Projektor

39 – 43’	Rendszerezés: Azon tény megállapítása, hogy nem elegendő egy megoldás megtalálása, mindig meg kell győződnünk róla, hogy az optimális-e, hatékony-e! Melyik vesszővel való kiíratással fog dolgozni az elkövetkezőkben? Ha a tömbit inicializáló blokkal hozom létre, még mi történik a létrehozáson kívül?	Rendszerezés	Vita A KÖVETKEZŐ ÓRA MEGTARTÁSÁRA KÖZVETLENÜL EZ UTÁN AZ ÓRA UTÁN KERÜL SOR, EZÉRT NINCS HÁZI FELADAT.

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
	Válasz: Inicializáló blokkal való létrehozáskor a tömböt fel is töltöm elemekkel.		Megbeszélés Frontális munka
43 – 45’	Tanulók munkájának értékelése	Értékelés	Megbeszélés Egész órát végig kíséri az értékelés.

TÁBLAKÉP 1.


Feltételes kiértékelés
feltétel ? kif1 : kif2
Feltételvizsgálatok száma I., II.:
0 < 6
0 < 5
1 < 6
1 < 5
2 < 6
2 < 5
3 < 6
3 < 5
4 < 6
4 < 5
5 < 6
5 < 5
6 < 6
6 * 2 + 1
dArray.length * 2 + 1
Feltételvizsgálatok száma III.:
0 < 5
1 < 5
2 < 5
3 < 5
4 < 5
5 < 5
6
dArray.length

TÁBLAKÉP 2.

Inicializáló blokk

```
    <elemtípus>[] <tömbAzonosító> = {<érték0>, <érték1>,…};

                    int[] iArray = {0, 20, 40, 60, 100};
```

Feladatok

1.) Előző órai feladat folytatása.
Írassa ki a dArray tömb elemeit vesszővel elválasztva úgy, hogy az utolsó elem után ne legyen vessző!
Keressen többféle megoldást!

2.) Az inicializáló blokk felhasználásával hozzon létre
   a) egy mgh nevű tömböt, mely az angol ábécé magánhangzóit tartalmazza!
   Írassa ki a tömb elemeit vesszővel elválasztva!
   b) egy szep nevű, logikai elemtípusú tömböt, mely a lehetséges logikai értékeket tartalmazza!

Reflexió

Ezt az órát úgy terveztem meg, hogy e tantárgy szempontjából az egyik legfontosabb informatikai kompetencia komponenssel, az algoritmikus gondolkodással és annak fejlesztésével foglalkoztam. Mindennapjainkban, mind a tanulásban, mind a munkában lépten-nyomon algoritmusokat hajtunk végre, mégis e tevékenységsorozatok megfogalmazása gondot okoz. Ezen probléma kialakulásához több tényező is hozzájárul, mint például azon egyszerű ok, hogy az emberek nem beszélgetnek egymással, keveset olvasnak, nem kommunikálnak, elmennek egymás mellett, magukba fordulnak. Ezt figyelembe véve, óráimon mindig törekszem arra, hogy diákjaimat beszéltessem, hogy kommunikáljunk, mindezzel hozzájárulva ahhoz, hogy szakmai nyelven is pontosan, precízen fogalmazzák meg észrevételeiket és így közvetett módon is fejlesszem algoritmikus gondolkodásukat.

A vesszővel való kiíratás feladatkörét tekintve, a különböző megközelítésű részfeladatok egy sor, az algoritmikus gondolkodás kompetencia szintjeit képező összetevőket fejleszt. Ezek: az algoritmus megértése, azaz megérteni, hogy mit kell tenni és megérteni, hogy miért azt kell tenni, az algoritmus felismerése, az algoritmus elemzése, az algoritmus végrehajtása, az algoritmus alkotása, az algoritmus megvalósítása, az algoritmus módosítása, átalakítása, az algoritmus tervezése. Láthatjuk, hogy ez az első ránézésre banálisnak tűnő feladat is milyen fejlesztési lehetőségeket rejt magában. A diákok emellett élvezték, hogy mindig más típusú feladatokat kellett végrehajtaniuk, hol algoritmust kitalálni, hol megérteni, hol átalakítani, hol újabb lehetőségeket keresni, így érdeklődésüket az egész feladat során fenn tudtam tartani. Azt tapasztaltam, hogy az is nagyon jó motiváló erő számukra, hogy hol alkalmazói, hol fejlesztői környezetben dolgoznak.

Az óra végére egy új ismereteket feldolgozó egységet választottam, így sor került új ismeretek elsajátítására is. Sokat töprengtem, hogy vajon jó gondolat-e, jó ötlet-e az óra végén bevezetni egy új fogalmat. Az előző órai feladatot mindenképpen folyamatában szerettem volna hogy lássák a diákok (hat óra egy napon, egymás után van megtartva minden héten), így a kérdés csak az volt, hogy az óra végére, vagy a következő óra elejére tegyem, és az órát egy gyakorló feladattal zárjam. Ez az új anyag viszont annyira egyszerű, úgy éreztem inkább feldobná az órát, új erőt kapnának a diákok, és gondoltam kipróbálom. Megérzésem valóra vált, örültek az új tömbfeltöltési lehetőségnek és a differenciálás mindenképpen a feladat megoldására sarkallta őket.

Ennek az órának a felépítéséért tett lépések sorozata megérte a fáradozást, a diákok élveztek a változatosságot.

ÓRATERVEZET - VÉLETLENSZÁMOK GENERÁLÁSA. TÖMB ELEMEINEK FELTÖLTÉSE VÉLETLENSZÁMOKKAL.

A pedagógus neve: Kovács Henriette

A szakmacsoport száma és megnevezése: 7. Informatika

A szakképesítés azonosító száma: 54 213 05

A szakképesítés megnevezése: Szoftverfejlesztő

Iskolai rendszerű szakképzés (a szakképzési évfolyamok száma: 2)

Szakmai követelménymodul: 10817-12 Hálózatok, programozás és adatbázis-kezelés

Tantárgy: Adatbázis- és szoftverfejlesztés gyakorlat

Programozási nyelv: Java

Fejlesztői környezet: NetBeans IDE 7.3

Osztály - csoport: I/13. S - SZ2

Az óra tananyaga: Véletlenszámok generálása. Tömb elemeinek feltöltése véletlenszámokkal.

Az óra cél- és feladatrendszere

A fejlesztendő attitűd, készségek, képességek, a tanítandó ismeretek (fogalmak, szabályok stb.) és az elérendő fejlesztési szint, tudásszint megnevezése:

- A tanuló tartsa be a számítógépes munka szabályait, különös tekintettel a balesetek megelőzésére. A berendezésekkel fegyelmezetten, a használati utasításokat pontosan követve dolgozzon. Legyen tisztában a számítógépes környezet alapvető ergonómiai kérdéseivel, az egészségvédelem lehetőségeivel számítógépes munkakörnyezetben.
- Felkelteni és folyamatosan ébren tartani a tanulók érdeklődését a szoftverfejlesztés iránt.
- Korszerű alkalmazói készség kialakítása.
- Az algoritmikus gondolkodás fejlesztése.
- Legyen képes a NetBeans fejlesztői környezet kényelmi funkcióinak készségszintű használatára.
- Önálló munkára nevelés és felzárkóztatás.
- Alkotó munkára nevelés.
- Az esztétikai készség fejlesztése: igény és készség a számítógépes produktum esztétikus formájának kialakítására, a program helyes strukturálására.
- *Tudja tömb elemszámát bekérni.*
- *Tudjon tömb típusú változót deklarálni, létrehozni, feltölteni.*
- *Legyen képes a tömbök elemeit elérni.*
- *Tudjon tömböt megjeleníteni.*
- *Ismerje az üres tömb és üres String fogalmakat.*
- *Tudja mi a különbség a Math osztály random() metódusa által adott és a Random osztály metódusai által generált számok között.*
- *Ismerje az álvéletlen szám és véletlenszám fogalmakat.*
- *Tudjon intervallumot bővíteni és eltolni.*
- *Tudjon véletlenszámot előállítani.*
- *Tudja tömb elemeit véletlenszámmal feltölteni.*
- *Tudja mikor és hogyan kell egy osztályt importálni.*
- *Tudjon tesztelni, diagnosztizálni.*
- *Tudja a futási eredményt értelmezni.*
- *Ismerje a következő kivételeket: ArrayIndexOutOfBoundsException, NegativeArraySizeException, InputMismatchException.*

Fejlesztendő készségek, képességek

Visszaemlékezés, kommunikációs képesség, rendszerező képesség, szakmai nyelvű beszédkészség, feladatértelmezés, állománykezelés, hallott szakmai szöveg megértése, programkód írása, szintaktika követése, precizitás, pontosság, futási eredmény analizálása, algoritmikus gondolkodás, algoritmus tervezése, struktogram olvasása, értelmezése, önellenőrzés, tesztelés, diagnosztizálás, hibakeresés, olvasott angol nyelvű szakmai szöveg megértése, algoritmus alkotása.

Fejlesztendő attitűd

- Legyen képes: számítógépet kezelni, szoftvereket használni.
- Tudja önállóan használni a hálózatot és annak alapszolgáltatásait.
- Legyen képes az adott probléma megoldásához kiválasztani az általa ismert módszerek, eszközök és alkalmazások közül a megfelelőt.
- Ismerje fel és legyen képes különféle formákban megfogalmazni a környezetében előforduló tevékenységek algoritmizálható részleteit.
- Adott feladat megoldásához legyen képes algoritmusokat megfogalmazni, tervezni és megvalósítani.
- Legyen képes alkalmazást (szoftvert) tervezni, fejleszteni, futtatni, tesztelni és magyarázó megjegyzésekkel ellátni, előkészítve ezzel a dokumentáció elkészítését.
- Legyen képes programozási feladatot ellátni, egy szoftvert elkészíteni (kódolni), Java programnyelven implementálni.
- Legyen képes értelmezni és javítani a NetBeans által megjelenített hibaüzeneteket.
- Legyen képes programjait úgy megírni, hogy azt később Ő maga is, vagy tanulótársa is fel tudja használni.
- Legyen képes eligazodni és tudjon keresni a Java hivatalos dokumentációjában.

Az óra didaktikai feladatai

- Ismétlés
- Rendszerezés
- Motiválás
- A figyelem felkeltése
- Tanulók informálása a tanulási célokról
- Feladatkitűzés
- Visszacsatolás
- Összehasonlítás
- Aktivizálás
- Fogalomalkotás
- Gyakorlás
- Következtetés
- Információ elemzése
- A tények elemzése
- Információ gyűjtése
- Új ismereteket megalapozó tények biztosítása
- Alkalmazás
- Fokozatosság
- Összefoglalás
- A házi feladat kijelölése
- Értékelés

Tantárgyi kapcsolatok: Adatbázis- és szoftverfejlesztés elmélet, Matematika, Angol, Hálózati ismeretek I., Informatikai alapismeretek

Felhasznált források (tankönyv, munkafüzet, feladat- és szöveggyűjtemény, digitális tananyag, online források, szakirodalom stb.):

Szerző(k)	Cím	Kiadó	ISBN kód
Vég Csaba – dr. Juhász István	Java – start!	Logos 2000	963 03 9005 1
Vég Csaba	Instant Java/Java EE/SOA I-II	Logos 2000	978-963-06-2601-9 978-963-06-2602-6
Dirk Louis – Peter Müller:	Java 5 Belépés a programozás világába	Panem	963 545 454 6
Benkő Tiborné, Tóth Bertalan	Együtt könnyebb a programozás Java	Computerbooks	963 618 337 6
Rogers Cadenhead	Tanuljuk meg a Java programozási nyelvet 24 óra alatt	Kiskapu Kft.	963 9637 07 6
Angster Erzsébet	Objektumorientált tervezés és programozás Java 1 - 2	4KÖR Bt.	963 00 6262 3 963 00 6263 1
Peter J. DePasquale	Java zsebkönyv	Kiskapu Kft.	963 9637 10 6
Nyékyné Gaizler Judit, Frohner Ákos, Kozsik Tamás (Szerkesztők)	Java 2 útikalauz programozóknak	ELTE TTK Hallgatói Alapítvány	963 463 305 6
Kathy Sierra, Bert Bates	Agyhullám: Java	Kiskapu Kft.	978 963 9637 79 5

http://docs.oracle.com/javase/7/docs/api/

Dátum: éééé. hh. nn.

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MÓDSZEREK TANULÓI MUNKAFORMÁK ESZKÖZÖK MEGJEGYZÉSEK
0 – 3’	Bevezetés: Tömbök deklarálása Tömbök létrehozása Tömbök mérete → mai órán: 0 esetén mi jön létre és negatív szám esetén mi történik? Tömbök feltöltése → mai órán: véletlenszámokkal és nem álvéletlen számokkal → új osztály megismerése Tömbök kiíratása Tömbelemek indexe Tömbelemek elérése ArrayIndexOfBoundsException	Visszaemlékezés Kommunikációs képesség Rendszerező képesség Szakmai nyelvű beszédkészség Ismétlés Rendszerezés Motiválás A figyelem felkeltése Tanulók informálása a tanulási célokról	Számítógép Bekapcsoltatom a gépeket. Megbeszélés Frontális munka

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
3 – 6’	Feladat: 1.) Írjon programot, a) melyben létrehoz egy egész számokból álló, a felhasználó által megadott elemszámú tömböt, b) feltölti a tömb elemeit olyan véletlenszámokkal, melyek a [-6, 16] zárt intervallumba esnek, c) kiíratja a tömböt úgy, hogy elemei egymástól a ’|’ karakterrel legyenek elválasztva, de az utolsó elem után ne legyen ’|’ karakter! Létrehozni saját hálózati meghajtón: Gyakorlat13 projekt/gyakorlat13 csomag/Gyakorlet13_6.java Gyakorlat13 projekt/alapértelmezett csomag/ElemSzam.java Az a) feladat kapcsán: Felhasználó 0-t ad meg: Üres String Üres tömb is létezik Java-ban.	Feladatkitűzés Feladatértelmezés Állománykezelés Visszacsatolás Összehasonlítás	Hálózat Projektor Számítógép NetBeans IDE 7.3 Java 1.5 Egyéni munka Magyarázat

<table>
<tr><th>IDŐKERET</th><th>AZ ÓRA MENETE</th><th>CÉLOK ÉS FELADATOK</th><th>NEVELÉSI-OKTATÁSI STRATÉGIA
MEGJEGYZÉSEK</th></tr>
<tr>
<td>6 – 9’

9 – 10’</td>
<td>Kód:


```
public class ElemSzam {
    public static void main(String[] args) {

        String str1 = new String(); // vagy
        String str2 = ""; // üres String-ek
        System.out.println(str1 + "és" + str2); // str1, str2 léteznek

        int[] array = new int[0]; // üres tömb
        System.out.println(array); // array tömb létezik, ki tudom íratni

    }
}
```

Futási eredmény:


```
run:
és
[I@45bab50a
BUILD SUCCESSFUL (total time: 1 second)
```

</td>
<td>Feladat kitűzése
Hallott szakmai szöveg megértése

Programkód írása
Szintaktika követése
Precizitás, pontosság

Aktivizálás

Futási eredmény analizálása</td>
<td>Frontális munka
Velem párhuzamosan mindenki önállóan dolgozik, a feladatokat élő szóban fogalmazom meg.

Projektor
Számítógép
NetBeans IDE 7.3
Java 1.5

Magyarázat</td>
</tr>
<tr>
<td>10 – 11’</td>
<td>Felhasználó negatív számot ad meg:

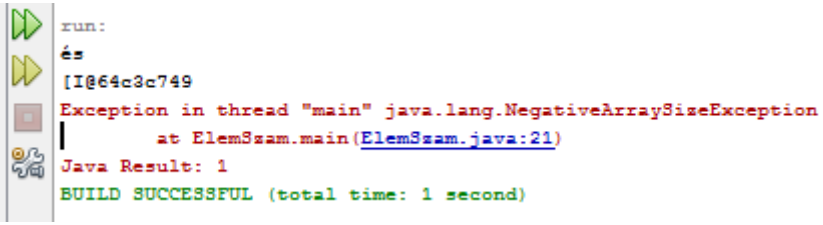
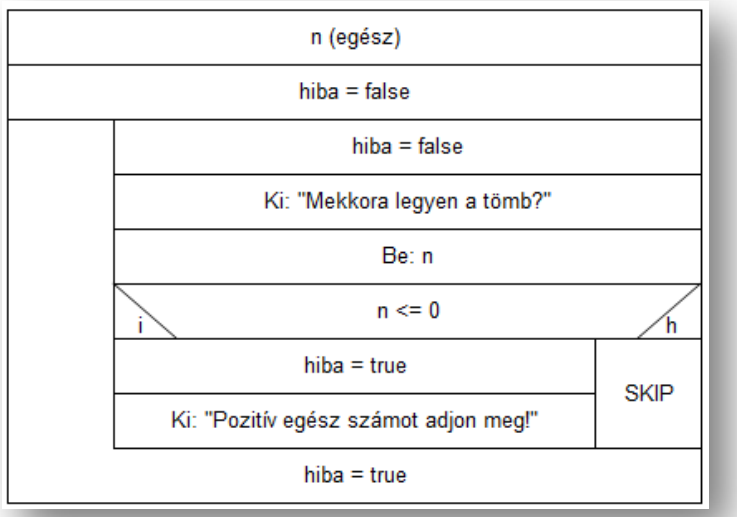
NegativeArraySizeException váltódik ki.

Kód:


```
int[] array1 = new int[-8];
System.out.println(array1);
```

</td>
<td>Hallott szakmai szöveg megértése</td>
<td>Számítógép
NetBeans IDE 7.3

Frontális munka</td>
</tr>
</table>

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
11 – 12’	Futási eredmény:  	Futási eredmény analizálása Fogalomalkotás	Magyarázat
12 – 15’	Az a) feladat tervezése:  	Algoritmikus gondolkodás Algoritmus tervezése Visszacsatolás Struktogram olvasása, értelmezése Gyakorlás	Megbeszélés Frontális munka Struktogramot ez után megkapják és ez alapján kódolnak Hálózat 1a_Struktogram_06_ora.png Egyéni munka

<table>
<tr><th>IDŐKERET</th><th>AZ ÓRA MENETE</th><th>CÉLOK ÉS FELADATOK</th><th>NEVELÉSI-OKTATÁSI STRATÉGIA
MEGJEGYZÉSEK</th></tr>
<tr>
<td>15 – 20’</td>
<td>Az a) feladat kódja:

<pre>
package gyakorlat13;

import java.util.Scanner;

public class Gyakorlat13_6 {
    public static void main(String[] args) {

        // 1.a) feladat
        Scanner billentyuzet = new Scanner(System.in);

        int n;
        boolean hiba = false;

        do{
            hiba = false;
            System.out.println("Mekkora legyen a tömb?");
            n = billentyuzet.nextInt();
            billentyuzet.nextLine();
            if(n <= 0) {
                hiba = true;
                System.out.println("Pozitív egész számot adjon meg!");
            }
        } while (hiba);

    }
}
</pre>
</td>
<td>Programkód írása
Szintaktika követése
Precizitás, pontosság

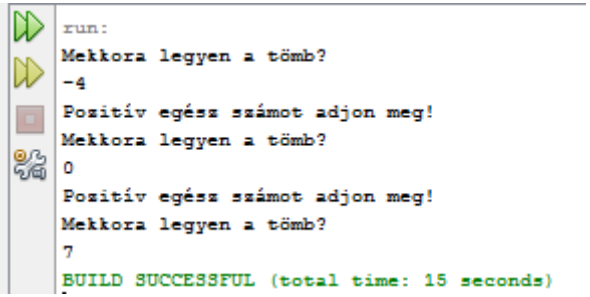
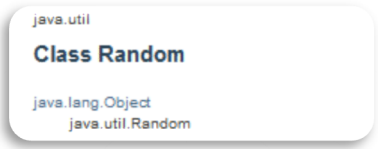
Önellenőrzés</td>
<td>Egyéni munka

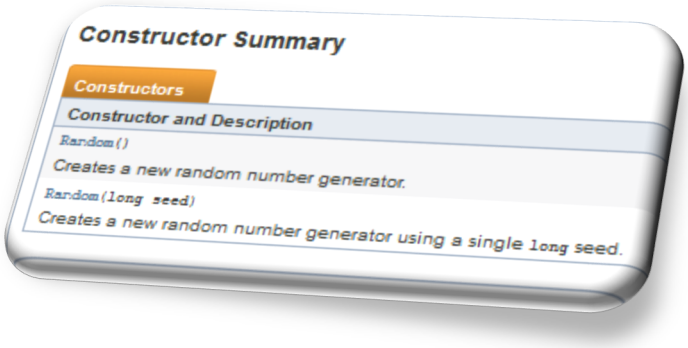
Számítógép
NetBeans IDE 7.3
Java 1.5

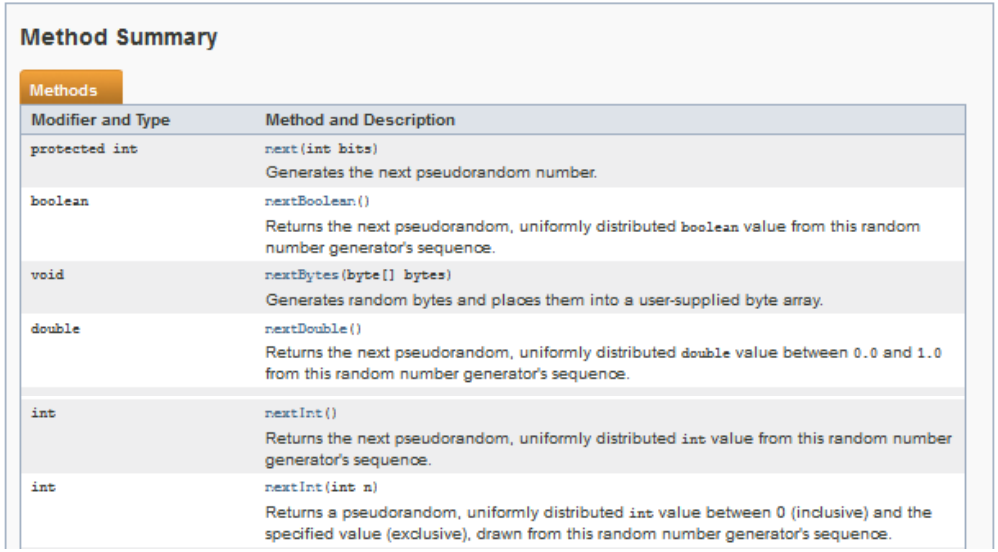
Követem munkájukat, ha szükséges segítek</td>
</tr>
<tr>
<td>20 – 21’</td>
<td>Az eddigi forráskód tesztelése:

negatív egész számmal, 0-val és pozitív egész számmal

valós szám esetén: InputMismatchException váltódik ki.</td>
<td>Tesztelés</td>
<td>Egyéni munka</td>
</tr>
</table>

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
21 – 22’		Diagnosztizálás, hiba-keresés	Egyéni munka
22 – 23’	Az b) feladat kapcsán: Eddig álvéletlen számokkal foglalkoztunk: Math osztály: random() metódus: [0,0; 1,0[intervallumban álvéletlen számot ad. De a Random osztály: véletlenszámok előállítására szolgáló osztály http://docs.oracle.com/javase/7/docs/api/java/util/Random.html util csomagban van → importálni kell 	Fokozatosság Visszacsatolás Fogalomalkotás Összehasonlítás Információ gyűjtése Következtetés	Frontális munka Tanulói elbeszélés Elbeszélés, közlés Internet http://docs.oracle.com/javase/7/docs/api/java/util/Random.html

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
23 – 25’	Random() konstruktor: készít egy új véletlenszám-generátort. Konstruktorok:  nextInt() metódus: A teljes int értéktartományban generál egy int típusú értéket. nextLong() metódus: A teljes long értéktartományban generál egy long típusú értéket. nextInt(int n) metódus: A [0; n[intervallumban generál egy int típusú értéket. nextDouble() metódus: A [0,0; 1,0] intervallumban generál egy double típusú értéket. nextBoolean() metódus: Generál egy logikai értéket.	Információ elemzése A tények elemzése Információ gyűjtése Új ismereteket megalapozó tények biztosítása	Beszélgetés Frontális munka Internet http://docs.oracle.com/javase/7/docs/api/java/util/Random.html

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
	Metódusok:  **Method Summary** Methods Modifier and Type — Method and Description protected int — next(int bits) — Generates the next pseudorandom number. boolean — nextBoolean() — Returns the next pseudorandom, uniformly distributed boolean value from this random number generator's sequence. void — nextBytes(byte[] bytes) — Generates random bytes and places them into a user-supplied byte array. double — nextDouble() — Returns the next pseudorandom, uniformly distributed double value between 0.0 and 1.0 from this random number generator's sequence. int — nextInt() — Returns the next pseudorandom, uniformly distributed int value from this random number generator's sequence. int — nextInt(int n) — Returns a pseudorandom, uniformly distributed int value between 0 (inclusive) and the specified value (exclusive), drawn from this random number generator's sequence.	Olvasott angol nyelvű szakmai szöveg megértése	Frontális munka Internet http://docs.oracle.com/javase/7/docs/api/java/util/Random.html Beszélgetés

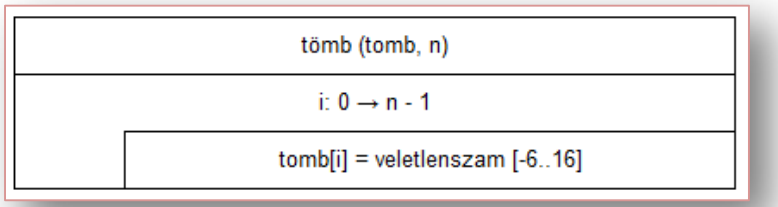
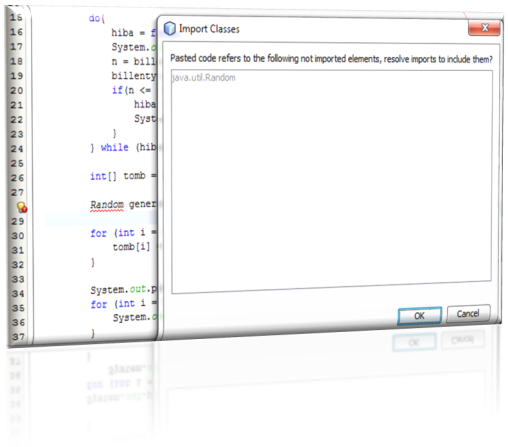
25 – 28'	Létrehozni saját hálózati meghajtón:
	Gyakorlat13 projekt/alapértelmezett csomag/VeletlenSzam.java
	Random objektum létrehozása
	Random osztály importálása
	`[-6;16]  16 - (-6) = 22`
	`nextInt(23) - 6`
	`nextDouble() * 22 - 6`
		Állománykezelés
		Alkalmazás
			Hálózat
			Frontális munka
			Magyarázat
			Tábla

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK

28 – 30’	Kód:
	
	`import java.util.Random;`
	
	`/**`
	` *`
	` * @author Kovács Henriette`
	` */`
	`public class VeletlenSzam {`
	`  public static void main(String[] args) {`
	`  Random generator = new Random();`
	
	`  // Egész véletlenszám generálása a [-6;16] intervallumban`
	`  int iszam = generator.nextInt(23) - 6;`
	`  System.out.println("Egész véletlenszám: " + iszam);`
	
	`  // Valós véletlenszám generálása a [-6;16] intervallumban`
	`  double dszam = generator.nextDouble() * 22 - 6;`
	`  System.out.println("Valós véletlenszám: " + dszam);`
	
	`  }`
	`}`
		Programkód írása
		Szintaktika követése
		Precizitás, pontosság
		
		Önellenőrzés
			Egyéni munka
			
			Számítógép
			NetBeans IDE 7.3
			Java 1.5

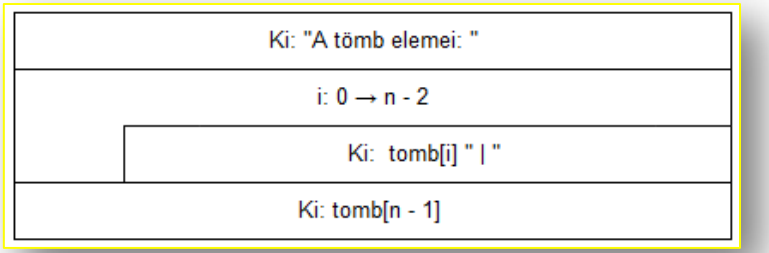
	Futási eredmény:
	
	`run:`
	`Egész véletlenszám: 7`
	`Valós véletlenszám: 13.17065317138843`
	`BUILD SUCCESSFUL (total time: 1 second)`
		Diagnosztizálás	Egyéni munka
			Számítógép
			NetBeans IDE 7.3
			Java 1.5

	A b) feladat struktogramja:		

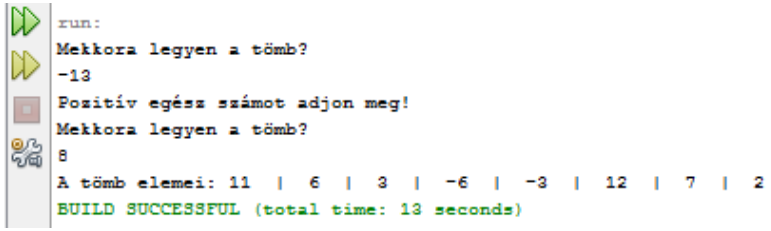
IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
30 – 31’	 A véletlenszámokat nem kell eltárolni külön változóba, egyből értékül lehet adni a tömb elemeinek.	Algoritmus alkotása	Hálózat 1b_Struktogram_06_ora.png Megbeszélés Frontális munka
31 – 32’	Ha a kódolás során elfeledkezik az importálásról. 	Figyelemfelhívás Olvasott angol nyelvű szöveg megértése	NetBeans IDE 7.3

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK

32 – 34’	A b) feladat kódja:
	
	`Random generator = new Random();`
	
	`for (int i = 0; i < n; ++i) {`
	`  tomb[i] = generator.nextInt(23) - 6;`
	`}`
		Programkód átalakí-tása
		Szintaktika követése
		Precizitás, pontosság
			Egyéni munka

34 – 35’	A c) feladat struktogramja: (feltételvizsgálatok száma!) 	Visszacsatolás	Hálózat 1c_Struktogram_06_ora.png Megbeszélés Frontális munka

35 – 38’	A c) feladat kódja:
	
	`System.out.print("A tömb elemei: ");`
	`for (int i = 0; i < n - 1; ++i) {`
	`  System.out.print(tomb[i] + "  |  ");`
	`}`
	`System.out.println(tomb[n - 1]);`
	
	`  }`
	`}`
		Programkód módosí-tása
		
		Szintaktika követése
		Precizitás, pontosság
			Egyéni munka
			
			NetBeans IDE 7.3

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
	Futási eredmény:  	Tesztelés	NetBeans IDE 7.3 Egyéni munka
38 – 40’	Gyűjtsük össze címszavakat felsorolva, hogy miket tanultunk ma, ebben a hat órában	Összefoglalás	Frontális munka
40 – 42’	1.) Írjon programot, melyben a) létrehoz egy valós számokból álló, a felhasználó által megadott elemszámú tömböt, b) feltölti a tömb elemeit olyan két tizedesjegyre kerekített véletlenszámokkal, melyek a [-50, 50] zárt intervallumba esnek, c) kiíratja a tömb elemeit úgy, hogy elemei egymástól a ’ ❄ ’ karakterrel legyenek elválasztva, de az utolsó elem után ne legyen ’ ❄ ’ karakter, d) kommentezi a forráskódot úgy, hogy az tükrözze, melyik kódrészletben mi történik!	A házi feladat kijelölése	Hálózat Gyakorlat13_6_HF.pdf Veletlenszam_tabla.png Az elkészített kód lesz az alapja a következő órának, az itt elkészített kódot fogják a diákok felhasználni. Ezt közlöm is velük. Tanári közlés

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
	A feladathoz készítsen **struktogramot**! Mentse *Gyakorlat13_6_HF_Vezeteknev_Keresztnev* néven! A forráskódot mentse *Gyakorlat13_6_HF_Vezeteknev_Keresztnev* néven! A program futási eredménye a következő formában jelenjen meg: `A tömb elemei: 6.39 ❄ 3.02 ❄ 19.95 ❄ -5.93 ❄ 34.58 ❄ 15.37`		
42 – 45’	Tanulók munkájának értékelése	Értékelés	Megbeszélés Egész órát végig kíséri az értékelés.

TÁBLAKÉP

```
[-6;16]    16 - (-6) = 22
```

```
nextInt(23)                 [0;23[

                            [0;22]      /-6

nextInt(23) - 6             [-6;16]
```

```
nextDouble()                [0;1]       /*22

nextDouble() * 22           [0;22]      /-6

nextDouble() * 22 - 6       [-6;16]
```

Feladatok

1.) Írjon programot,
   a) melyben létrehoz egy egész számokból álló, a felhasználó által megadott elemszámú tömböt,
   b) feltölti a tömb elemeit olyan véletlenszámokkal, melyek a [-6, 16] zárt intervallumba esnek,
   c) kiíratja a tömböt úgy, hogy elemei egymástól a ’|’ karakterrel legyenek elválasztva, de az utolsó elem után ne legyen ’|’ karakter!

Házi feladat

1.) Írjon programot, melyben

a) létrehoz egy valós számokból álló, a felhasználó által megadott elemszámú tömböt,

b) feltölti a tömb elemeit olyan két tizedesjegyre kerekített véletlenszámokkal, melyek a [-50, 50] zárt intervallumba esnek,

c) kiíratja a tömb elemeit úgy, hogy elemei egymástól a ' ❄ ' karakterrel legyenek elválasztva, de az utolsó elem után ne legyen ' ❄ ' karakter,

d) kommentezi a forráskódot úgy, hogy az tükrözze, melyik kódrészletben mi történik!

A feladathoz készítsen **struktogramot**! Mentse *Gyakorlat13_6_HF_Vezeteknev_Keresztnev* néven!

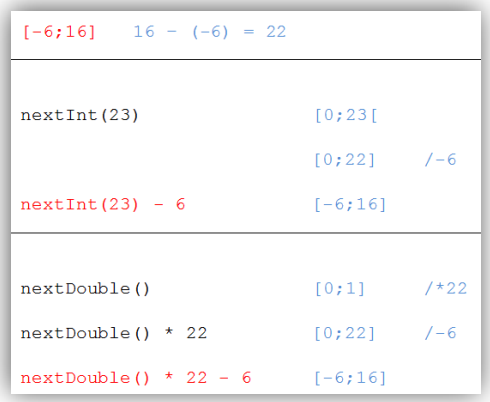
A forráskódot mentse *Gyakorlat13_6_HF_Vezeteknev_Keresztnev* néven!

A program futási eredménye a következő formában jelenjen meg:

```
A tömb elemei:   6.39 ❄ 3.02 ❄ 19.95 ❄ -5.93 ❄ 34.58 ❄ 15.37
```

Segítség a feladathoz:

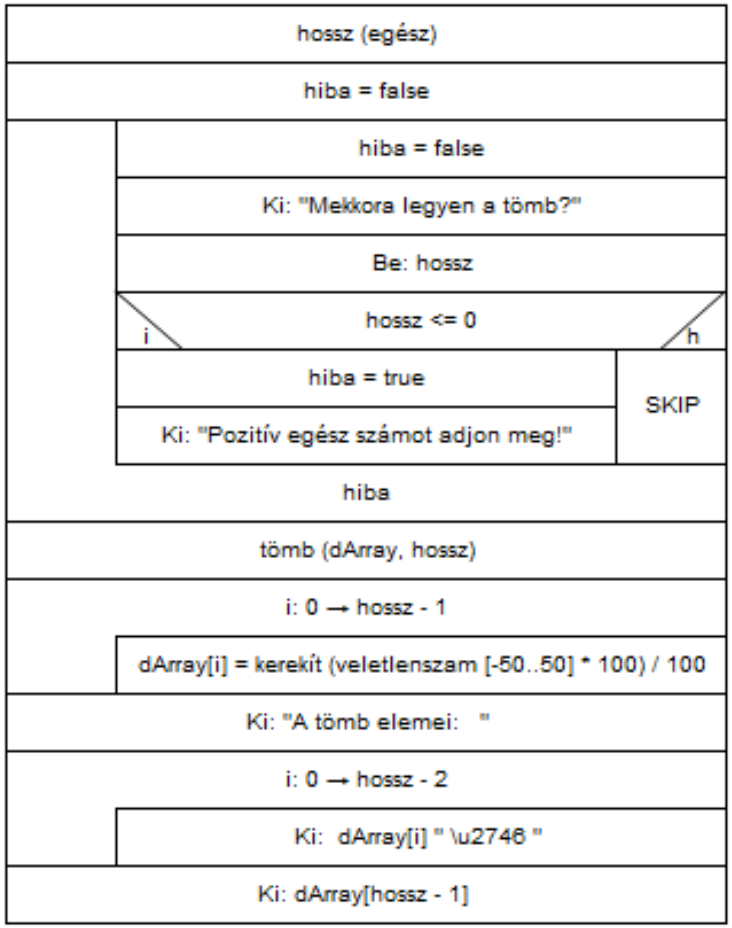
- Veletlenszam_tabla.png-ben is megtalálható az alábbi szemléltető kép a hálózaton, töltsék le, lehet, hogy a későbbiekben még szükség lesz rá!



- A Heavy chevron snowflake, ❄ karakter megtalálható a 3. órán használt unicode táblában.
  - ➢ http://unicode-table.com/en/
- Két tizedesjegyre kerekítés algoritmusa (ha már elfelejtették volna): nézzenek utána! ☺

Házi feladat megoldása

Struktogram:



Kód:

```
import java.util.random;
import java.util.Scanner;

/**
 *
 * @author kovács henriette
 */
public class Gyakorlat13_6_HF_Kovacs_Henriette {

    public static void main(String[] args) {

        Scanner billentyuzet = new Scanner(System.in);
        random generator = new random();

        // 1.a) feladat
        // tömb elemszámának bekérése

        int hossz;
        boolean hiba = false;

        do {
            hiba = false;
            System.out.println("mekkora legyen a tömb?");
            hossz = billentyuzet.nextInt();
            billentyuzet.nextLine();
            if (hossz <= 0) {
                hiba = true;
                System.out.println("pozitív egész számot adjon meg!");
            }
        } while (hiba);

        // tömb létrehozása
        double[] darray = new double[hossz];

        // 1.b) feladat
        // tömb elemeinek feltöltése véletlenszámmal
        for (int i = 0; i < hossz; i++) {
//            darray[i] = generator.nextDouble() * 100 - 50;  // véletlenszám
//            darray[i] = math.round(darray[i] * 100) / 100.0; // kerekítés

            darray[i] =
                    math.rint((generator.nextDouble() * 100 - 50) * 100) / 100;
        }

        // 1.c) feladat
        // tömb elemeinek kiíratása
        System.out.print("A tömb elemei:\t");
        for (int i = 0; i < hossz - 1; ++i) {
            System.out.print(darray[i] + " \u2746 ");
        }
        System.out.println(darray[hossz - 1]);

    }
}
```

Reflexió

Az üres String és üres tömb párhuzamba állítása az óra kezdetén nagyon hasznosnak bizonyult. Üres Stringgel már az előző órán is foglalkoztunk, sőt már többször előkerült, de az üres tömb még nem. Ennek megtanítása most azért volt aktuális, mert a felhasználó által megadott elemszámnál, - hiába létezik az üres tömb – nincs értelme nullát megadni. Ez úgy éreztem, kellőképpen tudatosult bennük. Az összehasonlítás pedig abban segített, hogy így már egy ismert fogalomhoz tudták kötni az új ismereteket és azzal, hogy kódírás által meg is tapasztalhatták ennek lényegét és nemcsak elméletben beszéltünk róla, sokkal jobban elraktározódott bennük.

A programozást, az algoritmusok tanítását nagymértékben motiválja a véletlenszám generálása. Ez most is így volt és már az előző órákon is szóba került, úgyhogy kellőképpen „fel voltak csigázva" a diákok. Már foglalkoztunk ezzel a témával, de csak a Math osztályon keresztül, amiről most megtudták a tanulók – nagy meglepetésükre - hogy csak álvéletlen számot ad. Ezek után még jobban érdekelte őket, hogy hogyan is lehet akkor véletlenszámokat előállítani. A táblaképet nagyon tudatosan megterveztem, mert az eddigi tapasztalataim alapján tudtam, hogy ez problémát okozhat és a forráskód írása során nem szerettem volna, hogy gondot jelentsen. A tábláról szinte le tudták másolni az erre vonatkozó kódrészletet és mivel még külön file-ban is lekódoltuk, a rögzítés, megerősítés, elmélyítés szempontjából ez nagyon hasznos volt. A külön file-ban történő forráskód írásának emellett részemről az volt még a célja, ha bármikor előkerül véletlenszám generálás, akkor szükség esetén könnyen megtalálják.

Nagyon örültem, hogy végre adhattam házi feladatot is, amit a diákok mindig vasárnap éjfélig (keddenként van hat óránk) adnak be nekem. Természetesen, ha valami kérdésük van, vagy valami miatt nem fut a kódjuk, akkor is leadják, én reagálok rá, segítek nekik és ez alapján javítják. Ha nem adják le határidőre a feladatot, ha egy kicsit csúsznak, ennek még soha nem volt semmilyen következménye, mert nagyon kedvesek, ilyenkor elnézést kérnek és általában meg is indokolják a késés okát, de nem fordult elő sűrűn. Egyszer volt, hogy reggel kaptam házi feladatot, az órák előtt másfél órával, de megnéztem és az illető meglepődésére még vissza is jeleztem ☺. A házi feladat olyan szempontból is nagyon fontos, hogy a következő óra ennek a feladatnak a

folytatása lesz, az itt létrehozott tömbbel fogunk tovább dolgozni. Ha valaki valamilyen oknál fogva nem készít házi feladatot, annak a teljes megoldás elérhető lesz a hálózaton.

A feladat alapjaiban az előző öt órában megtanult ismeretek begyakorlását, megerősítését, rögzítését, tudatosítását, megszilárdítását, beidegződését, rutinná válását, valamint a diákok kommunikációs, rendszerező, tesztelő képességének, logikus és algoritmikus gondolkodásának fejlesztését szolgálta.

ÓRATERVEZET - ALAPVETŐ ADATGYŰJTÉSI ÉS FELDOLGOZÁSI ALGORITMUSOK

A pedagógus neve: Kovács Henriette

A szakmacsoport száma és megnevezése: 7. Informatika

A szakképesítés azonosító száma: 54 213 05

A szakképesítés megnevezése: Szoftverfejlesztő

Iskolai rendszerű szakképzés (a szakképzési évfolyamok száma: 2)

Szakmai követelménymodul: 10817-12 Hálózatok, programozás és adatbázis-kezelés

Tantárgy: Adatbázis- és szoftverfejlesztés gyakorlat

Programozási nyelv: Java

Fejlesztői környezet: NetBeans IDE 7.3

Osztály - csoport: I/13. S - SZ2

Az óra tananyaga: Alapvető adatgyűjtési és feldolgozási algoritmusok

Az óra cél- és feladatrendszere

A fejlesztendő attitűd, készségek, képességek, a tanítandó ismeretek (fogalmak, szabályok stb.) és az elérendő fejlesztési szint, tudásszint megnevezése:

- A tanuló tartsa be a számítógépes munka szabályait, különös tekintettel a balesetek megelőzésére.
- Felkelteni és folyamatosan ébren tartani a tanulók érdeklődését a szoftverfejlesztés iránt. Kialakítani a tanulókban az informatikai ismereteik folyamatos bővítésének és megújításának igényét.
- Korszerű alkalmazói készség kialakítása.
- Az algoritmikus gondolkodás fejlesztése: a tanulók önálló, rendszerezett, logikus gondolkodásának fejlesztése; a logikus gondolkodás lényeges a problémamegoldásban, és az algoritmusok elkészítésében, kódolásában.
- A feladatmegoldáshoz megfelelő adattípus kiválasztásának fejlesztése.
- Legyen képes a NetBeans fejlesztői környezet kényelmi funkcióinak készségszintű használatára.
- Önálló munkára nevelés és felzárkóztatás.
- Alkotó munkára nevelés.
- Az esztétikai készség fejlesztése: igény és készség a számítógépes produktum esztétikus formájának kialakítására, a program helyes strukturálására.
- *Tudjon a programban karakterek unicode-jával dolgozni, érdekes karaktereket unicode-juk felhasználásával megjeleníteni*
- *Tudjon a NetBeans IDE 7.3 –ban állományokat másik projektbe, másik csomagba, más néven átmásolni.*
- *Tudjon tömb típusú változót deklarálni, tömböt létrehozni, feltölteni.*
- *Tudjon tömbelemeket megjeleníteni.*
- *Tudjon explicit konverziót alkalmazni.*
- *Tudjon algoritmust, programkódot megvalósítani, felhasználni, átalakítani, módosítani.*
- *Legyen képes a tömbök elemeit elérni, feldolgozni és velük műveleteket végezni.*
- *Legyen képes tesztelni.*

Fejlesztendő készségek, képességek

- Visszaemlékezés, esztétikai érzék, állománykezelés, alkalmazás, megfigyelés, programkód átalakítása, módosítása, diagnosztizálás, hibakeresés, precizitás, pontosság, algoritmikus gondolkodás, algoritmus felismerése, megértése, algoritmus tervezése, algoritmus alkotása, struktogram készítése, programkód írása, szintaktika követése, kreativitás, tesztelés, programkód javítása, képletalkotás,

típusválasztás, futási eredmény analizálása, problémamegoldás sokszínűsége, következtetés, futási eredmény átláthatósága, követhetősége, struktogram átalakítása, rögzítés, ellenőrzés, megerősítés, összehasonlítás, módosítás, strukturáltság.

Fejlesztendő attitűd

- Legyen képes: számítógépet kezelni, szoftvereket használni.
- Tudja önállóan használni a hálózatot és annak alapszolgáltatásait.
- Legyen képes az adott probléma megoldásához kiválasztani az általa ismert módszerek, eszközök és alkalmazások közül a megfelelőt.
- Ismerje fel és legyen képes különféle formákban megfogalmazni a környezetében előforduló tevékenységek algoritmizálható részleteit.
- Legyen képes a problémamegoldáshoz szükséges adatok és az eredmény kapcsolatának megtervezésére, értelmezésére.
- Adott feladat megoldásához legyen képes algoritmusokat megfogalmazni, tervezni és megvalósítani.
- Legyen képes alkalmazást (szoftvert) tervezni, fejleszteni, futtatni, tesztelni és magyarázó megjegyzésekkel ellátni, előkészítve ezzel a dokumentáció elkészítését.
- Legyen képes programozási feladatot ellátni, egy szoftvert elkészíteni (kódolni), Java programnyelven implementálni.
- Legyen képes értelmezni és javítani a NetBeans által megjelenített hibaüzeneteket.
- Legyen képes programjait úgy megírni, hogy azt később Ő maga is, vagy tanulótársa is fel tudja használni.
- Legyen képes alapvető adatgyűjtési és feldolgozási algoritmusokat (összegzés, számlálás) feladatmegoldásban alkalmazni.

Az óra didaktikai feladatai

- Ellenőrzés
- Motiválás
- Ráhangolódás
- Aktivizálás
- Összehasonlítás
- Összefüggések keresése
- Feladat kitűzése
- Rögzítés
- Figyelemfelhívás
- Figyelem felkeltése
- Aktivizálás
- Ismétlés
- Ellenőrzés
- Előzetes ismeretek felidézése
- Megerősítés
- Rendszerezés
- Értékelés

Tantárgyi kapcsolatok: Adatbázis- és szoftverfejlesztés elmélet, Matematika, Angol, Hálózati ismeretek I., Informatikai alapismeretek

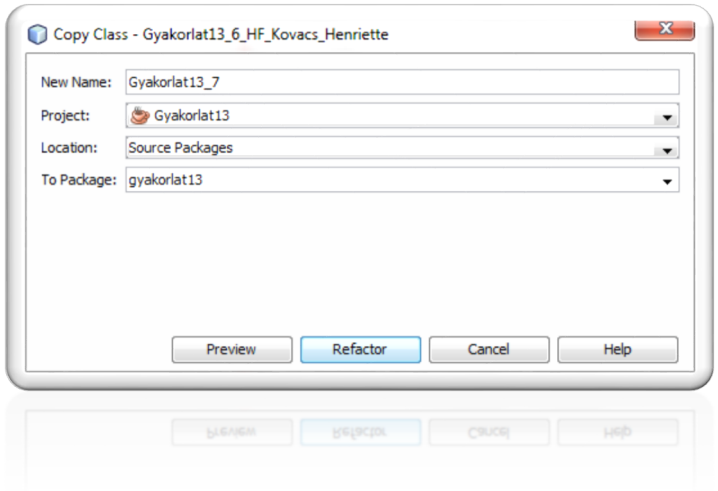
Felhasznált források (tankönyv, munkafüzet, feladat- és szöveggyűjtemény, digitális tananyag, online források, szakirodalom stb.):

Szerző(k)	Cím	Kiadó	ISBN kód
Vég Csaba – dr. Juhász István	Java – start!	Logos 2000	963 03 9005 1
Vég Csaba	Instant Java/Java EE/SOA I-II	Logos 2000	978-963-06-2601-9 978-963-06-2602-6
Dirk Louis – Peter Müller:	Java 5 Belépés a programozás világába	Panem	963 545 454 6
Benkő Tiborné, Tóth Bertalan	Együtt könnyebb a programozás Java	Computerbooks	963 618 337 6
Rogers Cadenhead	Tanuljuk meg a Java programozási nyelvet 24 óra alatt	Kiskapu Kft.	963 9637 07 6
Angster Erzsébet	Objektumorientált tervezés és programozás Java 1 - 2	4KÖR Bt.	963 00 6262 3 963 00 6263 1
Peter J. DePasquale	Java zsebkönyv	Kiskapu Kft.	963 9637 10 6
Nyékyné Gaizler Judit, Frohner Ákos, Kozsik Tamás (Szerkesztők)	Java 2 útikalauz programozóknak	ELTE TTK Hallgatói Alapítvány	963 463 305 6
Kathy Sierra, Bert Bates	Agyhullám: Java	Kiskapu Kft.	978 963 9637 79 5

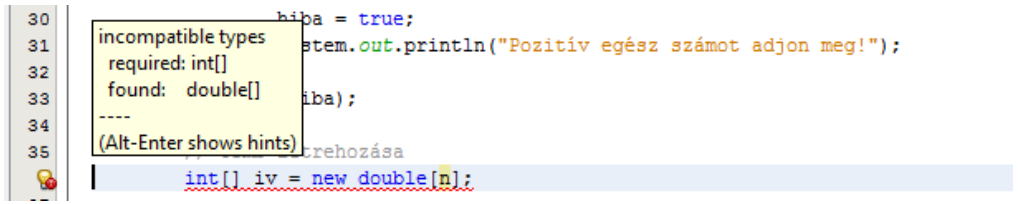
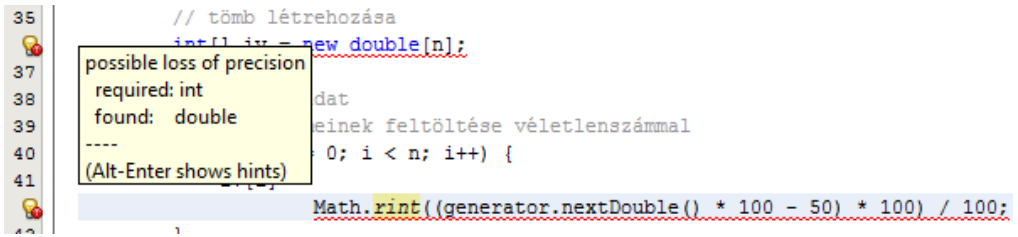
http://docs.oracle.com/javase/7/docs/api/

Dátum: éééé. hh. nn.

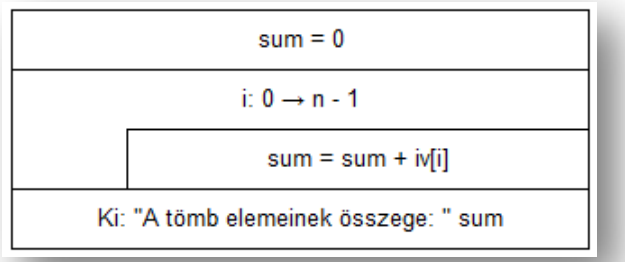
IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MÓDSZEREK TANULÓI MUNKAFORMÁK ESZKÖZÖK MEGJEGYZÉSEK
0 – 5’	Tanulók üdvözlése ☺ A házi feladat kódját szókártyákra vágtam, összekevertem. A szókártyákat a megfelelő sorrendbe kell rakniuk, vagy a számozás alapján a helyes sorrendet fel kell állítaniuk. (organikus okok) Gépek bekapcsolása. Házi feladat megnyitása, a benne előforduló karakterről, (Heavy chevron snowflake - Vastag ék alakú rangjelzés hópihe) néhány érdekes momentum: A karakter és a rendfokozat (Főtörzszászlós) jelzése közötti azonosság.   http://hu.wikipedia.org/wiki/Rendfokozat#mediaviewer/F%C3%A1jl:Rank%20Army%20Hungary%20OR-09.svg http://hu.wikipedia.org/wiki/Rendfokozat# http://unicode-table.com/en/	Ellenőrzés Motiválás Visszaemlékezés Ráhangolódás Aktivizálás Esztétikai érzék Összehasonlítás Összefüggések keresése	Házi feladat ellenőrzése már elektronikusan megtörtént, a hibák ki vannak javítva, de hogy lássam, valóban tudják-e az eddigieket egy gyors ellenőrzési módszert választottam. Szókártyák Egyéni munka Számítógép Tanári elbeszélés, bemutatás „A részletekben rejlik a lényeg.” Frontális munka Hálózat Internet

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
5 – 7’	A Gyakorlat13_6_HF_Vezeteknev_Keresztnev file másolása ugyanezen projekt gyakorlat13 csomagjába Gyakorlat13_7 néven! Jobb klikk a file nevén/Refactor/Copy… és a megjelenő párbeszédablak kitöltése: 	Állománykezelés Alkalmazás Megfigyelés	Magyarázat Szemléltetés Irányított, önálló munka Tanári bemutatás Projektor Számítógép NetBeans IDE 7.3
7 – 8’	Feladat: 1. Változtassa meg a kódot! A tömb legyen int elemtípusú, a neve legyen iv, elemszáma legyen n. További szükséges változtatások:	Programkód átalakítása, módosítása	Egyéni munka
8 – 10’	Visszajelzések a fejlesztői környezettől:		

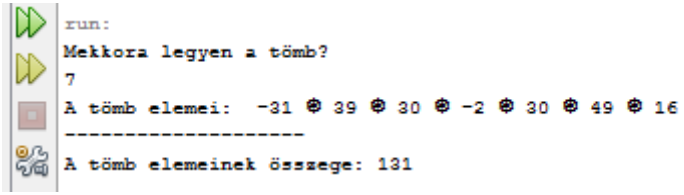
IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK

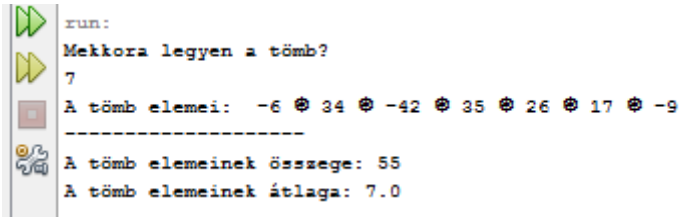
	Egyrészt:
	
	`new double[n]` →
	`new int[n]`
	Másrészt: nem kell kerekíteni
	
	`Math.rint((generator.nextDouble() * 100 - 50) * 100) / 100` →
	`generator.nextInt(101) - 50;`
		Diagnosztizálás, hiba-keresés
		Programkód átalakí-tása, módosítása
			Egyéni munka
			ha nem megy →
			Megbeszélés
			Irányított, önálló munka
			Projektor
			Számítógép
			NetBeans IDE 7.3

	Feladat sorszámát jelző kommentek javítása.	Precizitás, pontosság	Egyéni munka

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
10 – 11’	Feladat: 2. Határozza meg a tömb elemeinek a) összegét, b) átlagát!	Feladat kitűzése Algoritmus felismerése, megértése	Tanári közlés Projektor Hálózat
11 – 15’	Összeg: Struktogram tervezése, létrehozása: 	Algoritmikus gondolkodás Kreativitás Algoritmus tervezése Algoritmus alkotása Struktogram készítése	Beszélgetés Magyarázat Irányított önálló munka Számítógép Structorizer Microsoft Office Excel 2007, 2010

15 – 17’	Kód elkészítése:
	`// 2. feladat`
	`// tömb elemeinek összege`
	`int sum = 0;`
	`for (int i = 0; i < n; ++i)`
	`  sum += iv[i];`
	`System.out.println("A tömb elemeinek összege: " + sum);`
		Programkód írása
		Precizitás, pontosság
		Szintaktika követése
		Diagnosztizálás
			Számítógép
			NetBeans IDE 7.3
			Java 1.5
			Egyéni munka

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
	Futási eredmény:  run: Mekkora legyen a tömb? 7 A tömb elemei: -31 ❁ 39 ❁ 30 ❁ -2 ❁ 30 ❁ 49 ❁ 16 --- A tömb elemeinek összege: 131	Tesztelés Hibakeresés Programkód javítása, módosítása Esztétikai érzék fejlesztése	Egyéni munka NetBeans IDE 7.3 Számítógép Projektor

17 – 19’	Átlag:
	átlag = elemek összege / elemek száma
	Az átlag valós szám!
	Kód írása
	`// tömb elemeinek átlaga`
	`double atlag = sum / n;`
	`System.out.println("A tömb elemeinek átlaga: " + atlag);`
	Futási eredmény:
	
	run:
	Mekkora legyen a tömb?
	7
	A tömb elemei: -6 ❁ 34 ❁ -42 ❁ 35 ❁ 26 ❁ 17 ❁ -9
	--- 	
	A tömb elemeinek összege: 55
	A tömb elemeinek átlaga: 7.0
		Képletalkotás
		Rögzítés
		Figyelemfelhívás
		Típusválasztás
		Programkód írása
		Futási eredmény analizálása
		Diagnosztizálás
		Önellenőrzés
			Megbeszélés
			Frontális munka
			Java 1.5
			NetBeans IDE 7.3
			Számítógép
			Projektor
			Egyéni munka

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK

19 – 23’	Nem jó! Egész osztás eredménye kerül a double típusú változóba! Valós osztást kell alkalmazni! → Típuskényszerítés: Explicit, bővítő konverzió
	
	Kód javítása:
	
	`// tömb elemeinek átlaga`
	`double atlag = (double)sum / n;`
	`System.out.println("A tömb elemeinek átlaga: " + atlag);`
	
	Futási eredmény: Mivel érdemes tesztelni?
	
	`run:`
	`Mekkora legyen a tömb?`
	`7`
	`A tömb elemei:  28 48 18 -33 8 18 -20`
	`---`
	`A tömb elemeinek összege: 67`
	`A tömb elemeinek átlaga: 9.571428571428571`
	
	Kód átalakítása, hogy az átlag két tizedesjegy pontossággal jelenjen meg!
	
	Kódírási lehetőségek:
	
	`// tömb elemeinek átlaga`
	`double atlag = (double)sum / n;`
	`System.out.println("A tömb elemeinek átlaga: "`
	`+ Math.rint(atlag * 100) / 100);`
	
	vagy:
		Figyelem felkeltése
		Aktivizálás
		Ismétlés
		Következtetés
		
		Alkalmazás
		Programkód javítása
		
		Tesztelés
		
		Feladatkitűzés
		Esztétikai érzék fejlesztése
		
		Programkód átalakítása
			Irányított, önálló munka
			Megbeszélés
			
			Java 1.5
			NetBeans IDE 7.3
			Számítógép
			Projektor
			
			Beszélgetés
			Egyéni munka

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK

	`// tömb elemeinek átlaga`
	`double atlag = (double)sum / n;`
	`System.out.println("A tömb elemeinek átlaga: "`
	`+ Math.round(atlag * 100) / 100.0);`
	
	Futási eredmény:
	
	`run:`
	`Mekkora legyen a tömb?`
	`7`
	`A tömb elemei: 48 17 -50 -27 -9 -13 -44`
	`---`
	`A tömb elemeinek összege: -78`
	`A tömb elemeinek átlaga: -11.14`
		Problémamegoldás sokszínűsége
		
		Futási eredmény átlát-hatósága, követhető-sége
		Ellenőrzés
			Megbeszélés
			Frontális munka
			
			NetBeans IDE 7.3
			Egyéni munka

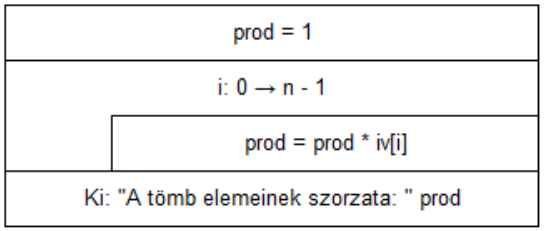
23 – 24’	Feladat: 2. Határozza meg a tömb elemeinek c) négyzetösszegét, módosítsa az összegnél készített struktogramot, mentse Gyakorlat13_7_2c néven! Struktogram: sum = 0 i: 0 → n - 1 sum = sum + (iv[i])² Ki: "A tömb elemeinek négyzetösszege: " sum	Struktogram átalakí-tása Állománykezelés Ellenőrzés	Egyéni munka Java 1.5 NetBeans IDE 7.3 Számítógép Projektor

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK

24 – 25’	Kód másolása, átalakítása:
	
	`// tömb elemeinek négyzetösszege`
	`sum = 0;`
	`for (int i = 0; i < n; ++i) {`
	`  sum += Math.pow(iv[i], 2);`
	`}`
	`System.out.println("A tömb elemeinek négyzetösszege: " + sum);`
	
	Futási eredmény.
	
	`run:`
	`Mekkora legyen a tömb?`
	`7`
	`A tömb elemei:  21 -24 12 3 -31 7 30`
	`--------------------`
	`A tömb elemeinek összege: 18`
	`A tömb elemeinek átlaga: 2.57`
	`--------------------`
	`A tömb elemeinek négyzetösszege: 3080`
	`--------------------`
		Programkód átalakítása
		Alkalmazás
		
		Rögzítés
		Megerősítés
		
		Önellenőrzés
		
		Ellenőrzés
			Egyéni munka
			
			Java 1.5
			NetBeans IDE 7.3
			Számítógép
			
			Projektor

25 – 26’	Feladat: 3. Határozza meg az iv tömb elemeinek szorzatát, készítsen hozzá struktogramot, melyet mentsen Gyakorlat13_7_3 néven! Miben különbözik az összeg és a szorzat meghatározása? Az előző struktogram és kód felhasználása Kezdeti érték beállítása! Szükséges változtatások	Összehasonlítás Módosítás	Megbeszélés Magyarázat Frontális munka

<table>
<tr><th>Időkeret</th><th>Az óra menete</th><th>Célok és feladatok</th><th>Nevelési-oktatási stratégia
Megjegyzések</th></tr>
<tr>
<td>26 – 28’

28 – 30’</td>
<td>Struktogram másolása, módosítása:


Kód másolása, átalakítása:

```
// 3. feladat
// tömb elemeinek szorzata
long prod = 1;
for (int i = 0; i < n; ++i) {
    prod  *= iv[i];
}
System.out.println("A tömb elemeinek szorzata: " + prod);
```

Futási eredmény:

```
A tömb elemeinek szorzata: -3836250
```
</td>
<td>Állománykezelés

Struktogram átalakítása

Programkód átalakítása

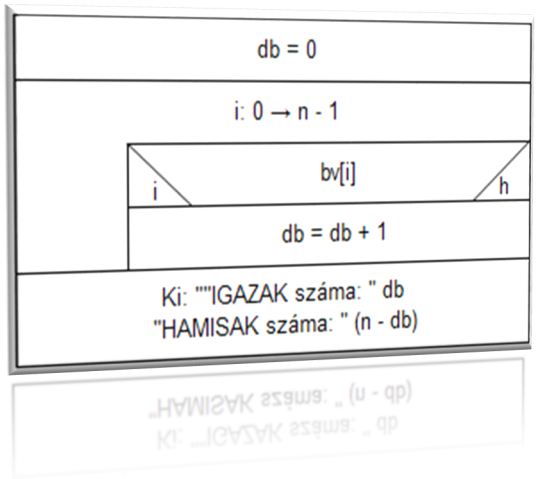
Önellenőrzés → Hibajavítás</td>
<td>Egyéni munka
Java 1.5
NetBeans IDE 7.3
Számítógép

Folyamatosan követem munkájukat, igény szerint segítek.</td>
</tr>
<tr>
<td>30 – 31’</td>
<td>Feladat: 4. Írjon programot,
a) melyben létrehoz egy logikai értékekből álló, szintén n elemszámú bv nevű tömböt,
b) feltölti a tömb elemeit véletlen logikai értékekkel,
c) kiíratja a tömb elemeit tetszőleges elválasztó karakterrel,</td>
<td>Feladatkitűzés
Előzetes ismeretek felidézése
Alkalmazás</td>
<td>Megbeszélés
Frontális munka
Projektor</td>
</tr>
</table>

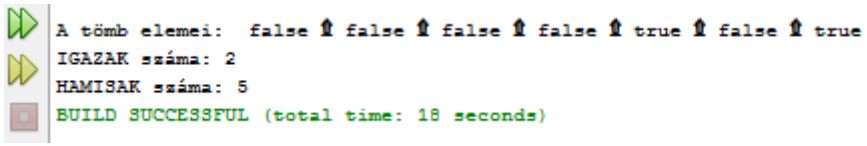
IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK

27 – 31’	Kód írása:
	
	`// 4. feladat`
	`// tömb létrehozása`
	`boolean[] bv = new boolean[n];`
	
	`// tömb elemeinek feltöltése véletlen logikai értékkel`
	`for (int i = 0; i < n; i++) {`
	`bv[i] = generator.nextBoolean();`
	`}`
	
	`// tömb elemeinek kiíratása`
	`System.out.print("A tömb elemei:\t");`
	`for (int i = 0; i < n - 1; ++i) {`
	`System.out.print(bv[i] + " \u06E9 ");`
	`}`
	`System.out.println(bv[n - 1]);`
	
	Futási eredmény:
	
	
	A tömb elemei: false ⇧ true ⇧ true ⇧ false ⇧ true ⇧ false ⇧ true
		Programkód írása
		Szintaktika követése
		Precizitás, pontosság
		Strukturáltság
		
		Alkotó képesség
		Önellenőrzés
		Esztétikai érzék
			Egyéni munka
			
			Számítógép
			NetBeans IDE 7.3
			Java 1.5

31 – 32’	Feladat: d) meghatározza a true és false értékek darabszámát! Ehhez a feladatrészhez készítsen algoritmust!	Algoritmikus gondol-kodás Algoritmus felisme-rése, megértése Algoritmus alkotása	Tanulói elbeszélés Beszélgetés Frontális munka

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
32 – 37’	Struktogram elkészítése:  	Algoritmus megvalósítása Struktogram készítése	Frontális munka Structorizer Microsoft Office Excel 2007, 2010

37 – 41’	Kód írása:
	`// true és false elemek darabszámának megszámolása`
	`int db = 0;`
	`for (int i = 0; i < n; i++) {`
	`if (bv[i]) {`
	`db++;`
	`}`
	`}`
	`System.out.println("IGAZAK száma: " + db`
	`+ "\nHAMISAK száma: " + (n - db));`
	`}`
	`}`
		Programkód készítése
		Szintaktika követése
		Strukturáltság
		Esztétikai érzék
			Egyéni munka
			NetBeans IDE 7.3
			Java 1.5

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
41 – 42’	Futási eredmény  	Önellenőrzés → Hibajavítás	NetBeans IDE 7.3 Egyéni munka
42 – 44’	Új algoritmusok: Sorozatszámítás Megszámolás	Ismétlés Rögzítés Megerősítés Rendszerezés	Megbeszélés Frontális munka
44 – 45’	Értékelés	Értékelés	Beszélgetés, elbeszélés Frontális munka

Feladatok

1. A Gyakorlat13_6_HF_Vezeteknev_Keresztnev file-t másolja ugyanezen projekt gyakorlat13 csomagjába Gyakorlat13_7 néven!
   Változtassa meg a kódot! A tömb legyen int elemtípusú, a neve legyen iv, elemszáma legyen n.
2. Határozza meg az iv tömb elemeinek
   a) összegét,
      készítsen hozzá struktogramot, melyet mentsen Gyakorlat13_7_2a néven,
   b) átlagát,
   c) négyzetösszegét,
      módosítsa az összegnél készített struktogramot, mentse Gyakorlat13_7_2c néven!
3. Határozza meg az iv tömb elemeinek szorzatát,
   készítsen hozzá struktogramot, melyet mentsen Gyakorlat13_7_3 néven!
4. Írjon programot,
   a) melyben létrehoz egy logikai értékekből álló, szintén n elemszámú bv nevű tömböt,
   b) feltölti a tömb elemeit véletlen logikai értékekkel,
   c) kiíratja a tömb elemeit tetszőleges elválasztó karakterrel,
   d) meghatározza a true és false értékek darabszámát,
      ehhez a feladatrészhez készítsen algoritmus, melyet mentsen Gyakorlat13_7_4 néven!

Reflexió

A házi feladat ellenőrzése részemről mindig elektronikus úton történik, meghatározott időre adják be a feladatokat a diákok, amiket ha kell többszörös levélváltás után javítanak. A feladat kapcsán ismét előkerült a unicode-tábla, mely alkalmas volt annak megláttatására, hogy az apró részletekben is milyen mély dolgok rejlenek, hogy milyen összefüggések fedezhetők fel még látszólag teljesen különböző objektumok között is. Ki gondolná, hogy egy karakter és egy rendfokozat jelzése között is találunk kapcsolatot. Ezek után a diákok már kellőképpen motiváltak voltak és kezdhettük a feladatokat.

A programozás tanítása során nagyon fontosnak tartom már megírt kódrészletek ismételt felhasználását, alkalmazását, annak szükség esetén megfelelő átalakítását, módosítását, javítását. Tulajdonképpen, amikor beépített metódusokat használunk, felülírunk, munkánk alapja egy már megírt kód, tehát használjuk, amit már megírtak. Mindig arra ösztönzöm diákjaimat, hogy ami már elkészült azt ne akarjuk még egyszer megvalósítani (ha tüzet akarunk gyújtani, gyufát keressünk és ne követ ☺). Ez a szemlélet az egész órát végig kísérte, tulajdonképpen a kezdetektől fogva, amikor még a házi feladatot is felhasználtuk további munkánkhoz, és az első feladatban leírt átalakításokat végrehajtva dolgoztunk tovább. Ezáltal fejlesztettem többek között a diákok logikus és algoritmikus gondolkodását, a precizitást, a pontosságot, a szintaktika követését, a diagnosztizálást, a hibakeresést, az önellenőrzést.

Az új ismereteket a gyakorlásba úgy építettem be, hogy tulajdonképpen szinte észre sem vették a diákok, hogy most már a gyakorlatban is alkalmaztak két programozási tételt, a sorozatszámítást (már Stringekkel kapcsolatban is előkerült) és a megszámolást, ezáltal így tömb elemeivel különböző műveleteket tudtak végezni, mint például összeadást, szorzást és hatványozást. Az óra végi összefoglalásban mindezt tudatosítottam, azáltal, hogy nevén neveztem a dolgokat, így látták a párhuzamot az elmélet és a gyakorlat között. Meg is jegyezték, hogy elméletben mindez sokkal bonyolultabbnak tűnt.

Az utolsó feladat nagyon jó választásnak bizonyult, a logikai értékekből álló tömb egy kicsit feldobta azt a tényt, hogy nagyrészt csak szám típusú tömbökkel dolgoztunk és mellesleg tökéletesen alkalmas volt a megszámolás algoritmusának elsajátítására.

Nagyon sok feladat megoldására sor került, nem voltam benne biztos, hogy lesz idő a negyedik feladat megvalósítására. Ebben az esetben az a), b), c) feladatok forráskódját a hálózatról töltötték volna le, a kódot elemeztük volna és a d) feladat megoldására került volna „csak” sor.

Az órán megvalósítottam azokat a dolgokat, amiket célul tűztem ki magam elé.

ÓRATERVEZET - MAXIMUMKIVÁLASZTÁS. TÖMBÖK ELEMEIN VÉGZETT MŰVELETEK.

A pedagógus neve: Kovács Henriette

A szakmacsoport száma és megnevezése: 7. Informatika

A szakképesítés azonosító száma: 54 213 05

A szakképesítés megnevezése: Szoftverfejlesztő

Iskolai rendszerű szakképzés (a szakképzési évfolyamok száma: 2)

Szakmai követelménymodul: 10817-12 Hálózatok, programozás és adatbázis-kezelés

Tantárgy: Adatbázis- és szoftverfejlesztés gyakorlat

Programozási nyelv: Java

Fejlesztői környezet: NetBeans IDE 7.3

Osztály - csoport: I/13. S - SZ2

Az óra tananyaga: Maximumkiválasztás. Tömbök elemein végzett műveletek.

Az óra cél- és feladatrendszere

A fejlesztendő attitűd, készségek, képességek, a tanítandó ismeretek (fogalmak, szabályok stb.) és az elérendő fejlesztési szint, tudásszint megnevezése:

- A tanuló tartsa be a számítógépes munka szabályait, különös tekintettel a balesetek megelőzésére.
- Felkelteni és folyamatosan ébren tartani a tanulók érdeklődését a szoftverfejlesztés iránt.
- Korszerű alkalmazói készség kialakítása.
- Az algoritmikus gondolkodás fejlesztése: a tanulók önálló, rendszerezett, logikus gondolkodásának fejlesztése; a logikus gondolkodás lényeges a problémamegoldásban, és az algoritmusok elkészítésében, kódolásában.
- Legyen képes a NetBeans fejlesztői környezet kényelmi funkcióinak készségszintű használatára.
- Önálló munkára nevelés és felzárkóztatás.
- Alkotó munkára nevelés: a végeredmény egy új termék, egy program lesz.
- Az esztétikai készség fejlesztése: igény és készség a számítógépes produktum esztétikus formájának kialakítására, a program helyes strukturálására.
- *Tudjon tömb típusú változót deklarálni, létrehozni, feltölteni.*
- *Tudjon tömböt megjeleníteni.*
- *Legyen képes algoritmust átalakítani, módosítani, javítani.*
- *Legyen képes forráskódot átalakítani, módosítani, javítani.*
- *Legyen képes algoritmizálni.*
- *Legyen képes a tömbök elemeit elérni, feldolgozni és velük műveleteket végezni.*
- *Tudjon tömb elemei közül maximumot kiválasztani.*
- *Tudjon algoritmust, programkódot megvalósítani, felhasználni, átalakítani, módosítani.*
- *Legyen képes tesztelni.*

Fejlesztendő készségek, képességek

- Algoritmikus gondolkodás, algoritmus felismerése, algoritmus megértése, algoritmizálás, algoritmus elemzése, algoritmus alkotása, visszaemlékezés, struktogram készítése, önellenőrzés, algoritmus megvalósítása, programkód átalakítása, módosítása, javítása, tesztelés, programkód írása, logikus gondolkodás, problémamegoldás, algoritmus módosítása, átalakítása, elemzés, fokozatosság, összehasonlítás, precizitás, pontosság, összefüggések

felismerése, algoritmus végrehajtása, alkotó képesség, szintaktika követése, önállóság.

Fejlesztendő attitűd

- Legyen képes: számítógépet kezelni, szoftvereket használni.
- Tudja önállóan használni a hálózatot és annak alapszolgáltatásait.
- Legyen képes az adott probléma megoldásához kiválasztani az általa ismert módszerek, eszközök és alkalmazások közül a megfelelőt.
- Ismerje fel és legyen képes különféle formákban megfogalmazni a környezetében előforduló tevékenységek algoritmizálható részleteit.
- Adott feladat megoldásához legyen képes algoritmusokat megfogalmazni, tervezni és megvalósítani.
- Legyen képes alkalmazást (szoftvert) tervezni, fejleszteni, futtatni, tesztelni és magyarázó megjegyzésekkel ellátni, előkészítve ezzel a dokumentáció elkészítését.
- Legyen képes programozási feladatot ellátni, egy szoftvert elkészíteni (kódolni), Java programnyelven implementálni.
- Legyen képes értelmezni és javítani a NetBeans által megjelenített hibaüzeneteket.
- Legyen képes programjait úgy megírni, hogy azt később Ő maga is, vagy tanulótársa is fel tudja használni.
- Legyen képes alapvető adatgyűjtési és feldolgozási algoritmusokat (összegzés, számlálás) feladatmegoldásban alkalmazni.

Az óra didaktikai feladatai

- Motiválás
- A figyelem felkeltése
- Ismétlés
- Feladat kitűzése
- Aktivizálás
- A tanulók informálása a tanulási célról
- Szemléletesség
- Az új ismeretet megalapozó tények biztosítása
- A folyamat elemzése
- Következtetés
- Absztrakció
- Általánosítás
- Rögzítés
- Ellenőrzés
- Értékelés
- Visszacsatolás

Tantárgyi kapcsolatok: Adatbázis- és szoftverfejlesztés elmélet, Matematika, Angol, Hálózati ismeretek I., Informatikai alapismeretek, Biológia

Felhasznált források (tankönyv, munkafüzet, feladat- és szöveggyűjtemény, digitális tananyag, online források, szakirodalom stb.):

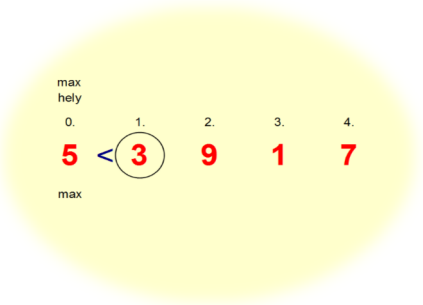
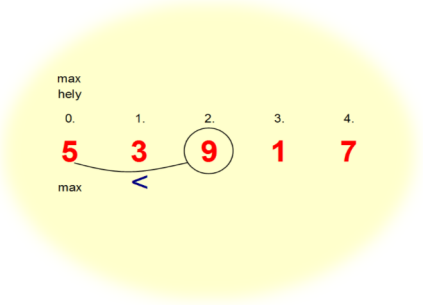
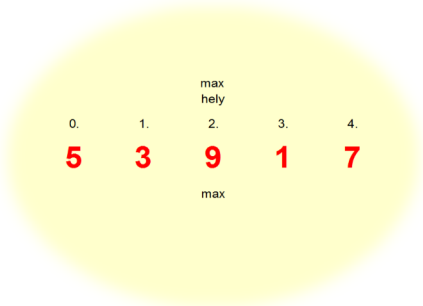
Szerző(k)	Cím	Kiadó	ISBN kód
Vég Csaba – dr. Juhász István	Java – start!	Logos 2000	963 03 9005 1
Vég Csaba	Instant Java/Java EE/SOA I-II	Logos 2000	978-963-06-2601-9 978-963-06-2602-6
Dirk Louis – Peter Müller:	Java 5 Belépés a programozás világába	Panem	963 545 454 6
Benkő Tiborné, Tóth Bertalan	Együtt könnyebb a programozás Java	Computerbooks	963 618 337 6
Rogers Cadenhead	Tanuljuk meg a Java programozási nyelvet 24 óra alatt	Kiskapu Kft.	963 9637 07 6
Angster Erzsébet	Objektumorientált tervezés és programozás Java 1 - 2	4KÖR Bt.	963 00 6262 3 963 00 6263 1
Peter J. DePasquale	Java zsebkönyv	Kiskapu Kft.	963 9637 10 6
Nyékyné Gaizler Judit, Frohner Ákos, Kozsik Tamás (Szerkesztők)	Java 2 útikalauz programozóknak	ELTE TTK Hallgatói Alapítvány	963 463 305 6
Kathy Sierra, Bert Bates	Agyhullám: Java	Kiskapu Kft.	978 963 9637 79 5

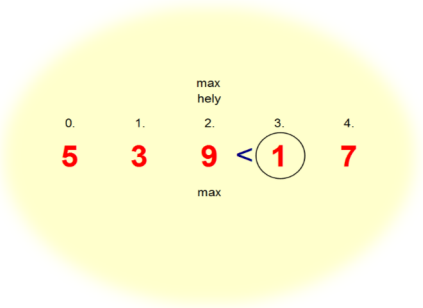
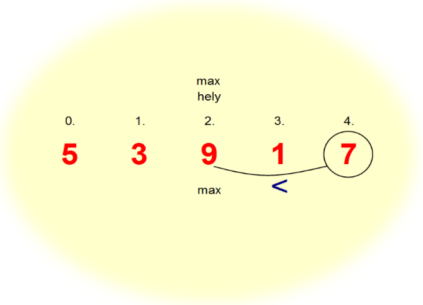
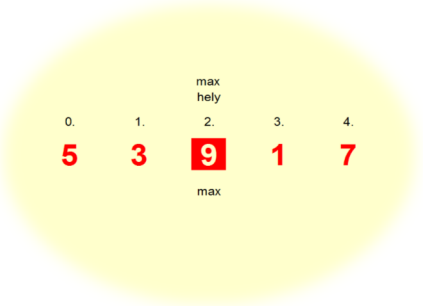
http://docs.oracle.com/javase/7/docs/api/

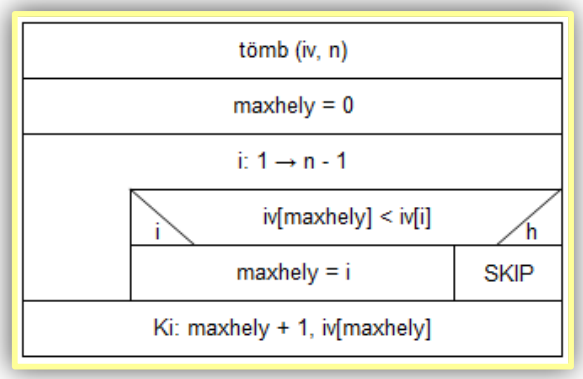
Dátum: éééé. hh. nn.

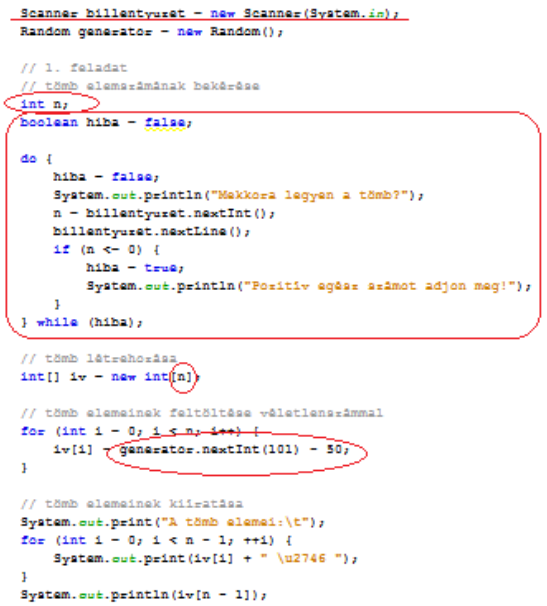
IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MÓDSZEREK TANULÓI MUNKAFORMÁK ESZKÖZÖK MEGJEGYZÉSEK
0 – 4’	Egészséges táplálkozás – legjobb, legegészségesebb áru megvételére törekvés A mai órán mi is a maximális megtalálására törekszünk, megkeressük, hol van egy tömb legnagyobb eleme és melyik az! Az előző órán algoritmusokat alkottunk: kiszámoltuk a tömbelemek összegét, négyzetösszegét, átlagát, szorzatát, illetve megszámláltunk adott tulajdonságú elemeket. Vegyünk egy 5 elemű tömböt a következő elemekkel: 5, 3, 9, 1, 7 Milyen algoritmust tudnánk kitalálni, hogy meg tudjuk valósítani célunkat? Kövesse figyelemmel a következő képek sorozatát, melyek szemléltetik az algoritmus lényegét! Próbálja megfogalmazni a képek alapján az algoritmust!	Motiválás, a figyelem felkeltése,. Ismétlés Feladat kitűzése Aktivizálás	Elbeszélés Megbeszélés Frontális munka

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
4 – 10’	 5 3 9 1 7 *Van egy ötelemű tömbünk.* max hely 0. 1. 2. 3. 4. 5 3 9 1 7 max *Tegyük fel, hogy a 0. elem a leg-nagyobb, azaz az 5, így a 0 lesz a maximumhely, az 5 pedig a maximumérték.*	A tanulók informálása a tanulási célról Szemléletesség Az új ismeretet meg-alapozó tények biztosí-tása A folyamat elemzése	Frontális munka Projektor Szemléltetés Tanári bemutatás Tanári, tanulói elbeszélés

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
	 *Az 1. elemet összehasonlítjuk az aktuális maximummal, azaz a 0. elemmel. Ha nagyobb, akkor ő lesz az aktuális maximum, ha nem, megyünk tovább.*  *A következő, azaz a 2. elemet összehasonlítjuk az aktuális maximummal, azaz a 0. elemmel. Mivel nagyobb nála,*  *ezért a 2. elem lesz az aktuális maximum. Tehát most a maximumhely a 2, a maximumérték a 9.*	Algoritmikus gondolkodás Algoritmus felismerése Algoritmus megértése	Frontális munka Szemléltetés Projektor Magyarázat Megbeszélés

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
	 *A következő, azaz a 3. elemet összehasonlítjuk az aktuális maximummal, azaz a 2. elemmel. Nem nagyobb nála.*	Algoritmizálás	Frontális munka Szemléltetés Projektor
	 *A következő, azaz a 4. indexű elemet összehasonlítjuk az aktuális maximummal, azaz a 2. elemmel. Nem nagyobb nála.*	Algoritmus elemzése	Tanulói elbeszélés Ha úgy ítélem meg, hogy van rá idő: mi történne, ha nem <, hanem ≤ relációjel szerepelne.
	 *Végig értünk a tömbön, nincs több eleme, tehát a 2. indexű eleme a legnagyobb, azaz a 9.*	Algoritmus alkotása Következtetés	szükség esetén Tanári elbeszélés Magyarázat

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
10 – 12’	Fogalmazza meg az algoritmust konkrét adatok nélkül!	Absztrakció Általánosítás	Tanulói elbeszélés szükség esetén Tanári elbeszélés
12 – 17’	Struktogram :  	Visszaemlékezés Struktogram készítése Rögzítés Önellenőrzés Ellenőrzés	Structorizer Microsoft Office Excel 2007, 2010 Irányított, önálló munka Projektor
17 – 18’	Feladat: 1. a) Készítsen alkalmazást, amely [1; 20] zárt intervallumban lévő véletlenszámmal feltölt egy 10 elemű, egész elemtípusú, iv nevű tömböt! A kódolás során a length mezőt használja! Mely kódrészletünket tudnánk felhaszmálni? Gyakorlat13_7-ből a 2. feladatig a kódrészlet átmásolása	Feladat kitűzése Visszaemlékezés	Hálózat Projektor Megbeszélés

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
18 – 22’	Javítások: Scanner objektum törlése hátultesztelő ciklus törlése tömb létrehozásánál n → 10 n átnevezése (CTRL+R) → iv. length → feleslegessé vált deklaráló sor törlése véletlenszám javítása → nextInt(20) + 1 A kódrészlet: 	Algoritmus felismerése Algoritmus alkotása Algoritmizálás Algoritmus megvalósítása Programkód átalakítása, módosítása, javítása	Megbeszélés Fontos a sorrend! Irányított, önálló munka Projektor

```
Scanner billentyuzet = new Scanner(System.in);
Random generator = new Random();

// 1. feladat
// tömb elemszámának bekérése
int n;
boolean hiba = false;

do {
    hiba = false;
    System.out.println("Mekkora legyen a tömb?");
    n = billentyuzet.nextInt();
    billentyuzet.nextLine();
    if (n <= 0) {
        hiba = true;
        System.out.println("Pozitív egész számot adjon meg!");
    }
} while (hiba);

// tömb létrehozása
int[] iv = new int[n];

// tömb elemeinek feltöltése véletlenszámmal
for (int i = 0; i < n; i++) {
    iv[i] = generator.nextInt(101) - 50;
}

// tömb elemeinek kiíratása
System.out.print("A tömb elemei:\t");
for (int i = 0; i < n - 1; ++i) {
    System.out.print(iv[i] + " \u2746 ");
}
System.out.println(iv[n - 1]);
```

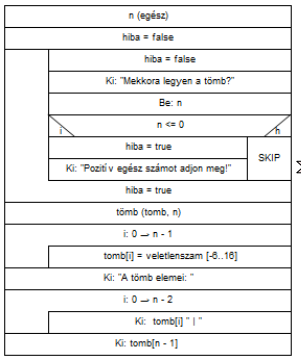
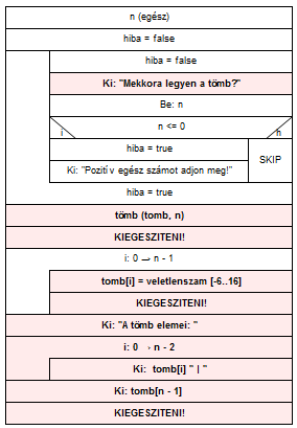
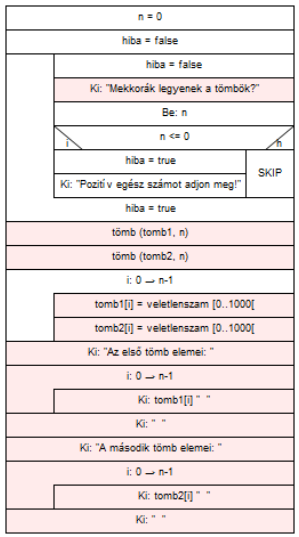
IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK

	A helyes kódrészlet:
	
	`package gyakorlat13;`
	
	`import java.util.Random;`
	`import java.util.Scanner;`
	
	`/**`
	`*`
	`* @author Kovács Henriette`
	`*/`
	`public class Gyakorlat13_8 {`
	
	`public static void main(String[] args) {`
	
	`// 1. feladat`
	`Random generator = new Random();`
	
	`// tömb létrehozása`
	`int[] iv = new int[10];`
	
	`// tömb elemeinek feltöltése véletlenszámmal`
	`for (int i = 0; i < iv.length; i++) {`
	`iv[i] = generator.nextInt(20) + 1;`
	`}`
	
	`// tömb elemeinek kiíratása`
	`System.out.print("A tömb elemei:\t");`
	`for (int i = 0; i < iv.length - 1; ++i) {`
	`System.out.print(iv[i] + " \u2746 ");`
	`}`
	`System.out.println(iv[iv.length - 1]);`
	
	A programkód futtatása
		Programkód írása
		
		Önellenőrzés
		
		Ellenőrzés
		
		Tesztelés
			Importálást ne töröljük, ha a további kódrészletek során nincs rá szükség, azt úgyis jelzi a fejlesztői környezet.
			
			(szükségünk lesz rá ☺)
			
			Egyéni munka
			
			Projektor
			
			Számítógép
			NetBeans IDE 7.3
			Java 1.5

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK

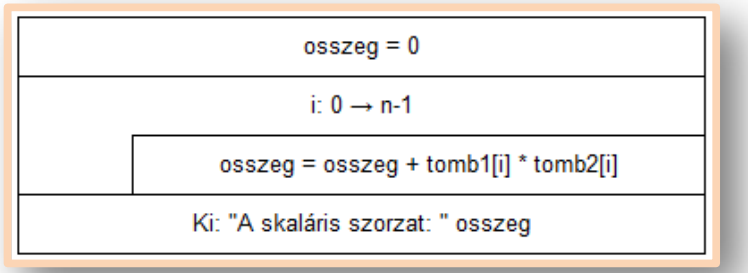
22 – 26’	Feladat:
	b) Adja meg, hogy a tömbben hányadik helyen fordul először elő a tömb legnagyobb eleme és ez mennyi!
	A feladat tulajdonképpen az elkészített algoritmus megvalósítása, kódolása..
	Kód elkészítése:
	`// maximumkiválasztás`
	`int maxhely = 0; // az első elem lokális maximum`
	`for (int i = 1; i < iv.length; ++i) {`
	`  if (iv[i] > iv[maxhely]) {`
	`  maxhely = i;`
	`  }`
	`}`
	`System.out.println("A legnagyobb elem a/az " + (maxhely + 1) + "., "`
	`  + "aminek az értéke " + iv[maxhely] + ".");`
	Futási eredmény:
	`run:`
	`A tömb elemei:  12 8 20 19 20 20 8 2 16 16`
	`---`
	`A legnagyobb elem a/az 3., aminek az értéke 20`
	`---`
	A tanulók munkájának elemzése
		Feladat kitűzése
		Visszaemlékezés
		Visszacsatolás
		Algoritmus megvalósí-tása
		Programkód írása
		Önellenőrzés
		Ellenőrzés
		Értékelés
			Frontális munka
			Projektor
			Hálózat
			Számítógép
			NetBeans IDE 7.3
			Java 1.5
			Projektor

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
26 – 30’	Feladat: 2. a) Töltsön fel 2 azonos méretű tömböt egész számokkal (a tömb méretét beolvasással határozza meg; a feltöltés automatikusan, véletlenszámokkal történjen). Írassa ki az első, majd a második tömb elemeit tabulátorral elválasztva! Mely kódrészletünket tudnánk felhaszmálni? Gyakorlat13_6.java file-ban található kódot. Gyakorlat13_6-hoz tartozó struktogram (Gyakorlat13_6.xlsx) megnyitása. A szükséges változtatások megbeszélése: • a ciklusban a kiíratás megváltoztatása • tömb neve tomb → tomb1 • tomb2 létrehozása (KIEGÉSZÍTÉS) • véletlenszám-generálás javítása • tomb2 tömb feltöltése véletlenszámmal (KIEGÉSZÍTÉS) • kiíratás javítása → tabulátor • tomb2 elemeinek kiíratása (KIEGÉSZÍTÉS) Gyakorlat13_8_2.xlsx tartalmazza az új struktogramot, melyet a hálózatra is felteszek, ahonnan a diákok letöltve a kódolást ez alapján végzik.	Feladatkitűzés Visszaemlékezés Visszacsatolás Összefüggések felismerése Algoritmizálás Logikus gondolkodás Problémamegoldás Algoritmus módosítása, átalakítása	Projektor Gyakorlat13_6.xlsx Megbeszélés Frontális munka Gyakorlat13_8_2.xlsx Hálózat

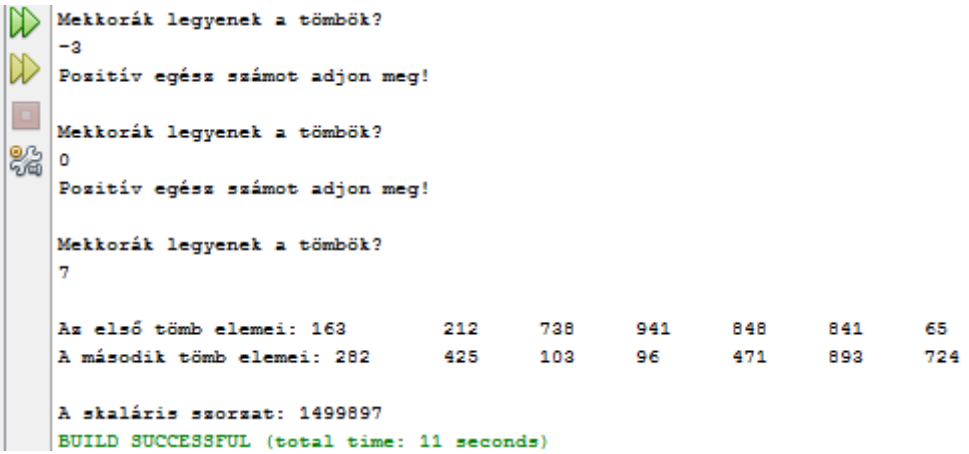
IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
	Gyakorlat13_6.xlsx Struktogram  A szükséges változtatá-sok megbeszélése  Gyakorlat13_8_2.xlsx Megváltoztatott struktogram  	Elemzés Fokozatosság Összehasonlítás Szemléletesség	Hálózat Gyakorlat13_8_2.xlsx Szemléltetés Megbeszélés Projektor Frontális munka
30 – 34	Kód elkészítése: A beírt kommentek, megjegyzések módosítása	Algoritmus megvalósí-tása Precizitás, pontosság	Számítógép NetBeans IDE 7.3 Java 1.5

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK

	``
	// 2. feladat
	Scanner billentyuset = new Scanner(System.in);
	
	// tömbök elemszámának bekérése
	int n;
	boolean hiba = false;
	
	do {
	    hiba = false;
	    System.out.println("Mekkorák legyenek a tömbök?");
	    n = billentyuset.nextInt();
	    billentyuset.nextLine();
	    if (n <= 0) {
	        hiba = true;
	        System.out.println("Pozitív egész számot adjon meg!\n");
	    }
	} while (hiba);
	
	// tömbök létrehozása
	int[] tomb1 = new int[n];
	int[] tomb2 = new int[n];
	
	// 2 tömb elemeinek feltöltése véletlenszámmal
	for (int i = 0; i < n; ++i) {
	    tomb1[i] = generator.nextInt(1000);
	    tomb2[i] = generator.nextInt(1000);
	}
	
	// 2 tömb elemeinek kiíratása
	System.out.print("\nAz első tömb elemei: ");
	for (int i = 0; i < n; ++i) {
	    System.out.print(tomb1[i] + "\t");
	}
	System.out.println();
	
	System.out.print("A második tömb elemei: ");
	for (int i = 0; i < n; ++i) {
	    System.out.print(tomb2[i] + "\t");
	}
	System.out.println();
	``
		Algoritmus végrehajtása
		
		Alkotó képesség
		
		Programkód módosítása, átalakítása, javítása, írása
		Szintaktika követése
		
		Önállóság
		
		Aktivizálás
		
		Gyakorlás
		Megerősítés
		Ellenőrzés
			Gyakorlat13_6.java
			
			Számítógép
			NetBeans IDE 7.3
			Java 1.5
			
			Egyéni munka

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
34 – 35’	Az eddigi kódrészlet futtatása	Futási eredmény elemzése	Egyéni munka
35 – 38’	Feladat: b) Számítsa és írassa ki a 2 tömb skaláris szorzatát! (A skaláris szorzat a megfelelő számpárok szorzatának összege.) Struktogram elkészítése: Valóban szorzatot kell számolnunk? 	Feladatkitűzés Figyelemfelhívás Visszaemlékezés Algoritmikus gondolkodás Algoritmus alkotása Struktogram készítése Ellenőrzés	Projektor Hálózat Frontális munka Megbeszélés Egyéni munka Structorizer Microsoft Office Excel 2007, 2010
38 – 42’	Kód elkészítése:		

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK

	``
	        //Skaláris szorzat
	        long osszeg = 0;
	        for (int i = 0; i < n; ++i) {
	            osszeg += tomb1[i] * tomb2[i];
	        }
	        System.out.println();
	
	        System.out.println("A skaláris szorzat: " + osszeg);
	
	    }
	}
	``
	
	Futási eredmény:
	
	
	``
	Mekkorák legyenek a tömbök?
	-3
	Pozitív egész számot adjon meg!
	
	Mekkorák legyenek a tömbök?
	0
	Pozitív egész számot adjon meg!
	
	Mekkorák legyenek a tömbök?
	7
	
	Az első tömb elemei: 163 212 738 941 848 841 65
	A második tömb elemei: 282 425 103 96 471 893 724
	
	A skaláris szorzat: 1499897
	BUILD SUCCESSFUL (total time: 11 seconds)
	``
		Algoritmus megvalósítása
		Programkód írása
		
		Önellenőrzés
		Ellenőrzés
		
		Futási eredmény elemzése
			Egyéni munka
			
			Számítógép
			NetBeans IDE 7.3
			Java 1.5

42 – 44’	Maximumkiválasztás lényege Ha van idő: minimumkiválsztás miben különbözik ettől?	Rögzítés Megerősítés	Beszélgetés

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
44 – 45’	Tanulók munkájának minősítése	Értékelés	Megbeszélés

Feladatok

1. a) Készítsen alkalmazást, amely [1; 20] zárt intervallumban lévő véletlenszámmal feltölt egy 10 elemű, egész elemtípusú, iv nevű tömböt! A kódolás során a length mezőt használja!

   b) Adja meg, hogy a tömbben hányadik helyen fordul először elő a tömb legnagyobb eleme és ez mennyi!

2. a) Töltsön fel 2 azonos méretű tömböt egész számokkal (a tömb méretét beolvasással határozza meg; a feltöltés automatikusan, véletlenszámokkal történjen). Írassa ki az első, majd a második tömb elemeit tabulátorral elválasztva!

   b) Számítsa és írassa ki a 2 tömb skaláris szorzatát! (A skaláris szorzat a megfelelő számpárok szorzatának összege.)

Reflexió

Aki kezébe veszi az óravázlatomat, meglepődhet azon, hogy hogyan kerül ide az egészséges táplálkozás, mi köze van a maximumkiválasztáshoz. Ehhez tudni kell, hogy nagyon sokat beszélgetek a diákokkal az egészséges életmódról, annak élettani hatásairól és bár annak ellenére, hogy kerekesszékhez kötöttek (1 kivételével), sportolnak, de náluk különösen fontos, hogy mit visznek be szervezetükbe, hogy az elhízást elkerüljék, mert ugye amit beviszünk, azzá válunk. A cél tehát a legegészségesebb, legjobb kiválasztása – ami manapság elég nehéz feladat a tartósítószerek térhódítása miatt – vagyis törekszünk, mint mindenben a maximalizmus elvére. Így tehát a tömbelemek közül is szeretnénk kiválasztani a maximumot. Ezzel a kis bevezetővel kellőképpen ráhangoltam a diákokat az óra anyagára, tudtam, erről mindig szívesen beszélgetnek velem.

Az óra az algoritmikus gondolkodás kompetencia szintjeit képező, valamennyi összetevőt fejleszti. Ezek: az algoritmus megértése, azaz megérteni, hogy mit kell tenni és megérteni, hogy miért azt kell tenni, az algoritmus felismerése, az algoritmus elemzése, az algoritmus végrehajtása, az algoritmus alkotása, az algoritmus megvalósítása, az algoritmus módosítása, átalakítása, az algoritmus tervezése.

A maximumkiválasztás algoritmusára nem jöttek rá, nem tudták megközelíteni a problémát, de a szemléletesség, mint didaktikai feladat bevetése most igazán célszerű volt, mert teljesen akadálymentesen tudtam haladni a diasorokon és a végén az általánosítás, az absztrakt gondolkodás sem ütközött akadályokba. A feladat algoritmusának leírása, a struktogram készítése ennek ellenére egy kicsit nehézkesen ment, a kódolás viszont, az algoritmus megvalósítása nem okozott problémát.

A második feladatban nagy hangsúlyt fektettem elkészített struktogramok és megírt kódrészletek felhasználására. Ezáltal lehetőség nyílt újabb algoritmizálásra (hiszen a forráskódban nem mindegy, hogy milyen sorrendben hajtjuk végre a módosításokat), a logikus gondolkodás fejlesztésére, forráskód átalakítására, módosítására. Utolsó lépésként pedig azt gondoltam, hogy a sorozatszámítás gondot fog jelenteni, mivel két tömbön kellett dolgozniuk, de ez nem okozott problémát; viszont nem összeadásban, hanem szorzásban gondolkoztak volna a skaláris szorzat elnevezés miatt, ekkor felhívtam figyelmüket a zárójelbe

tett magyarázó szövegre, melyből egyből kiderült számukra, hogy mit is kell tenniük. Tudatosítottam bennük, hogy mindig legyenek precízek, pontosak és a zárójelben lévő, látszólag nem lényeges gondolatok is jelentősek lehetnek.

Az órát sikeresnek ítélem meg, mert a reflexiókban felsoroltakon kívül még számos, a terv elején felsorolt készségeket és képességeket is fejlesztettem.

ÓRATERVEZET - ÖSSZETETT ELEMTÍPUSÚ TÖMBÖK. TÖMBELEMEK CSERÉJE.

A pedagógus neve: Kovács Henriette

A szakmacsoport száma és megnevezése: 7. Informatika

A szakképesítés azonosító száma: 54 213 05

A szakképesítés megnevezése: Szoftverfejlesztő

Iskolai rendszerű szakképzés (a szakképzési évfolyamok száma: 2)

Szakmai követelménymodul: 10817-12 Hálózatok, programozás és adatbázis-kezelés

Tantárgy: Adatbázis- és szoftverfejlesztés gyakorlat

Programozási nyelv: Java

Fejlesztői környezet: NetBeans IDE 7.3

Osztály - csoport: I/13. S - SZ2

Az óra tananyaga: Összetett elemtípusú tömbök. Tömbelemek cseréje.

Az óra cél- és feladatrendszere

A fejlesztendő attitűd, készségek, képességek, a tanítandó ismeretek (fogalmak, szabályok stb.) és az elérendő fejlesztési szint, tudásszint megnevezése:

- A tanuló tartsa be a számítógépes munka szabályait, különös tekintettel a balesetek megelőzésére.
- Felkelteni és folyamatosan ébren tartani a tanulók érdeklődését a szoftverfejlesztés iránt. Kialakítani a tanulókban az informatikai ismereteik folyamatos bővítésének és megújításának igényét.
- Az algoritmikus gondolkodás fejlesztése: a tanulók önálló, rendszerezett, logikus gondolkodásának fejlesztése; a logikus gondolkodás lényeges a problémamegoldásban, és az algoritmusok elkészítésében, kódolásában.
- A feladatmegoldáshoz megfelelő adattípus kiválasztásának fejlesztése.
- Legyen képes a NetBeans fejlesztői környezet kényelmi funkcióinak készségszintű használatára.
- Önálló munkára nevelés és felzárkóztatás.
- Alkotó munkára nevelés: a végeredmény egy új termék, egy program lesz.
- Az esztétikai készség fejlesztése: igény és készség a számítógépes produktum esztétikus formájának kialakítására, a program helyes strukturálására.
- *Tudjon a szakirodalmak között tájékozódni.*
- *Tudjon algoritmust alkotni.*
- *Tudjon algoritmust magvalósítani.*
- *Tudjon megfelelő adatszerkezetet és adattípust választani.*
- *Tudjon a NetBeans IDE-ben osztályt létrehozni, állományt átnevezve másolni.*
- *Tudjon tömböt létrehozni, billentyűzetről feltölteni, kiíratni.*
- *Tudja egyszerű és összetett elemtípusú tömb alapértelmezett értékeit meghatározni.*
- *Tudjon a Java hivatalos dokumentációjában keresni.*
- *Tudjon Internetről információt gyűjteni.*
- *Tudjon Scanner objektumot úgy létrehozni, hogy a magyar ’ő’ és ’ű’ karakterek is megjeleníthetők legyenek.*
- *Tudjon programkódot módosítani, futtatni, a futási eredményt elemezni.*
- *Tudjon tömbelemeket cserélni.*
- *Tudja, hogy összetett típus értékadásakor, a referencia állítódik át.*
- *Ismerje a RuntimeException futási hibát.*
- *Tudja tömb referenciájának alapértelmezett értékét meghatározni.*

Fejlesztendő készségek, képességek

– Algoritmizálás, algoritmikus gondolkodás, algoritmus felismerése, algoritmus alkotása, adatszerkezet-választás, típusválasztás, állománykezelés, programkód írása, visszaemlékezés, precizitás, pontosság, futtatás, absztrakt gondolkodás, megfigyelés, kódrészlet írása, olvasott szakmai szöveg megértése, programkód átalakítása, önállóság, logikus gondolkodás, visszaemlékezés, algoritmus megvalósítása, programkód átalakítása, önellenőrzés, futási hiba felismerése, diagnosztizálás, hibakeresés, fejlesztői környezet jelzésének figyelembevétele, futási eredmény elemzése.

Fejlesztendő attitűd

– Legyen képes: számítógépet kezelni, szoftvereket használni.
– Tudja önállóan használni a hálózatot és annak alapszolgáltatásait. Tudjon adatokat megkeresni, elérni a hálózati szolgáltatások alkalmazásával.
– Ismerje fel és legyen képes különféle formákban megfogalmazni a környezetében előforduló tevékenységek algoritmizálható részleteit.
– Legyen képes elemi és összetett adatok megkülönböztetésére, kezelésére, használatára.
– Legyen képes a problémamegoldáshoz szükséges adatok és az eredmény kapcsolatának megtervezésére, értelmezésére.
– Adott feladat megoldásához legyen képes algoritmusokat megfogalmazni, tervezni és megvalósítani.
– Legyen képes alkalmazást (szoftvert) tervezni, fejleszteni, futtatni, tesztelni és magyarázó megjegyzésekkel ellátni, előkészítve ezzel a dokumentáció elkészítését.
– Legyen képes programozási feladatot ellátni, egy szoftvert elkészíteni (kódolni), Java programnyelven implementálni.
– Legyen képes értelmezni és javítani a NetBeans által megjelenített hibaüzeneteket.
– Legyen képes programjait úgy megírni, hogy azt később Ő maga is, vagy tanulótársa is fel tudja használni.
– Legyen képes eligazodni és tudjon keresni a Java hivatalos dokumentációjában.

Az óra didaktikai feladatai

- Motiválás
- A figyelem felkeltése
- Szemléletesség
- Figyelemfelhívás
- Szakirodalom megismertetése
- Tudományosság
- Feladat kitűzése
- Ismétlés
- A szükséges előzetes ismeretek felidézése
- A tanulók informálása a tanulási célról
- Absztrakció, általánosítás
- Új ismeret közlése
- Fokozatosság
- Információ gyűjtése, elemzése
- Problémafelvetés
- Új ismereteket megalapozó tények biztosítása
- Futási eredmény elemzése
- Megvalósítás
- Következtetés
- Visszacsatolás
- Megerősítés
- Rögzítés
- Aktivizálás
- Ellenőrzés
- Rendszerezés
- Értékelés
- Futási eredmény átláthatósága, követhetősége

Tantárgyi kapcsolatok: Adatbázis- és szoftverfejlesztés elmélet, Matematika, Angol, Hálózati ismeretek I., Informatikai alapismeretek, Könyvtárismeret, Magyar

Felhasznált források (tankönyv, munkafüzet, feladat- és szöveggyűjtemény, digitális tananyag, online források, szakirodalom stb.):

Szerző(k)	Cím	Kiadó	ISBN kód
Vég Csaba – dr. Juhász István	Java – start!	Logos 2000	963 03 9005 1
Vég Csaba	Instant Java/Java EE/SOA I-II	Logos 2000	978-963-06-2601-9 978-963-06-2602-6
Dirk Louis – Peter Müller:	Java 5 Belépés a programozás világába	Panem	963 545 454 6
Benkő Tiborné, Tóth Bertalan	Együtt könnyebb a programozás Java	Computerbooks	963 618 337 6
Rogers Cadenhead	Tanuljuk meg a Java programozási nyelvet 24 óra alatt	Kiskapu Kft.	963 9637 07 6
Angster Erzsébet	Objektumorientált tervezés és programozás Java 1 - 2	4KÖR Bt.	963 00 6262 3 963 00 6263 1
Peter J. DePasquale	Java zsebkönyv	Kiskapu Kft.	963 9637 10 6
Nyékyné Gaizler Judit, Frohner Ákos, Kozsik Tamás (Szerkesztők)	Java 2 útikalauz programozóknak	ELTE TTK Hallgatói Alapítvány	963 463 305 6
Kathy Sierra, Bert Bates	Agyhullám: Java	Kiskapu Kft.	978 963 9637 79 5

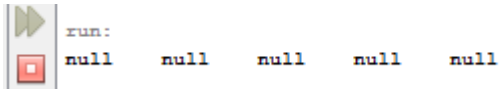
http://docs.oracle.com/javase/7/docs/api/

http://hu.wikipedia.org/wiki/ISO/IEC_8859-2

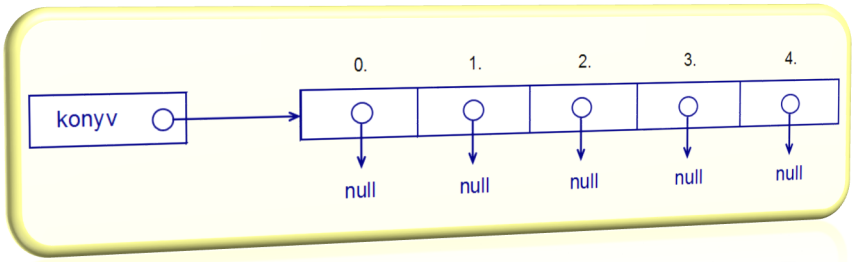
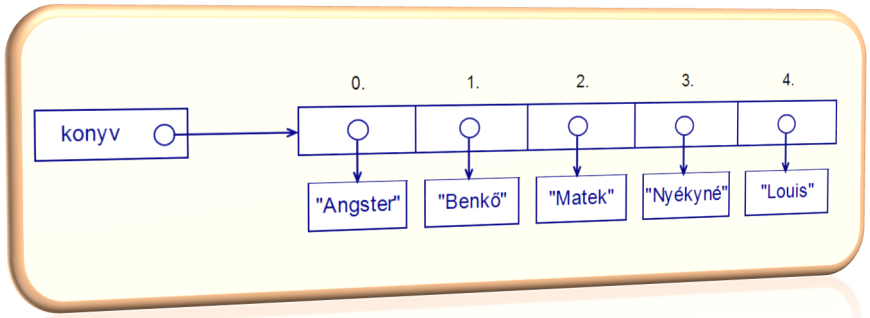
Dátum: éééé. hh. nn.

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MÓDSZEREK TANULÓI MUNKAFORMÁK ESZKÖZÖK MEGJEGYZÉSEK
0 – 3’	Szoktak-e olvasni? Milyen könyveket? Valamilyen szempont szerint rendezve vannak a könyveik a könyvespolcon? Nekem igen, külön vannak a szakkönyvek, és azon belül egymás mellett a Java, illetve a matek könyvek. A múltkor az egyik matek könyvet nem találtam, mert takarítás során rossz helyre, a Java könyvekhez lett betéve. Fogtam és kicseréltem őket. Mondjuk el azon lépéseket, hogyan cseréltem ki a két könyvet! Az órára a következő Java könyveket és egy matematikával kapcsolatos könyvet viszek be, ebbe a sorrendbe elhelyezve: Angster Erzsébet: Objektumorientált tervezés és programozás Java 1 - 2 Benkő Tiborné, Tóth Bertalan: Együtt könnyebb a programozás Java Obádovics J. Gyula: Matematika Dirk Louis – Peter Müller: Java 5 Belépés a programozás világába	Motiválás A figyelem felkeltése Algoritmizálás Szemléletesség Figyelemfelhívás Szakirodalom megismertetése Tudományosság	Beszélgetés Frontális munka Elbeszélés Megbeszélés Elbeszélés Könyvek Szemléltetés

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
3 – 4'	Nyékyné Gaizler Judit, Frohner Ákos, Kozsik Tamás (Szerkesztők): Java 2 útikalauz programozóknak Cserélés lépései, az algoritmus (közben végrehajtom amit mondanak): - Megfogom a matek könyvet és leteszem az asztalra. - Java könyvet beteszem a matek könyv helyére, (ami jelen pillanatban az asztalon van). - Az asztalról beteszem a matek könyvet a már levett Java könyv helyére.	Algoritmikus gondol-kodás Algoritmus felisme-rése Algoritmus alkotása Algoritmizálás Szemléletesség	Frontális munka Tanulói elbeszélés Könyvek Szemléltetés
4 – 5’	Feladat: Készítse el a feladat programját, azaz a program cseréljen ki két könyvet! Mibe tároljuk a könyveket? *Válasz: tömbbe tároljuk a könyveket* Milyen típusú elemei lesznek a tömbnek? *Válasz: String típusú elemei lesznek a tömbnek* Hány elemű lesz a tömb? *Válasz: 5 elemű lesz a tömb.* Milyen nevet adna a file-nak?	Feladat kitűzése Adatszerkezet-válasz-tás Típusválasztás	Elbeszélés Frontális munka Megbeszélés

<table>
<tr><th>IDŐKERET</th><th>AZ ÓRA MENETE</th><th>CÉLOK ÉS FELADATOK</th><th>NEVELÉSI-OKTATÁSI STRATÉGIA
MEGJEGYZÉSEK</th></tr>
<tr>
<td>5 – 6’</td>
<td>Hozzon létre a Gyakorlat13 projekt alapértelmezett csomagjába egy Elem-Csere nevű osztályt és abban egy 5 elemű String tömböt!
Tömb létrehozása:
<code>String[] konyv = new String[5];</code></td>
<td>Állománykezelés
Programkód írása</td>
<td>Egyéni munka
Számítógép
NetBeans IDE 7.3
Java 1.5</td>
</tr>
<tr>
<td>6 – 8’</td>
<td>Az újonnan létrehozott tömb elemei alapértelmezés szerinti értékeket vesznek fel. Egyszerű típusú változók esetén ezeket megtapasztaltuk.
primitív típusú változók esetén: 0, 0.0, \u0000, false
Hogyan tudjuk megnézni, hogy összetett elemtípusú tömb elemeinek mik az alapértelmezett értékei?
Ki kell íratni a tömböt.
<pre>// tömb kiíratása
for (int i = 0; i < konyv.length; i++) {
    System.out.print(konyv[i] + "\t");
}
System.out.println();</pre>Futási eredmény:


A memóriafoglalás:</td>
<td>Visszaemlékezés
Ismétlés
A szükséges előzetes ismeretek felidézése
Motiválás
A tanulók informálása a tanulási célról
Programkód írása
Precizitás, pontosság
Futtatás
Szemléletesség</td>
<td>Megbeszélés
Frontális munka
Számítógép
NetBeans IDE 7.3
Java 1.5
Projektor
Egyéni munka</td>
</tr>
</table>

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
8 – 9’	 Az alapértelmezett értékek: **referenciák** esetén: **null**	Figyelem felkeltése Absztrakt gondolkodás Absztrakció, általáno-sítás Új ismeret közlése	Szemléltetés Tanári bemutatás Magyarázat Frontális munka Tábla
9 – 10’	Töltsük fel a tömböt! Bevitt könyvek alapján történik 	Megfigyelés Fokozatosság	Szemléltetés Tanári bemutatás Frontális munka Tábla

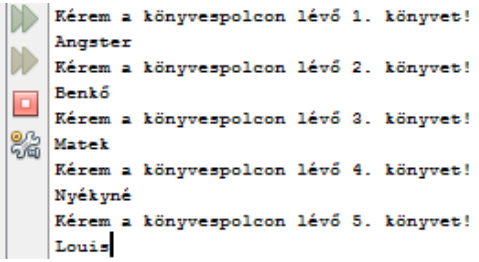
IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK

10 – 13’	A tömb elemeit a felhasználótól kérje be!
	
	Kód elkészítése:
	
	`// tömb feltöltése beolvasással`
	`Scanner billentyuzet = new Scanner(System.in);`
	
	`for (int i = 0; i < konyv.length; i++) {`
	`  System.out.println("Kérem a könyvespolcon lévő "`
	`  + (i + 1) + ". könyvet! ");`
	`  konyv[i] = billentyuzet.nextLine();`
	`}`
		Megvalósítás
		
		Kódrészlet írása
			Számítógép
			NetBeans IDE 7.3
			Java 1.5
			Projektor
			
			Egyéni munka

13 – 15’	Ékezetes karakterek beolvasásánál gond van.
	
	Hogyan tudnánk megoldani a problémát?
	
	A segítség a Java dokumentációja:
	
	http://docs.oracle.com/javase/7/docs/api/
	
	Scanner
	
	`public Scanner(InputStream source,`
	`  String charsetName)`
	
	Constructs a new `Scanner` that produces values scanned from the specified input stream. Bytes from the stream are converted into characters using the specified charset.
	
	Parameters:
	
	`source` - An input stream to be scanned
	
	`charsetName` - The encoding type used to convert bytes from the stream into characters to be scanned
	
	Throws:
	
	`IllegalArgumentException` - if the specified character set does not exist
		Problémafelvetés
		
		Információ gyűjtése, elemzése
			Irányított, önálló munka
			
			Számítógép
			Internet

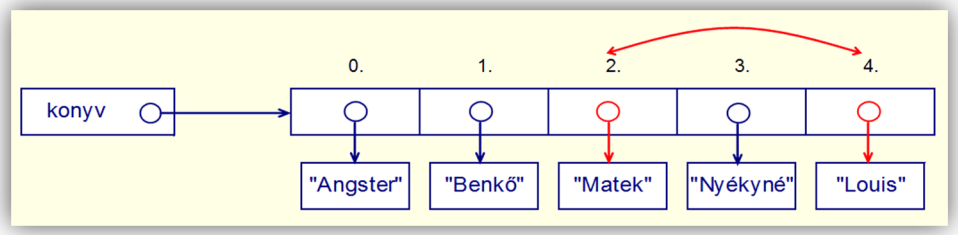
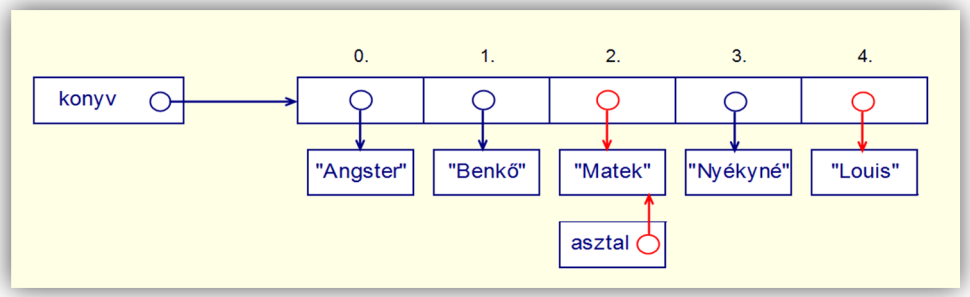
IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK

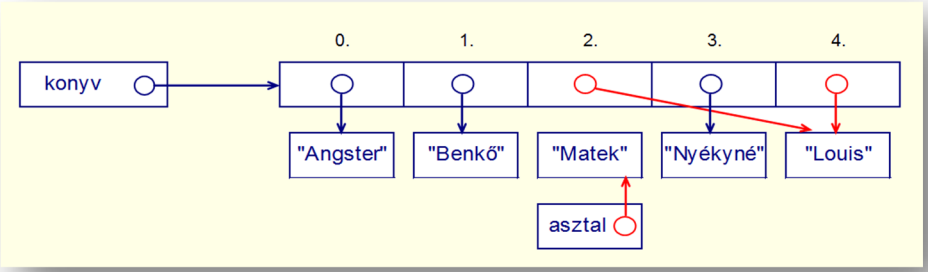
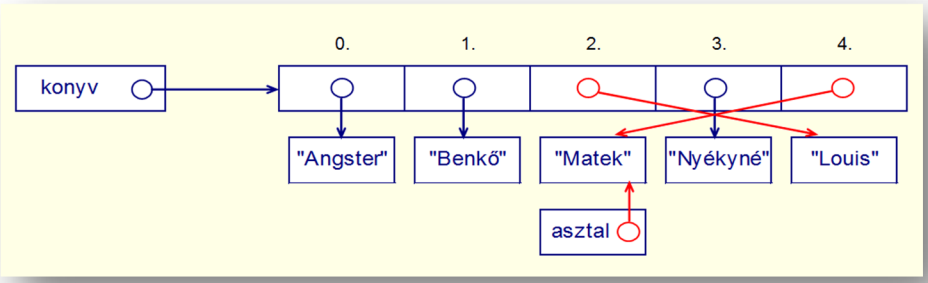
15 – 17’
 	
17 – 18’
	The encoding type used to convert bytes from the stream into characters to be scanned.
	
	Vagyis: A kódolás típusát használjuk a beolvasott karakterek konvertálására.
	
	Mi lesz vajon a kódolás típusa?
	
	Nézzünk utána pontosan!
	
	http://hu.wikipedia.org/wiki/ISO/IEC_8859-2
	
	ISO-8859-2 az ISO/IEC 8859-es karakterkódolási szabvány második része
	
	Az ISO által kettes számú latin ábécének nevezett 191 karakter mindegyikének egybájtos (nyolcbites) kódját adja meg. A 191 karakter között minden magyar ékezetes betű megtalálható (a sok más készletből hiányzó *ő* és *ű* is).
	
	Egészítsük ki a kódot:
	
	`// tömb feltöltése beolvasással`
	`Scanner billentyuzet = new Scanner(System.in, "ISO-8859-2");`
	
	`for (int i = 0; i < konyv.length; i++) {`
	`  System.out.println("Kérem a könyvespolcon lévő "`
	`  + (i + 1) + ". könyvet! ");`
	`  konyv[i] = billentyuzet.nextLine();`
	`}`
	
	Futtatás:
	
	most már nincs probléma a „Benkő”-vel sem.
		Olvasott szakmai szöveg megértése
		
		Új ismereteket megalapozó tények biztosítása
		
		Megvalósítás
		Programkód átalakítása
		
		Futási eredmény elemzése
		Következtetés
			Irányított, önálló munka
			
			Megbeszélés
			
			Számítógép
			Internet
			
			Egyéni munka

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
18 – 20’		Önállóság	Számítógép NetBeans IDE 7.3 Java 1.5 Projektor

20 – 22’	A végén szeretném látni a könyvek sorrendjét a csere előtt és a csere után is, ezt hogy érem el?
	Válasz: Kiíratom a könyvek sorrendjét a csere előtt is és a csere után is.
	Kódrészlet:
	`// tömb kiíratása`
	`System.out.println("\nA könyvek a csere előtt: ");`
	`for (int i = 0; i < konyv.length - 1; i++) {`
	`System.out.print(konyv[i] + " \u2746 ");`
	`}`
	`System.out.println(konyv[konyv.length - 1]);`
		Feladatkitűzés
		Logikus gondolkodás
		Precizitás, pontosság
			Megbeszélés
			Egyéni munka

22 – 23’	És most jön végre maga a csere. ☺ - Megfogom a matek könyvet és leteszem az asztalra. - Java könyvet beteszem a matek könyv helyére, (ami jelen pillanatban az asztalon van). - Az asztalról beteszem a matek könyvet a már kivett Java könyv helyére.	Visszacsatolás Visszaemlékezés Megerősítés	Frontális munka Tanulói elbeszélés

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
23 – 24’	A kódolással párhuzamosan kövessük azt is, hogy „körülbelül” mi is történik a memóriában:  Az előző ismereteinkből tudjuk, hogy Javaban: összetett típus értékadásakor a referencia állítódik át.	Szemléletesség Ismétlés Rögzítés Megerősítés	Frontális munka Tábla Megbeszélés
24 – 25’	Megfogom a matek könyvet és leteszem az asztalra. `String asztal = konyv[2];`  	Algoritmikus gondol-kodás Absztrakt gondolkodás	Frontális munka Tábla Elbeszélés Magyarázat

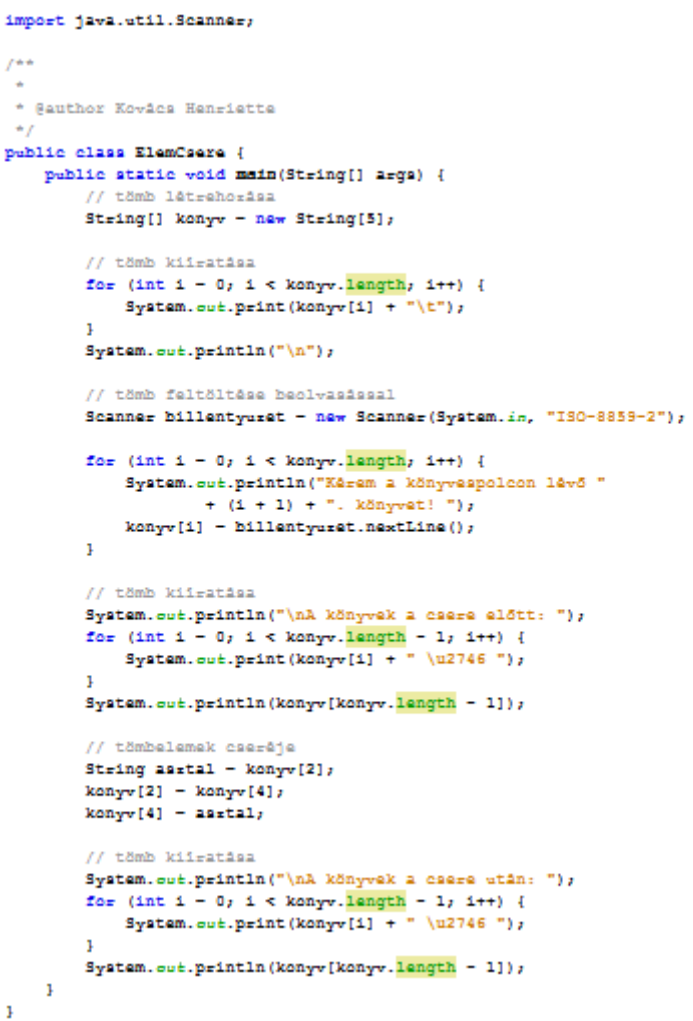
IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
25 – 27’	Java könyvet beteszem a matek könyv helyére. `konyv[2] = konyv[4];` 	Fokozatosság Logikus gondolkodás Megfigyelés	Frontális munka Tábla Megbeszélés Magyarázat
27 – 28’	Az asztalról beteszem a matek könyvet a már kivett Java könyv helyére `konyv[4] = asztal;` 	Tudományosság Aktivizálás	Frontális munka Tábla Tanulói elbeszélés

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK

28 – 30’	Kódrészlet elkészítése (a kommentekről ugye nem feledkeznek meg!):
	
	`// tömbelemek cseréje`
	`String asztal = konyv[2];`
	`konyv[2] = konyv[4];`
	`konyv[4] = asztal;`
	
	Kód elkészítése - csere utáni kiíratás:
	
	`// tömb kiíratása`
	`System.out.println("\nA könyvek a csere után: ");`
	`for (int i = 0; i < konyv.length - 1; i++) {`
	`System.out.print(konyv[i] + " \u2746 ");`
	`}`
	`System.out.println(konyv[konyv.length - 1]);`
	`}`
	`}`
		Figyelemfelhívás
		
		Algoritmus megvalósítása
		Programkód írása
		
		Programkód átalakítása
			Számítógép
			NetBeans IDE 7.3
			Java 1.5
			Projektor
			
			Egyéni munka

30 – 31’	Futtatás:
	
	`A könyvek a csere után:`
	`Angster ❆ Benkő ❆ Louis ❆ Nyékyné ❆ Matek`
	`BUILD SUCCESSFUL (total time: 1 minute 41 seconds)`
		Futási eredmény elemzése	NetBeans IDE 7.3
			
			Egyéni munka

	Az 1. feladat kódja :		

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
32 – 33’		Önellenőrzés Ellenőrzés Rendszerezés Rögzítés Megerősítés Értékelés	Szemléltetés Projektor Elbeszélés Frontális munka

```java
import java.util.Scanner;

/**
 *
 * @author Kovács Henriette
 */
public class ElemCsere {
    public static void main(String[] args) {
        // tömb létrehozása
        String[] konyv = new String[5];

        // tömb kiíratása
        for (int i = 0; i < konyv.length; i++) {
            System.out.print(konyv[i] + "\t");
        }
        System.out.println("\n");

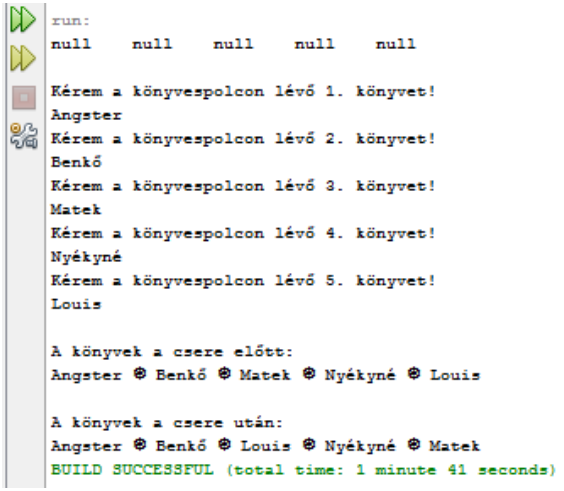
        // tömb feltöltése beolvasással
        Scanner billentyuzet = new Scanner(System.in, "ISO-8859-2");

        for (int i = 0; i < konyv.length; i++) {
            System.out.println("Kérem a könyvespolcon lévő "
                    + (i + 1) + ". könyvet! ");
            konyv[i] = billentyuzet.nextLine();
        }

        // tömb kiíratása
        System.out.println("\nA könyvek a csere előtt: ");
        for (int i = 0; i < konyv.length - 1; i++) {
            System.out.print(konyv[i] + " \u2746 ");
        }
        System.out.println(konyv[konyv.length - 1]);

        // tömbelemek cseréje
        String asztal = konyv[2];
        konyv[2] = konyv[4];
        konyv[4] = asztal;

        // tömb kiíratása
        System.out.println("\nA könyvek a csere után: ");
        for (int i = 0; i < konyv.length - 1; i++) {
            System.out.print(konyv[i] + " \u2746 ");
        }
        System.out.println(konyv[konyv.length - 1]);
    }
}
```

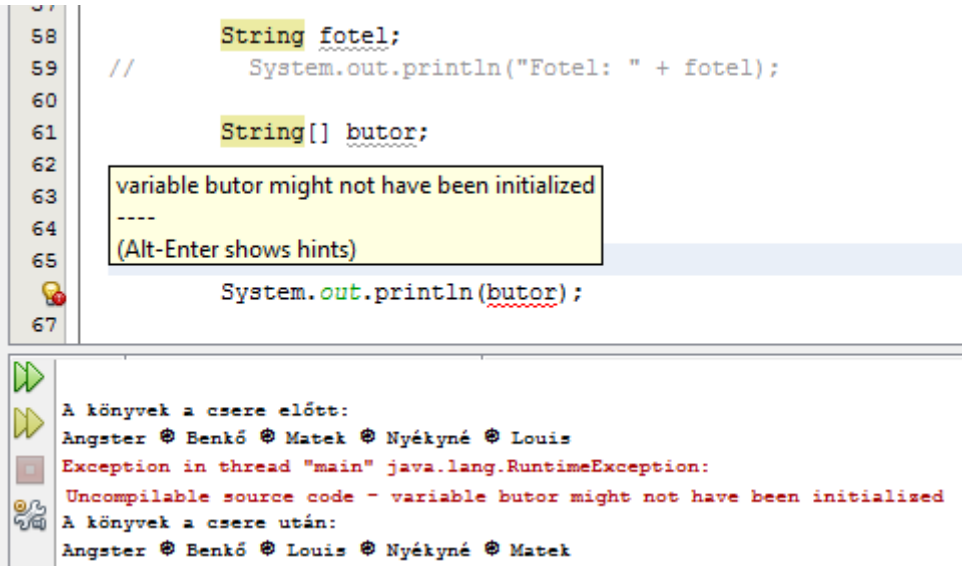
IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
33 – 34’	Futási eredmény:  	Precizitás, pontosság Futási eredmény átlátható-sága, követhető-sége	Megbeszélés Szemléltetés Projektor Frontális munka
34 – 35’	Feladat: 2. a) Győződjön meg róla, hogy mely objektumra mutatnak a cserében részt vevő tömbelemek és az asztal referencia (a cserénél bevezetett segédváltozó)! b) Állítsa az asztal referenciát null-ra, írassa ki, nézze meg, mi történik! c) Deklaráljon egy fotel nevű változót, mely egy String objektumra képes mutatni, írassa ki, nézze meg, mi történik!	Feladat kitűzése Fokozatosság	Projektor Hálózat

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK

35 – 38’	Kód:
	
	`// 2. feladat`
	`System.out.println(konyv[2]);`
	`System.out.println(konyv[4]);`
	`System.out.println(asztal);`
	
	`asztal = null;`
	`System.out.println("\nAsztal: " + asztal);`
	
	`String fotel;`
	`//  System.out.println("Fotel: " + fotel);`
	
	`}`
	`}`
		Megerősítés
		
		Ismétlés
		Új ismeretet megalapozó tények biztosítása
			Irányított, önálló munka
			Projektor
			Számítógép
			NetBeans IDE 7.3
			Java 1.5
			Megbeszélés

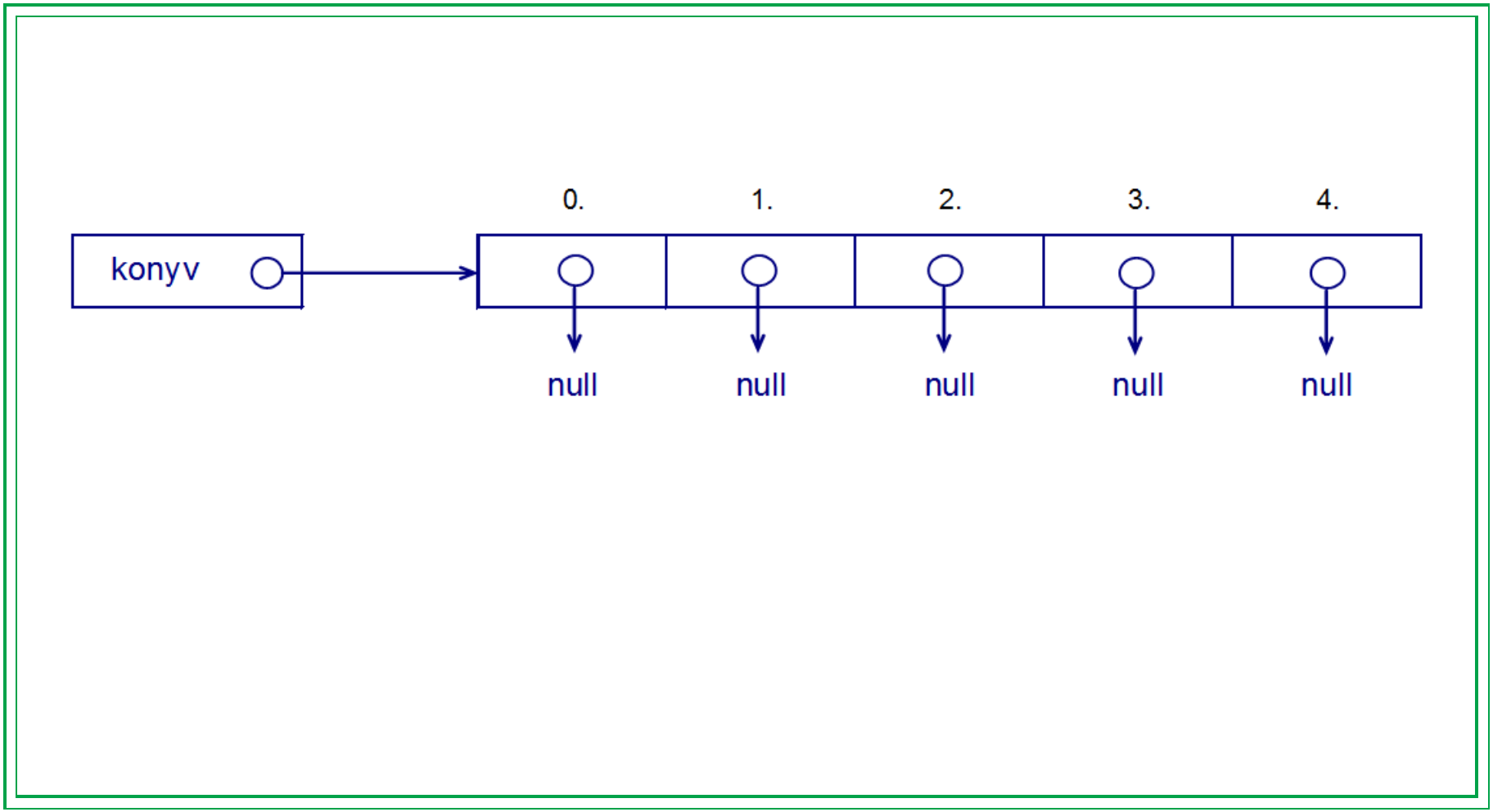
38 – 39’	Futási eredmény:
	
	`Louis`
	`Matek`
	`Matek`
	
	`Asztal: null`
	`BUILD SUCCESSFUL (total time: 32 seconds)`
	
	De, ha az utolsó utasítás nincs kikommentezve:
	
	`Exception in thread "main" java.lang.RuntimeException: Uncompilable source code - variable fotel might not have been initialized`
	`at gyakorlat13.Gyakorlat13_9.main(Gyakorlat13_9.java:59)`
	
	`java.lang.RuntimeException: Uncompil`
	`orlat13_9.main(Gyakorlat13_9.java:59`
	
	RuntimeException: futási hiba keletkezik, a fotel változó nem lett inicializálva
		Futási eredmény elemzése
		
		Olvasott szakmai szöveg megértése
		
		Futási hiba felismerése
			Frontális munka
			
			Megbeszélés
			Magyarázat
			Projektor
			Számítógép
			NetBeans IDE 7.3
			Java 1.5

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK

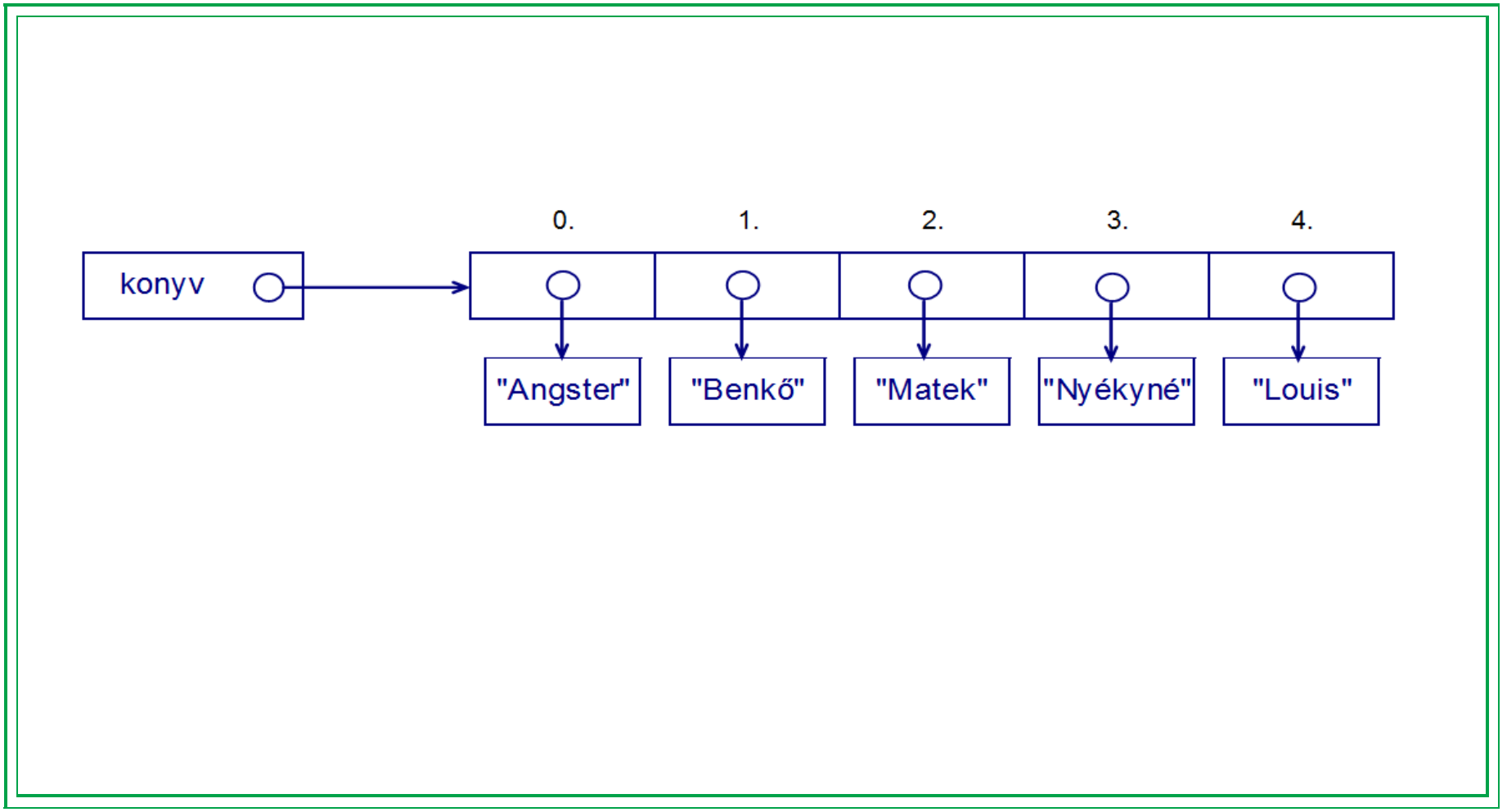
39 – 40’	Feladat:
	d) Deklaráljon egy butor nevű változót, mely egy olyan tömb objektumra képes mutatni, melynek elemei String-ek, írassa ki, nézze meg, mi történik!
	`String[] butor;`
	`System.out.println(butor);`
	Futás:
	
	
		Feladat kitűzése
		Programkód írása
		Diagnosztizálás, hibakeresés
		Fejlesztői környezet jelzésének figyelembe vétele
		Futási eredmény elemzése
			Projektor
			Hálózat
			Számítógép
			NetBeans IDE 7.3
			Java 1.5
			Frontális munka
			Megbeszélés

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
40 – 41’	Tehát tömböknél is ugyanaz történik, mint az előbb a fotel változónál: RuntimeException: futási hiba keletkezik, a butor nevű változó nem lett inicializálva – erre a fejlesztői környezet is felhívja a figyelmünket, ezért mindig figyeljük a NetBeans IDE jelzéseit.	Általánosítás	Megbeszélés Projektor NetBeans IDE Frontális munka
41 – 42’	A tömb referenciájának alapértelmezés szerinti értéke **lokális deklaráció** esetén: **definiálatlan**	Következtetés, absztrakció Új ismeret közlése	Elbeszélés
42 – 44’	Ismétlés: Tömbelemek cseréje	Rögzítés Megerősítés	Beszélgetés
44 – 45’	Tanulók munkájának minősítése	Értékelés	Megbeszélés

TÁBLAKÉP I.

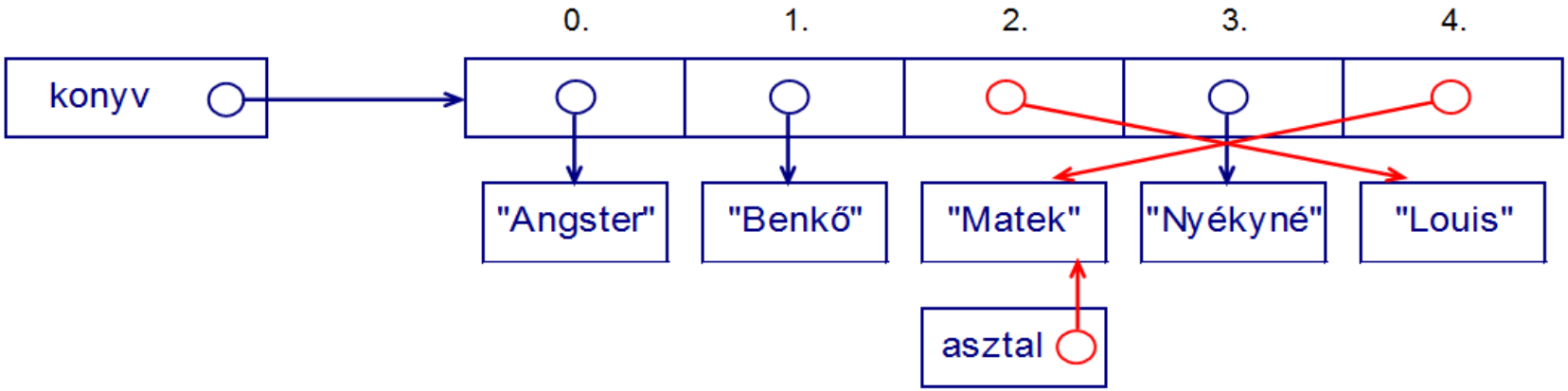


TÁBLAKÉP II.



TÁBLAKÉP III.

Tömbelemek cseréje




```
String asztal = konyv[2];

konyv[2] = konyv[4];

konyv[4] = asztal;
```

Feladatok

1. a) Írjon programot, mely egy könyvespolcon a harmadik és ötödik helyen lévő két könyvet kicseréli! A könyvespolcon 5 db könyv legyen, az egyes könyveket kérje be a felhasználótól (a könyveket ne egyszerű típusú változókkal azonosítsa)!

   b) Írassa ki a könyvespolcon lévő könyveket úgy, hogy látható legyen a csere!

2. a) Győződjön meg róla, hogy mely objektumra mutatnak a cserében részt vevő tömbelemek és az asztal referencia (a cserénél bevezetett segédváltozó)!

   b) Állítsa az asztal referenciát null-ra, írassa ki, nézze meg, mi történik!

   c) Deklaráljon egy fotel nevű változót, mely egy String objektumra képes mutatni, írassa ki, nézze meg, mi történik!

   d) Deklaráljon egy butor nevű változót, mely egy olyan tömb objektumra képes mutatni, melynek elemei String-ek, írassa ki, nézze meg, mi történik!

Reflexió

Az óra egyik fő célja a tömbelemek cseréjének megvalósítása volt. Ezen ismeretanyag motiválására sikerült egy olyan hétköznapi, valóságos példát kitalálnom, mely egyben realizálta másik célomat is, mégpedig azt, hogy a diákok összetett elemtípusú tömböket használjanak. Sőt, tovább megyek, nem én közöltem velük, hogy tömböt kell létrehozniuk, és azt sem, hogy milyen típusú elemekkel töltsék fel, így hozzájárultam ahhoz, hogy fejlesszem azon képességeiket, miszerint tudjanak megfelelő adatszerkezetet és adattípust választani. A példa ezen túl kiváló alkalmat teremtett a tárgyban megjelent szakirodalom ismertetésére, bemutatására, és a könyvek bevitelével valóság közelivé tettem a problémát és szó szerint el tudtuk „játszani" az algoritmust, konkrétan megvalósítottuk a cserét.

Az algoritmikus gondolkodás fejlesztésének egyik legnehezebb komponense az algoritmus felismerése és az algoritmus alkotása. Ez az esetek többségében eddig akadályokba ütközött, de ez a hétköznapi példa megtörte a jeget és képesek voltak az algoritmus megalkotására. ☺

Az eddigiek során egy létrehozott tömb elemeinek értékadás előtti kiíratásával tapasztalták meg, hogy mik az egyszerű elemtípusú tömb elemeinek alapértelmezett értékei. Most ugyanezt tettük az összetett elemtípusú tömbbel és megtudták, hogy ebben az esetben az alapértelmezett érték a null referencia.

Ezt, és a továbbiakban a tömb elemeinek feltöltését, a tömbelemek cseréjének minden egyes lépését a szemléltetés módszerével a táblán tettem érthetővé, mely tapasztalataim alapján határozottan alkalmas arra, hogy megtudják és megértsék: az objektumok „maradnak", csak a referencia állítódik át. A táblára általam felrajzolt ábrák mindegyike és természetesen a kódok is a hálózaton számukra elérhetők. Ezt minden egyes csoportomnál alkalmazom, de itt különösen fontos, tekintettel a csoport összetételére.

A kellő előkészítésnek köszönhetően két tanuló önállóan, egy tanuló egy kis segítséggel lekódolta az algoritmust.

Valószínű, hogy emiatt volt idő a második feladatra (nem voltam biztos, hogy ez belefér az órába, de azt az elvet vallom, hogy a van-tól nem kell félni, tehát inkább több feladat legyen a tarsolyomban, minthogy ott álljak üres zsebbel ☺), melyből kiderült számukra, hogy tömb referenciájának alapértelmezés szerinti értéke lokális deklaráció esetén definiálatlan.

Az óra a legjobb példája annak, hogy a megfelelően motivált diák sok mindenre képes.

ÓRATERVEZET - TÖMB ELEMEINEK MEGFORDÍTÁSA. GYAKORLÁS.

A pedagógus neve: Kovács Henriette

A szakmacsoport száma és megnevezése: 7. Informatika

A szakképesítés azonosító száma: 54 213 05

A szakképesítés megnevezése: Szoftverfejlesztő

Iskolai rendszerű szakképzés (a szakképzési évfolyamok száma: 2)

Szakmai követelménymodul: 10817-12 Hálózatok, programozás és adatbázis-kezelés

Tantárgy: Adatbázis- és szoftverfejlesztés gyakorlat

Programozási nyelv: Java

Fejlesztői környezet: NetBeans IDE 7.3

Osztály - csoport: I/13. S - SZ2

Az óra tananyaga: Tömb elemeinek megfordítása. Gyakorlás.

Az óra cél- és feladatrendszere

A fejlesztendő attitűd, készségek, képességek, a tanítandó ismeretek (fogalmak, szabályok stb.) és az elérendő fejlesztési szint, tudásszint megnevezése:

- A tanuló tartsa be a számítógépes munka szabályait, különös tekintettel a balesetek megelőzésére.
- Felkelteni és folyamatosan ébren tartani a tanulók érdeklődését a szoftverfejlesztés iránt.
- Korszerű alkalmazói készség kialakítása.
- Az algoritmikus gondolkodás fejlesztése: a tanulók önálló, rendszerezett, logikus gondolkodásának fejlesztése; a logikus gondolkodás lényeges a problémamegoldásban, és az algoritmusok elkészítésében, kódolásában.
- A feladatmegoldáshoz megfelelő adattípus kiválasztásának fejlesztése.
- Legyen képes a NetBeans fejlesztői környezet kényelmi funkcióinak készségszintű használatára.
- Önálló munkára nevelés és felzárkóztatás.
- Alkotó munkára nevelés: a végeredmény egy új termék, egy program lesz.
- Az esztétikai készség fejlesztése: igény és készség a számítógépes produktum esztétikus formájának kialakítására, a program helyes strukturálására.
- *Tudjon tömb típusú változót deklarálni, létrehozni, feltölteni.*
- *Tudjon tömböt megjeleníteni.*
- *Legyen képes a tömbök elemeit elérni, feldolgozni és velük műveleteket végezni.*
- *Tudjon tömböt létrehozni inicializáló blokk segítségével.*
- *Legyen képes a tömbök elemeit absztrakt módon elérni.*
- *Tudjon tömbelemeket cserélni.*

Fejlesztendő készségek, képességek

- Visszaemlékezés, algoritmikus gondolkodás, logikus gondolkodás, absztrakt gondolkodás, problémamegoldás, szabályalkotás, becslés, képletalkotás, algoritmus megvalósítása, struktogram elkészítése, precizitás, pontosság, az absztrakciók, általánosítások felhasználása, önellenőrzés, kód elkészítése, kommentek használata, szintaktika követése, futási eredmény elemzése, követhetősége, átláthatósága.

Fejlesztendő attitűd

- Legyen képes: számítógépet kezelni, szoftvereket használni.
- Tudja önállóan használni a hálózatot és annak alapszolgáltatásait.

– Legyen képes az adott probléma megoldásához kiválasztani az általa ismert módszerek, eszközök és alkalmazások közül a megfelelőt.
– Ismerje fel és legyen képes különféle formákban megfogalmazni a környezetében előforduló tevékenységek algoritmizálható részleteit.
– Legyen képes a problémamegoldáshoz szükséges adatok és az eredmény kapcsolatának megtervezésére, értelmezésére.
– Adott feladat megoldásához legyen képes algoritmusokat megfogalmazni, tervezni és megvalósítani.
– Legyen képes alkalmazást (szoftvert) tervezni, fejleszteni, futtatni, tesztelni és magyarázó megjegyzésekkel ellátni, előkészítve ezzel a dokumentáció elkészítését.
– Legyen képes programozási feladatot ellátni, egy szoftvert elkészíteni (kódolni), Java programnyelven implementálni.
– Legyen képes értelmezni és javítani a NetBeans által megjelenített hibaüzeneteket.
– Legyen képes programjait úgy megírni, hogy azt később Ő maga is, vagy tanulótársa is fel tudja használni.

Az óra didaktikai feladatai

- Ismétlés
- Megerősítés
- Motiváció
- A figyelem felkeltése
- Feladat kitűzése
- Aktivizálás
- Fokozatosság
- Szemléletesség
- Következtetés
- Összegzés
- Általánosítás
- A tények elemzése, összefüggések feltárása
- Visszacsatolás
- Összehasonlítás
- Ellenőrzés
- Értékelés
- Figyelemfelhívás

Tantárgyi kapcsolatok: Adatbázis- és szoftverfejlesztés elmélet, Matematika, Angol, Hálózati ismeretek I., Informatikai alapismeretek, Testnevelés

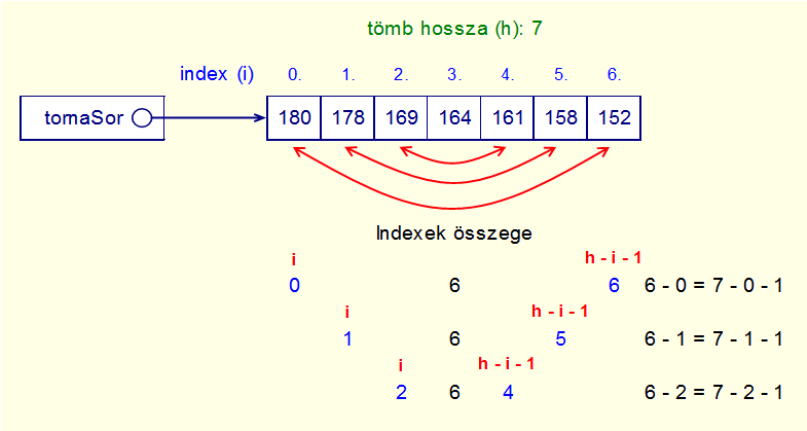
Felhasznált források (tankönyv, munkafüzet, feladat- és szöveggyűjtemény, digitális tananyag, online források, szakirodalom stb.):

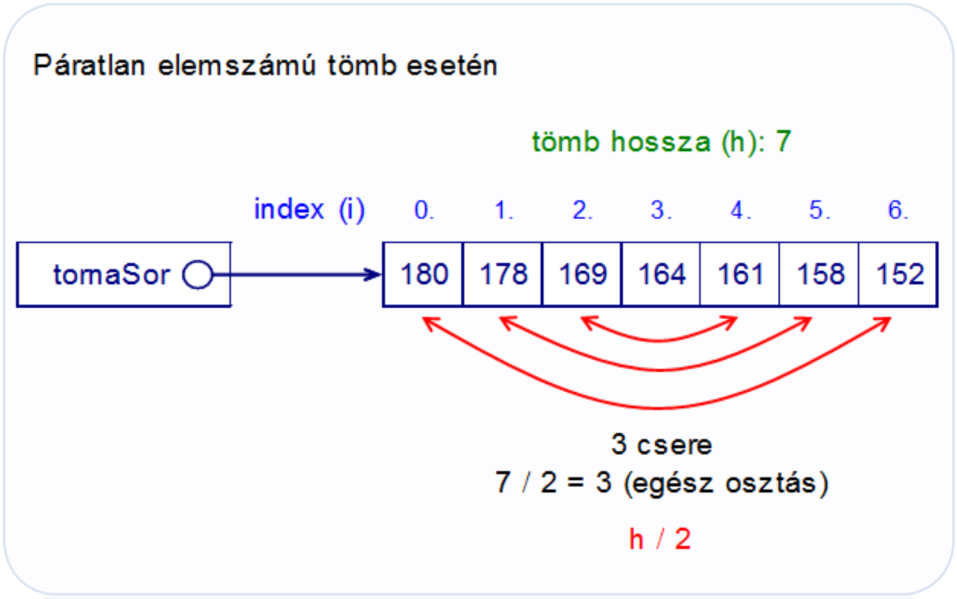
Szerző(k)	Cím	Kiadó	ISBN kód
Vég Csaba – dr. Juhász István	Java – start!	Logos 2000	963 03 9005 1
Vég Csaba	Instant Java/Java EE/SOA I-II	Logos 2000	978-963-06-2601-9 978-963-06-2602-6
Dirk Louis – Peter Müller:	Java 5 Belépés a programozás világába	Panem	963 545 454 6
Benkő Tiborné, Tóth Bertalan	Együtt könnyebb a programozás Java	Computerbooks	963 618 337 6
Rogers Cadenhead	Tanuljuk meg a Java programozási nyelvet 24 óra alatt	Kiskapu Kft.	963 9637 07 6
Angster Erzsébet	Objektumorientált tervezés és programozás Java 1 - 2	4KÖR Bt.	963 00 6262 3 963 00 6263 1
Peter J. DePasquale	Java zsebkönyv	Kiskapu Kft.	963 9637 10 6
Nyékyné Gaizler Judit, Frohner Ákos, Kozsik Tamás (Szerkesztők)	Java 2 útikalauz programozóknak	ELTE TTK Hallgatói Alapítvány	963 463 305 6
Kathy Sierra, Bert Bates	Agyhullám: Java	Kiskapu Kft.	978 963 9637 79 5

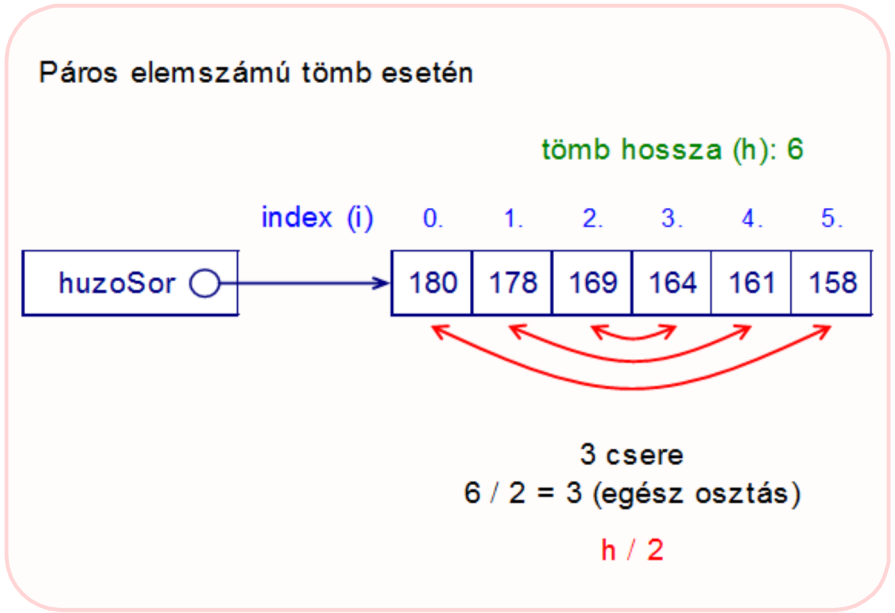
http://docs.oracle.com/javase/7/docs/api/

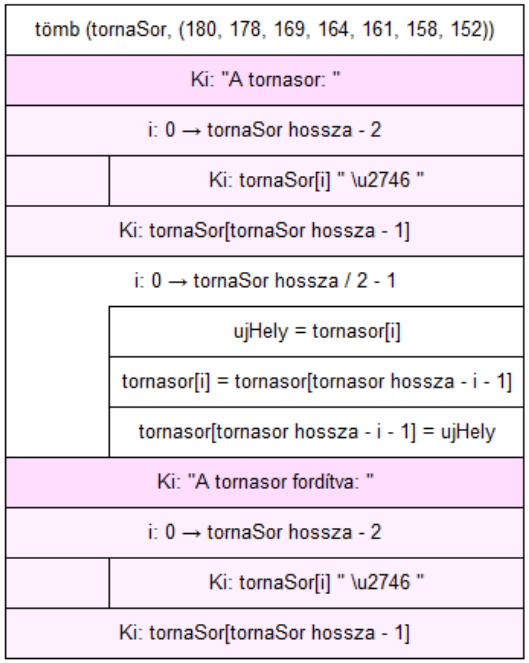
Dátum: éééé. hh. nn.

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MÓDSZEREK TANULÓI MUNKAFORMÁK ESZKÖZÖK MEGJEGYZÉSEK
0 – 4’	Ismétlés: Tömb deklarálása, létrehozása, elemeinek alapértelmezett értékei. Tömb elemeinek megjelenítése. Tömbelemek cseréje. A mai órán ezzel kapcsolatosan egy nagyon érdekes feladatot valósítunk meg. Tegnap, ahogy a tornaterem előtt elmentem, nyitva volt az ajtó és láttam, hogy pont akkor tornasorban sorakoztak a gyerekek. Eszembe jutott, hogy amikor én jártam iskolába az egyik barátnőm volt a legkisebb és mindig a tornasor végén állt. Soha nem értette, miért nem lehetne legalább egyszer fordítva állni. Váltsuk valóra álmát! Feladat: 1. Egy csoport 7 diákját állítsa fordított tornasorba! A diákokat milyen jellemzőjükkel azonosítaná? Jelenítse meg a változást! Gondoljuk át, hogyan tudnánk megvalósítani a feladatot!	Ismétlés Megerősítés Visszaemlékezés Motiváció A figyelem felkeltése Feladat kitűzése Logikus gondolkodás	Megbeszélés Frontális munka Hálózat Projektor

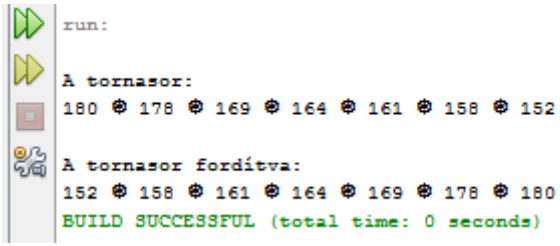
IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
4 – 6’	Milyen adatszerkezettel valósítjuk meg a feladatot? Válasz: tömb adatszerkezettel valósítjuk meg. Milyen tulajdonságukkal azonosítjuk a diákokat? magasság – cm Melyik módszerrel hozzuk létre a tömböt? inicializáló blokkal. Milyen típusú elemekből áll? egész	Aktivizálás Absztrakt gondolkodás Fokozatosság	Vita Megbeszélés Frontális munka
6 – 11’	Hogyan tudom megfordítani a tornasort?  Mit gondolnak, melyik az én adatom a tömbben? Egy cserét két tanuló eljátszik. Tehát a tömb i. és (h – i - 1). indexű elemeit kell kicserélni. Mely elemek ezek? tornaSor[i] és tornaSor[h – i – 1]	Problémamegoldás Szemléletesség Logikus gondolkodás Algoritmikus gondol-kodás Absztrakt gondolkodás Szabályalkotás Becslés Következtetés Összegzés Általánosítás	Szemléltetés Vita Magyarázat Frontális munka Tábla Hálózat Szerepjáték

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
11 – 13’	Hány cserét kell végezni? Mindegy, hogy páratlan vagy páros elemszámú a tömb?  Páratlan elemszámú tömb esetén (tömb hossza / 2) cserét kell végezni.	Fokozatosság A tények elemzése, összefüggések feltárása Szemléletesség Visszacsatolás Következtetés Képletalkotás	Szemléltetés Vita Frontális munka Magyarázat Tábla Hálózat

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
13 – 15’	 Páros elemszámú tömb esetén (tömb hossza / 2) cserét kell végezni. Páratlan és páros elemszámú tömb esetén is (tömb hossza / 2) cserét kell végezni.	Aktivizálás Összehasonlítás Fokozatosság Következtetés Képletalkotás Összegzés	Frontális munka Előző ábra projektorral kivetítve, ez az előzőt módosítva, a Tábla Frontális munka Megbeszélés Magyarázat

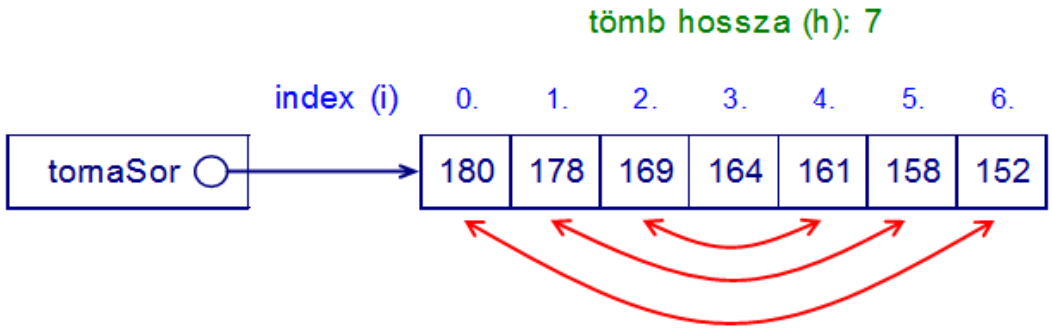
IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
15 – 28’	Struktogram elkészítése:  A hasonló vagy azonos részeket másolja és javítsa!	Algoritmikus gondolkodás Algoritmus megvalósítása Struktogram elkészítése Precizitás, pontosság Az absztrakciók, általánosítások felhasználása Önellenőrzés Ellenőrzés	Önálló munka nehézségek esetén Irányított, önálló munka (A struktogramban a tömb létrehozását, és az 1. kiíratást, ha úgy látom jónak készen megkapják.) Számítógép Structorizer Microsoft Office Excel 2007, 2010 Frontális munka

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
28 – 40’	Programkód: **package gyakorlat13;** **/**** ***** *** @author Kovács Henriette** ***/** **public class Gyakorlat13_10 {** **public static void main(String[] args) {** **// tömb létrehozása, feltöltése** **int[] tornasor = {180, 178, 169, 164, 161, 158, 152};** **//tömb kiíratása** **System.out.println("\nA tornasor: ");** **for (int i = 0; i < tornasor.length - 1; i++) {** **System.out.print(tornasor[i] + " \u2746 ");** **}** **System.out.println(tornasor[tornasor.length - 1]);** **//tömbelemek sorrendjének megforditása** **for (int i = 0; i < tornasor.length / 2; i++) {** **int ujHely = tornasor[i];** **tornasor[i] = tornasor[tornasor.length - i - 1];** **tornasor[tornasor.length - i - 1] = ujHely;** **}** **//tömb kiíratása fordított sorrendben** **System.out.println("\nA tornasor fordítva: ");** **for (int i = 0; i < tornasor.length - 1; i++) {** **System.out.print(tornasor[i] + " \u2746 ");** **}** **System.out.println(tornasor[tornasor.length - 1]);** **}** **}**	Kód elkészítése Kommentek használata Szintaktika követése Precizitás, pontosság Az absztrakciók, általánosítások felhasználása Önellenőrzés Ellenőrzés	Önálló munka Számítógép NetBeans IDE 7.3 Java 1.5 (Ha nem tudnák befejezni a kódolást, akkor vasárnapi határidővel házi feladatnak feladom.) Megbeszélés Projektor Frontális munka

IDŐKERET	AZ ÓRA MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
40 – 41’	Futási eredmény:  Ellenőrzés – közben is folyamatosan követtem munkájukat, ha szükséges segítek.	Futási eredmény elemzése, követhetősége, átláthatósága Ellenőrzés	Projektor Frontális munka
41 – 43’	Értékelés	Értékelés	Frontális munka
43 – 45’	Tájékoztatom őket, hogy jövő héten az eddigiek számonkérése lesz és az eddigi elméleti anyagról egy Tomb.pdf file-t raktam fel a hálózatra.	Figyelemfelhívás	Tanári közlés Tomb.pdf

TÁBLAKÉP I.

Tömbelemek cseréje



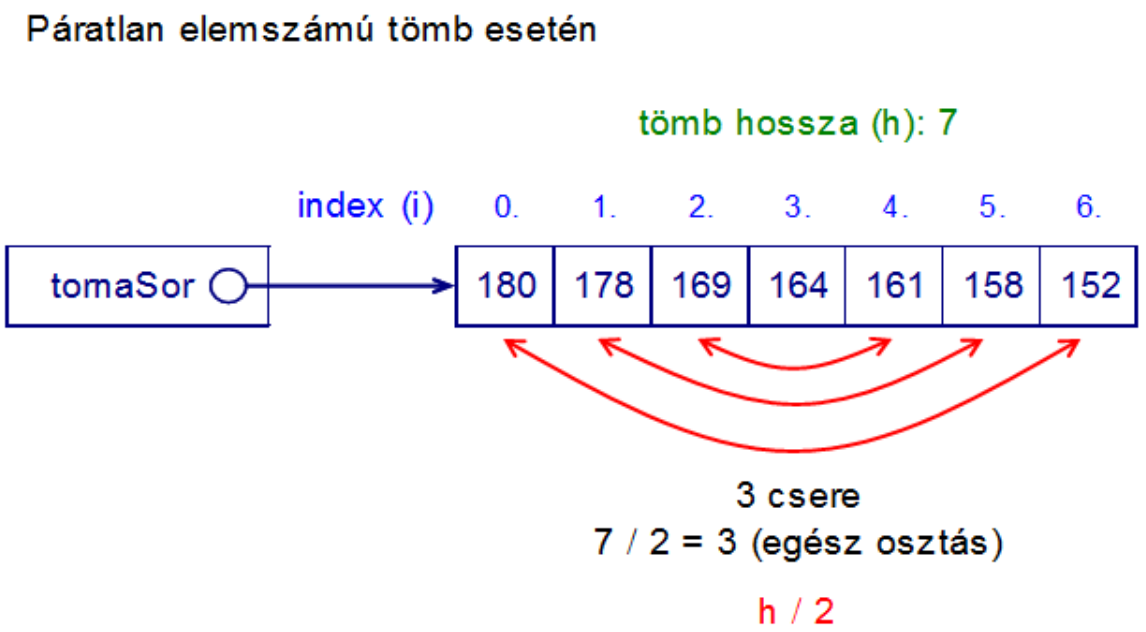
Indexek összege

i		h - i - 1	
0	6	6	6 - 0 = 7 - 0 - 1
i		h - i - 1	
1	6	5	6 - 1 = 7 - 1 - 1
i		h - i - 1	
2	6	4	6 - 2 = 7 - 2 - 1

tornaSor[i] és tornaSor[h – i – 1] elemek cseréjét kell elvégezni

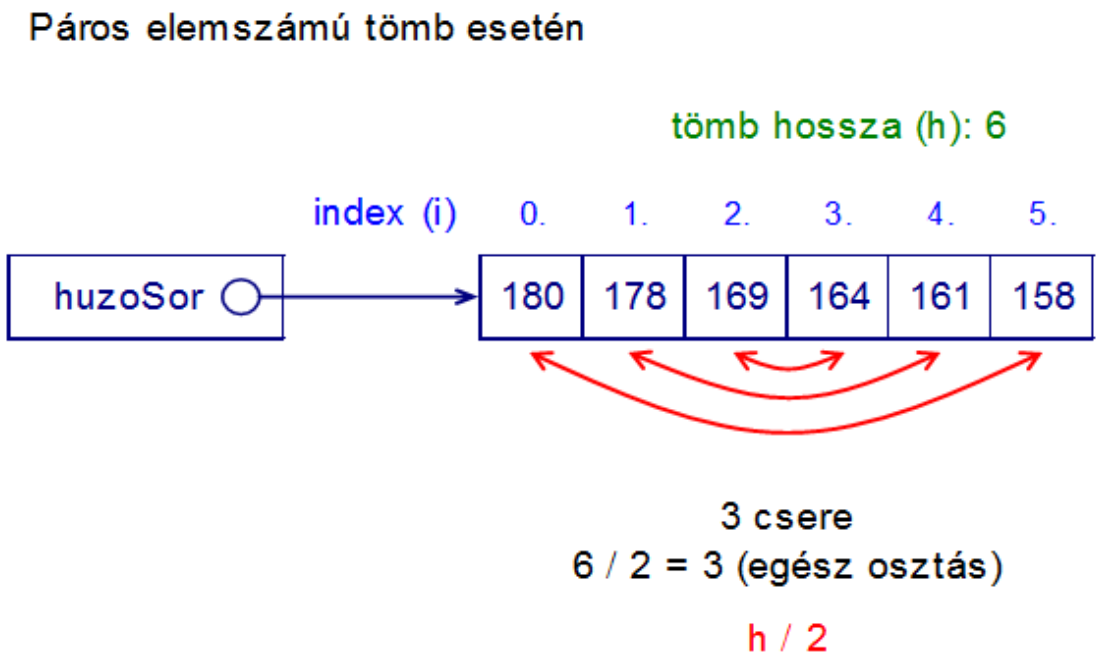
TÁBLAKÉP II.

Cserék száma



TÁBLAKÉP III.

Cserék száma



Feladat

Egy csoport 7 diákját állítsa fordított tornasorba! A diákokat milyen jellemzőjükkel azonosítaná? Jelenítse meg a változást! (Ha szükséges, a lehető legáltalánosabban alkosson szabályokat!)

A feladathoz készítsen algoritmust!

Tomb.pdf

Fix elemszámú, azonos típusú elemekből álló összetett adatszerkezet.

Az elemek

- indexelhetők, az index minden esetben 0-ról induló egész szám.
- lehetnek primitív vagy referencia típusúak.
- az elemek deklarált típusa azonos, de a referencia típusú változók utód típusú objektumokat is azonosíthatnak.

(Alacsonyszintű konténer, melynek csak adatai vannak, funkciói nincsenek.)

Használatában az a nagyszerű, hogy nem kell az elemeknek egyedi neveket kitalálni, hanem egy gyűjtőnevet adunk az egész csoportnak, és az egyes elemekre indexekkel hivatkozunk.

A tömb referencia típusú változó, melyet **deklarálnunk** kell.

1. elemek típusa
2. tömbképző operátor: szögletes zárójelpár
3. tömb azonosítója

ide is tehetjük!

```
<elemtípus>[] <tömbAzonosító>;
<elemtípus> <tömbAzonosító>[];
```

pl.:
```
int[] iArray1;
int iArray2[];
```

Az így deklarált változó képes egy megadott elemtípusú tömbre mutatni.

A változóban az objektum memóriabeli címét tároljuk.
Deklaráció hatására nem jön létre objektum.

Feladat: Melyik tömb a következő közül?

```
int[] iArray1, iArray2;
int iArray3[], iArray4;
```

A deklarálással csak a referenciának foglalunk helyet, ezen a ponton még nincs tömb.

A tömböt külön létre kell hoznunk!

new operátorral hozhatjuk létre a program futása közben

1. new operátor
2. elemek típusa
3. tömb mérete (elemek száma): nem negatív, int típusú kifejezés.
   maximum `Integer.MAX_VALUE` eleme lehet e tömbnek.

```
new <elemtípus>[<méret>];
```

Feladat: Mennyi a mérete, mi az elemtípusa?

```
new int[5];                 // 5, int
int k = 20;
new double [k/3+1];         // 7, double
new String[3];              // 3, String
```

`new` operátor: létrehozza a tömböt a memóriában

visszaadja a tömb referenciáját (memóriabeli mutatóját)

Ezt a referenciát aztán értékül adhatjuk egy olyan referencia típusú változónak, amely egy ilyen tömbre képes rámutatni.

```
int[] iArray;
double[] dArray;
String[] sArray;
iArray = new int[5];
dArray = new double[k/3+1];
sArray = new String[3];
int[] iArray = new int[5];
double[] dArray = new double[k/3+1];
String[] sArray = new String[3];
```

deklaráció

létrehozás

1 lépésben is elvégezhetők

Minden tömb objektumnak van egy length konstansa, amely megadja a tömb hosszát.

```
<tömbAzonosító>.length
```

A length értéke a tömb létrehozásától állandó, nem lehet megváltoztatni.

A tömb indexei 0 és length-1 közöttiek.

A tömb egyes elemeire így hivatkozhatunk:

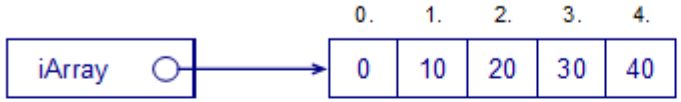
```
<tömbAzonosító>[index]
```

Az érvényes indextartományon kívül eső indexelés

`ArrayIndexOutOfBoundsException` kivételt vált ki.

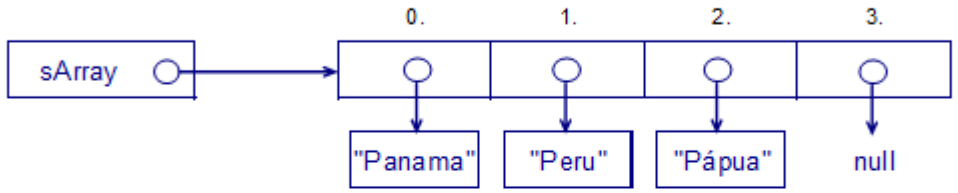
Tömb elemeinek feltöltése:

```
int[] iArray = new int[5];          //0, 10, 20, 30, 40-nel
for (int i = 0; i < iArray.length; i++)
        iArray[i] = i * 10;
```




```
 ---------------------------------------
double[] dArray = new double[20];   // beolvasás konzolról
for (int i = 0; i < dArray.length; i++){
        System.out.println("Írd be a tömb" + (i + 1) + ". elemét!");
        dArray[i] = billentyuzet.nextDouble();
        billentyuzet.nextLine();
}
```

```
String[] sArray = new String[4];    //Az elemek referenciák
sArray[0] = "Panama";
sArray[1] = "Peru";
sArray[2] = "Pápua";
```



Inicializáló blokk:

A tömb elemeinek kezdeti értékek adhatók

```
<elemtípus>[] <tömbAzonosító> = {<érték0>, <érték1>,…};
```

A tömb mérete akkora lesz, amennyi a felsorolt értékek száma.

```
int[] iArray = {0, 10, 20, 30, 40};          //5 elemű
char[] mgh = {'a', 'e', 'i', 'o', 'u'};      //5 elemű
boolean[] szep = {true, false};              //2 elemű
```

Kezdeti értékek:

A tömb referenciájának alapértelmezés szerinti értéke

osztályszintű deklaráció esetén: **null**

lokális deklaráció esetén: **definiálatlan**

Az újonnan létrehozott tömb elemei alapértelmezés szerinti értékeket vesznek fel.

referenciák esetén: **null**

primitív típusú változók esetén: **0**, **\u0000**, **false**

Értékadás:

A `t2` tömb értékadás szerint kompatibilis a `t1` tömbbel (`t1 = t2` megengedett), ha

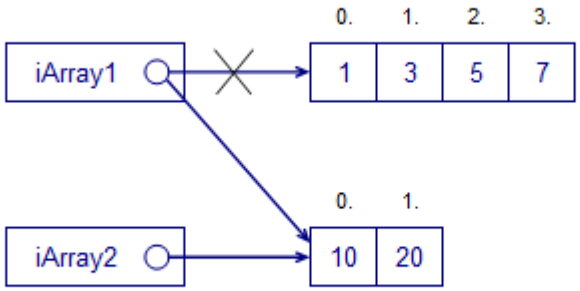
- primitív elemtípus esetén `t1` és `t2` elemtípusa azonos
- referencia elemtípus esetén

  `t2` elemtípusa `t1` elemtípusával azonos, vagy annak leszármazottja.

Tömb értékadásakor a tömb referenciája kap értéket, nem a tömb elemei. Így különböző hosszúságú tömbök is értékül adhatók egymásnak.

pl.:

```
int[] iArray1 = {1, 3, 5, 7};
int iArray2[] = {10, 20};
iArray1 = iArray2;        //{10, 20}
```



Mindkét mutató ugyanazt a tömböt azonosítja.

```
double[] dArray1 = {1, 2, 3, 4, 5, 6, 7.0}, dArray2 = {1.5, 6.0};
Object[] oArray = null;
String[] sArray = {"hétfő", "kedd"};
dArray1 = dArray2;          //{1.5, 6.0}
oArray = sArray;            //{"hétfő", "kedd"}
```

Helytelen értékadások:

```
iArray1 = dArray1;          //nem egyforma primitív típusú elemek
sArray = dArray1;           //egyik elemei objektumok, másiké primitívek
sArray = oArray             //Object-nek nem őse a String!
```

Értékadáskor a tömb nem másolódik, hanem átállítódik a mutatója.

Reflexió

Az első óra volt a téma elkezdése óta, mely tisztán gyakorló óra és új ismeretek elsajátítására nem került sor. Ezek az órák - így ez is - igazán alkalmasak az addig megtanított ismeretek rögzítése, elmélyítése, megerősítése mellett a logikus gondolkodás, az algoritmikus gondolkodás és az absztrakt gondolkodás fejlesztésére. Központi szerepet töltött be az órán a következtetés, az absztrakció, az általánosítás, mint didaktikai feladat megvalósulása, mely maga után vonta az összefüggések felismerésének és a szabályalkotás képességének fejlesztését.

A bevezető elbeszélésem megfelelően motiválta a diákokat, mert egy konkrét tevékenység megvalósítása, egy tényleges cél elérése mindig sokkal jobban felkelti érdeklődésüket és úgymond értelmét látják, mintha pusztán csak annyit mondtam volna, hogy „fordítsuk meg egy tömb elemeit”. A figyelem felkeltése - és ezen túl - a figyelem ébren tartása, az egyik legfontosabb feladatok közé tartozik, melyek az órán megvalósultak. A csoport kis létszáma és a csoportprofilban leírt okok miatt változatos munkaformák alkalmazhatósága korlátozott, de érdekes oktatási módszereket, mint pl. vita, játék ebbe az órába be tudtam vonni. Emellett nagy erőfeszítéseket tettem a szemléletesség, mint didaktikai feladat megvalósítására, mely nemcsak az adott probléma megértését, hanem a figyelem ébren tartását és ezen felül a következtetések, általánosítások, absztrakciók megkönnyítését is szolgálták.

A tábla, a számítógép és a projektor elengedhetetlenek voltak és azt szolgálták, hogy a diákok számára élvezetessé tegyem az órát. A lényeges, magyarázó, a megértést szolgáló táblai képeket elektronikusan is elkészítettem, így a tömbelemek cseréje és a cserék számával kapcsolatos képek is nagymértékben hozzájárultak a szabályok, képletek, összefüggések feltárásához, megértéséhez, segítették a logikus, absztrakt, valamint algoritmikus gondolkodás fejlesztését és mindemellett lehetővé teszik - szükség esetén - az összefüggések újbóli bemutatását, elmélyítését.

A megbeszéltek alapján a struktogram elkészítését önálló feladatnak szántam, de éreztem, hogy valószínű, hogy ez kisebb akadályokba ütközik, ezért is tüntettem fel az óratervben és sejtésem beigazolódott. A programkód írását viszont szépen teljesítették, használva a már megírt kódrészleteket, és a futási eredmény átláthatóan jelent meg a képernyőn.

Az órát eredményesnek ítélem meg, a diákok képességei fejlődtek, az ismeretek elmélyültek.

FOGLALKOZÁS TERVEZETE – AZ ISMÉTLÉS - ITERÁCIÓ. CIKLUSOK FAJTÁI, ÁTALAKÍTHATÓSÁGUK, ALKALMAZÁSUK.

A pedagógus neve: Kovács Henriette

Az ágazat megnevezése - besorolása: Informatika és távközlés - 12

A szakma azonosító száma: 5 0613 12 03

A szakma megnevezése: Szoftverfejlesztő és -tesztelő

Szakmai oktatás időtartama: 2 év

A szakma Európai Képesítési Keretrendszer szerinti szintje: 5

A szakma Magyar Képesítési Keretrendszer szerinti szintje: 5

Digitális Kompetencia Keretrendszer szintje: 8

Tantárgy: Programozási alapok

Programozási nyelv: Python

Fejlesztői környezet: Visual Studio Code

Osztály: 1/13. RL

A foglalkozás tananyaga: Az ismétlés - iteráció. Ciklusok fajtái, átalakíthatóságuk, alkalmazásuk.

A foglalkozás cél- és feladatrendszere

A fejlesztendő attitűd, készségek, képességek, a tanítandó ismeretek (fogalmak, szabályok stb.) és az elérendő fejlesztési szint, tudásszint megnevezése:

- A tanuló tartsa be a számítógépes munka szabályait, különös tekintettel a balesetek megelőzésére.
- Felkelteni és folyamatosan ébren tartani a tanulók érdeklődését a programozás iránt.
- Korszerű alkalmazói készség kialakítása.
- Az algoritmikus gondolkodás fejlesztése: a tanulók önálló, rendszerezett, logikus gondolkodásának fejlesztése; a logikus gondolkodás lényeges a problémamegoldásban, és az algoritmusok elkészítésében, kódolásában.
- A feladatmegoldáshoz megfelelő adattípus kiválasztásának fejlesztése.
- Legyen képes a Visual Studio Code fejlesztői környezet kényelmi funkcióinak készségszintű használatára.
- Önálló munkára nevelés és felzárkóztatás.
- Alkotó munkára nevelés: a végeredmény egy új termék, egy program lesz.
- Az esztétikai készség fejlesztése: igény és készség a számítógépes produktum esztétikus formájának kialakítására, a program helyes strukturálására.
- *Tudjon változókat azonosítani, értékadást végezni.*
- *Tudjon kifejezéseket az aritmetikai, logikai, relációs, értékadó operátorok felhasználásával létrehozni.*
- *Tudjon a print() függvénnyel, annak paraméterezésével adatokat megjeleníteni.*
- *Tudja használni az input() függvényt és a típuskonverziót.*
- *Tudja alkalmazni az escape karaktereket, megjegyzéseket.*
- *Tudjon egy- és többágú elágazásokat létrehozni és használni.*
- *Tudjon ciklusokat létrehozni, adott feladatnál célszerűen közülük választani.*
- *Legyen képes a végtelen ciklus felismerésére, elkerülésére.*
- *Tudjon a range() függvény segítségével sorozatot létrehozni.*
- *Legyen képes egy sorozat elemeit bejárni és velük műveleteket végezni.*

Fejlesztendő készségek, képességek

- Visszaemlékezés, precizitás, megfigyelés, feladatértelmezés, absztrakt gondolkodás, logikus gondolkodás, algoritmikus gondolkodás, problémamegoldás, programkód analizálása, írása, szintaktika követése, hibakeresés, tesztelés, önellenőrzés, önállóság, együttműködő képesség, meggyőzőképesség, szakmai nyelvű

beszédkészség, íráskészség, mondatszerű leírás elkészítése, rendszerező képesség.

Fejlesztendő attitűd

- Legyen képes: számítógépet kezelni, szoftvereket használni.
- Tudja önállóan használni a hálózatot és annak alapszolgáltatásait.
- Legyen képes az adott probléma megoldásához kiválasztani az általa ismert módszerek, eszközök és alkalmazások közül a megfelelőt.
- Ismerje fel és legyen képes különféle formákban megfogalmazni a tevékenységek algoritmizálható részleteit.
- Legyen képes adatok megkülönböztetésére, kezelésére, használatára.
- Legyen képes a problémamegoldáshoz szükséges adatok és az eredmény kapcsolatának megtervezésére, értelmezésére.
- Adott feladat megoldásához legyen képes algoritmusokat megfogalmazni, tervezni és megvalósítani.
- Legyen képes alkalmazást (szoftvert) tervezni, fejleszteni, futtatni, tesztelni és magyarázó megjegyzésekkel ellátni, előkészítve ezzel a dokumentáció elkészítését.
- Legyen képes programozási feladatot ellátni, egy szoftvert elkészíteni (kódolni), Python programnyelven implementálni.
- Legyen képes értelmezni és javítani a Visual Studio Code által megjelenített hibaüzeneteket.
- Legyen képes programjait úgy megírni, hogy azt később Ő maga is, vagy tanulótársa is fel tudja használni.

Az óra didaktikai feladatai

- Motiválás
- Aktivizálás
- Szemléletesség
- Ismétlés
- Visszacsatolás
- Új ismeret közlése
- A tények, jelenségek, folyamatok sokoldalú elemzése
- Rögzítés
- Gyakorlás
- Fokozatosság
- Megerősítés
- Ellenőrzés
- Értékelés
- Összefoglalás
- Rendszerezés
- A házi feladat kijelölése

Tantárgyi kapcsolatok: Matematika, Angol, Informatikai alapismeretek, Sport

Felhasznált források:

Szerző(k)	Cím
Gérard Swinnen	Tanuljunk meg programozni Python nyelven
Raphael Marvie	Bevezetés a Pythonba példákkal
Full Circle magazin	Programozzunk Pythonban
Takács Gábor	A Python programozási nyelv - jegyzet

https://github.com/nlharri/Pythonarium/blob/master/Documentation/hu/01_introduction.md

https://infopy.eet.bme.hu/

https://www.w3schools.com/python/

https://pythonidomar.wordpress.com/python-parancsok-magyar-jelentes/

https://www.tutorialspoint.com/python/

http://alea.uw.hu/c/Progr_tetelekRendezesek.htm

https://medium.com/pythonarium/halad%C3%B3-python-els%C5%91-r%C3%A9sz-cb0a898faa2e

https://gyires.inf.unideb.hu/EFOP344/PythonHTML/conditionals.html

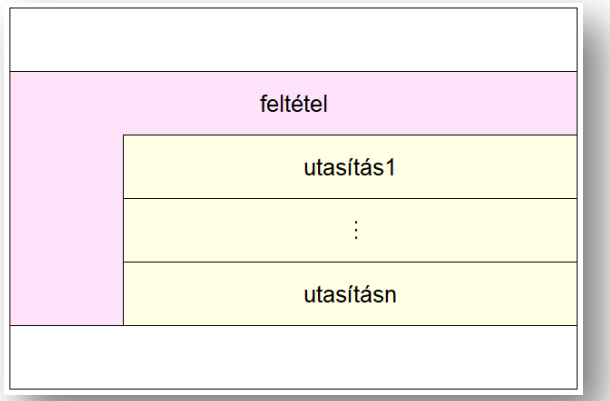
https://sulipython.fandom.com/hu/wiki/Ciklus?action=edit&section=1

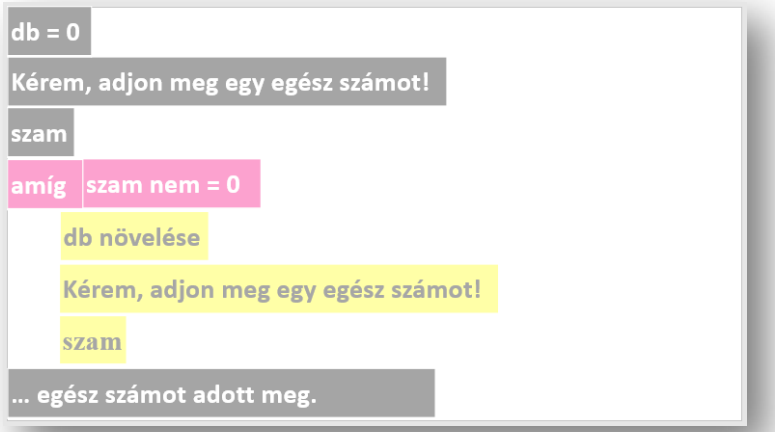
http://slpython.blogspot.com/2016/03/05elagazasokciklusokmd.html

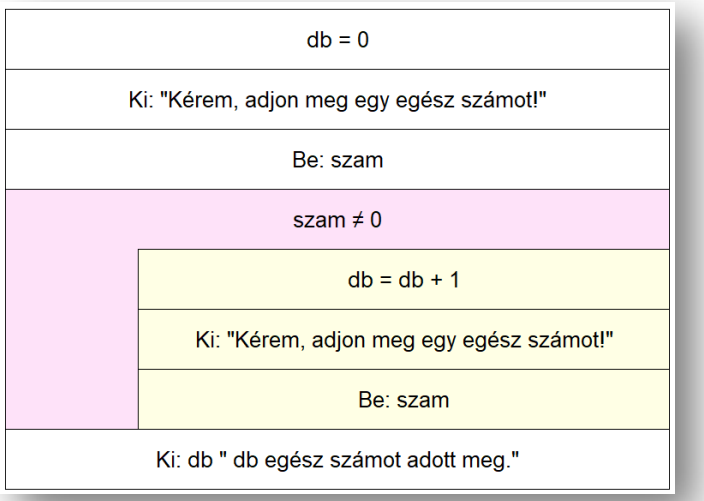
http://people.ubuntu.com/~kelemeng/.ufp3/native-datatypes.html

Dátum: éééé. hh. nn.

IDŐKERET	A FOGLALKOZÁS MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MÓDSZEREK TANULÓI MUNKAFORMÁK ESZKÖZÖK MEGJEGYZÉSEK

0 – 5’	Tanulók üdvözlése ☺, névsor olvasása – kártyák kiosztásával
	Házi feladatok, szorgalmi feladatok elkészítésével kapcsolatos észrevételek
	Ismétlés:
	Ciklus szükségessége, létrehozása.
	Ciklus működésének elve.
	Ciklus kulcsszava, a blokk lefutásának feltétele.
	
	
	`while feltétel:`
	`  utasítás1`
	`  utasítás2`
	`  utasításn`
		Ismétlés
		Megerősítés
		Visszaemlékezés
		Szemléletesség
		Esztétikai érzék
			Megbeszélés
			Frontális munka
			Struktogramok_alapok.xlsx
			Projektor
			Python_07_het_foglalko-zas.pdf

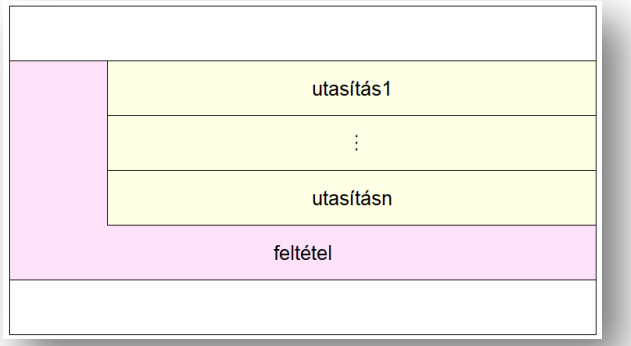
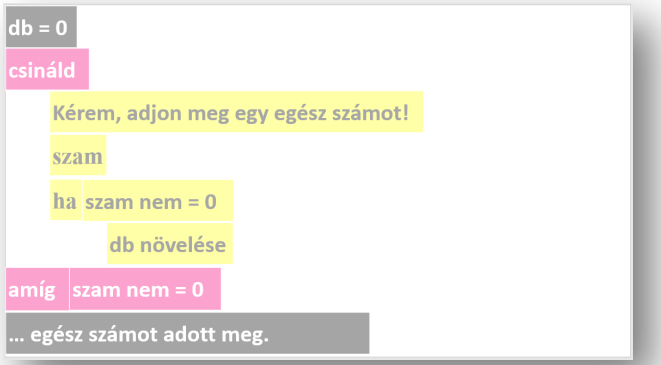
IDŐKERET	A FOGLALKOZÁS MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
6 – 10’	Alapfeladat: Olvasson be egész számokat mindaddig, amíg 0-t nem adnak meg! Ha 0-t adnak, írja ki, hogy hány darab szám került beolvasásra a 0 kivételével, és vége a feldolgozásnak! *Más megfogalmazásban:* Olvasson be egész számokat 0 végjelig, majd írja ki a beolvasott értékek számát! A végjel nem eleme a sorozatnak! A feladat eljátszása Algoritmizálás mondatszerű leírással (PowerPoint használatával):  	Feladat kitűzése Motiváció A figyelem felkeltése Esztétikai érzék Logikus gondolkodás	Hálózat Projektor Megbeszélés Frontális munka Játék Csoportmunka Ciklus0_Alapfeladat_megoldas.pptx COVID miatt lehet: Önálló munka helyzettől függően lehet: Frontális munka

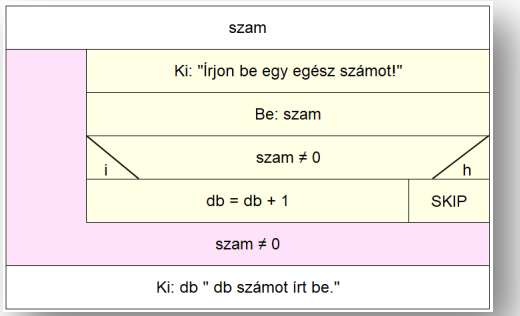
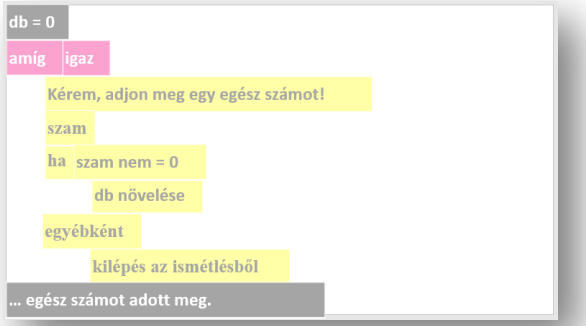
IDŐKERET	A FOGLALKOZÁS MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
10 – 13’	A db változó (üres a kártyán kezdetben) hova kell, hogy kerüljön? Melyik kódrészlet ismétlődik? Miért van szükség arra, hogy a ciklus előtt is bekérjük a változót? *Válasz: A feltétel a felhasználó által megadott számot vizsgálja.* Mikor fut le a ciklus blokkja? *Válasz: Ha a ciklus feltétele igaz* Hányszor ismétlődik a ciklusmag (sárga szín)? *Válasz: Nem tudjuk megmondani.* Struktogrammal: 	Aktivizálás Fokozatosság Visszacsatolás Problémamegoldás Szemléletesség Logikus gondolkodás Algoritmikus gondol-kodás Absztrakt gondolkodás	Vita Megbeszélés Frontális munka Szemléltetés Struktogramok_alapok.xlsx Önálló munka

IDŐKERET	A FOGLALKOZÁS MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK

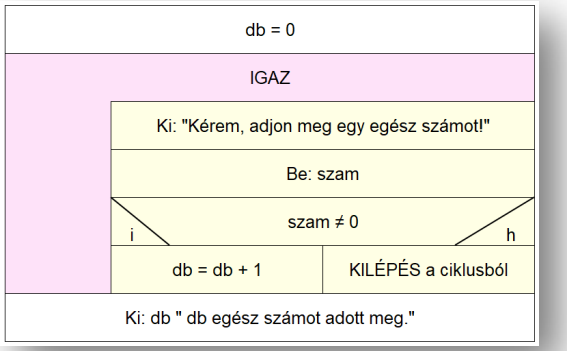
13 – 18	Forráskód elkészítése:
	Kiíratás és bekérés egymásba ágyazása – paraméter
	`db = 0`
	`szam = int(input("Kérem, adjon meg egy egész számot! "))`
	`while szam != 0:`
	`    db += 1`
	`    szam = int(input("Kérem, adjon meg egy egész számot! "))`
	`print(db, "db egész számot adott meg.")`
		Kód elkészítése
		Ismétlés
		
		Kommentek használata
		Szintaktika követése
		Precizitás, pontosság
		Önellenőrzés
		Futási eredmény elemzése, követhetősége, átláthatósága
			Önálló munka
			
			Visual Studio Code

18 – 20’	Keressünk egy egyszerűbb módszert adatok bekérésére, ne kelljen kétszer ugyanazt leírnunk. Ha a feltétel a ciklus blokkja után lenne, nem lenne szükség a ciklus előtti bekérésre. Ez az ún. hátultesztelő ciklus, melynek blokkja egyszer mindenképpen lefut.	Problémamegoldás Szemléletesség Új ismeret közlése	Számítógép Viszal Studio Code Python 3.

IDŐKERET	A FOGLALKOZÁS MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
		Szemléletesség Esztétikai érzék	Struktogramok_alapok.xlsx
	Algoritmizálás mondatszerű leírással 	Szemléletesség Megfigyelés Fokozatosság	Szemléltetés Tanári bemutatás Magyarázat Frontális munka

IDŐKERET	A FOGLALKOZÁS MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
	Struktogrammal:  	Szemléletesség	Szemléltetés Tanári bemutatás Magyarázat Struktogramok_alapok.xlsx
20 – 23’	Pythonban nincs. Oldjuk meg, hogy ne használjuk a feltételben a bekért számot! – Végtelen ciklus  	Új ismeret közlése Visszacsatolás Ismétlés A szükséges előzetes ismeretek felidézése az új ismeretek elsajátításához	Magyarázat Megbeszélés Szemléltetés

IDŐKERET	A FOGLALKOZÁS MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK

	Struktogrammal:
	
	
	Alakja:
	`while True:`
	`  utasítás`
	`  if feltétel:`
	`  utasítás`
	`while True:`
	`  utasítás1`
	`  utasítás2`
	`  if feltétel:`
	`  utasítás`
	`  utasításn`
		Szemléletesség
		Új ismeret közlése
		Absztrakció
			Magyarázat
			Struktogramok_alapok.xlsx
			Python_07_het_foglalko-zas.pdf

23 – 28’	Forráskód elkészítése:
	`db = 0`
	`while True:`
	`  szam = int(input("Kérem, adjon meg egy egész számot! "))`
	`  if szam != 0:`
	`  db += 1`
	`  else:`
	`  break`
	`print(db, "db egész számot adott meg.")`
	Hogyan tudjuk kikerülni a break-et?
	Logikai változó bevezetésével.
		Megvalósítás
		Kódrészlet írása
		Fokozatosság
			Számítógép
			Önálló munka
			Visual Studio Code

IDŐKERET	A FOGLALKOZÁS MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK

	``
	db = 0
	szeretnenkUjBekerest = True
	while szeretnenkUjBekerest:
	    szam = int(input("Kérem, adjon meg egy egész számot! "))
	    if szam != 0:
	        db += 1
	    else:
	        szeretnenkUjBekerest = False
	print(db, "db egész számot adott meg.")
	``
		Megvalósítás
		Programkód átalakítása
			Frontális munka
			Visual Studio Code

28 – 29’	Mikor használjuk a True logikai értéket tartalmazó elöltesztelő ciklust? *Adott tulajdonságú adatok bekérésére nagyon hasznos.*	Rendszerezés Aktivizálás Megerősítés	Beszélgetés
29 – 30’	1. feladat - Python_07_het_foglalkozas.pdf 21. feladata Írjon programot, ami bekér a felhasználótól egy pozitív számot! Ha a felhasználó nem a feltételnek megfelelő számot ad meg, ismételje meg a bekérést! Ha a szám megfelelő, írja ki a képernyőre a szám 5-ös maradékát!	Gyakorlás Fokozatosság	Python_07_het_foglalkozas.pdf

30 – 35’	Megfelelő típuskonverzió alkalmazása
	Kód írása:
	``
	while True:
	    szam = int(input("Kérem, adjon meg egy pozitív számot! "))
	    if szam > 0:
	        print("A/az {} ötös maradéka: {}".format(szam, szam % 5))
	        break
	``
	Futási eredmény, hibajavítás
		Megfelelő adattípus kiválasztása
		Visszacsatolás
		Önellenőrzés
			Megbeszélés
			Önálló munka
			Visual Studio Code
			Frontális munka

IDŐKERET	A FOGLALKOZÁS MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK

35 - 38	Alakítsuk át úgy a kódot, hogy a felhasználónak jelezzük, ha nem megfelelő számot adott meg!
	Kód másolása, átalakítása, futási eredmény
	`while True:`
	`  szam = int(input("Kérem, adjon meg egy pozitív számot! "))`
	`  if szam <= 0:`
	`  print("Nem pozitív számot adott meg!")`
	`  else:`
	`  print("A/az {} ötös maradéka: {}".format(szam, szam % 5))`
	`  break`
		Kód átalakítása
		Ellenőrzés	

38 – 39’	Visual Studio Code kényelmi funkciója:
	Több azonos sor egyszerre történő beírása
	Ctrl + *Alt* + *Le kurzormozgató*
	`szam1 = int(input("Kérem, ...."))`
	`szam2 = int(input("Kérem, ...."))`
		Fejlesztői környezet előnye	Bemutatás

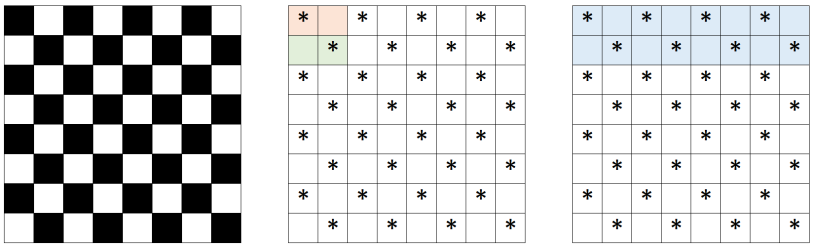
39 – 40	2. feladat - Python_07_het_foglalkozas.pdf 27. feladata Generáljon le egy véletlen egész számot az [1; 12[intervallumban, és írja ki a képernyőre a számot, valamint a szám 3-as maradékát! Ezek után kérdezze meg a felhasználót, hogy szeretne-e újabb számot! Ha igen, ismételje meg az előzőt, ha nem, lépjen ki a programból! A felhasználó választását (i/n) egy változóban tárolja el!	Feladat kitűzése	Python_07_het_foglalko-zas.pdf

IDŐKERET	A FOGLALKOZÁS MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK

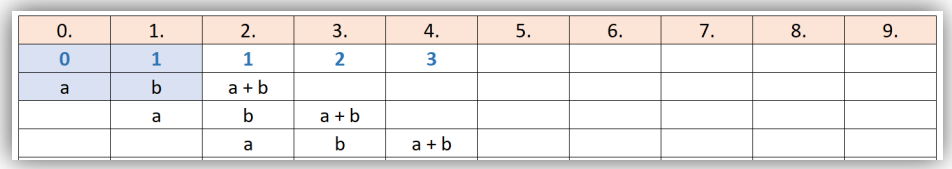
40 - 43	Véletlenszámok generálása
	`import random` – a random modul importálása
	
	`random.randrange(alsó határ, felső határ)`
	visszaad egy véletlen egész számot az [alsó határ; felső határ [balról zárt intervallumban
	pl.: random.randrange(3, 10) → a lehetséges értékek: 3, 4, 5, 6, 7, 8, 9,
	
	`random.randrange(alsó határ, felső határ, növekmény)`
	a 3. paraméter egy növekményt megadó egész szám, alpértelmezetten 1
	pl.: random.randrange(3, 10, 2) → a lehetséges értékek: 3, 5, 7, 9
	
	`random.randint(alsó határ, felső határ)`
	visszaad egy véletlen egész számot az [alsó határ; felső határ] zárt intervallumban
	pl.: random.randint(0,3)-→ a lehetséges értékek: 0, 1, 2, 3
	
	`random.random()`
	véletlenszerű valós számot (float) ad vissza a [0; 1] zárt intervallumban
	
	`random.uniform(alsó határ, felső határ)`
	véletlenszerű valós számot ad vissza az [alsó határ; felső határ] zárt intervallumban
	pl.: random.uniform(20, 60)
	
	A feladat megoldása során melyik függvényre lesz szükségünk?
	Válasz: randrange()
	Hány paraméterrel?
	Válasz: 2
		Visszacsatolás
		
		Megerősítés
		
		Rögzítés
		
		Alkalmazás
			Projektor
			
			Python_07_het_foglalko-zas.pdf
			
			Frontális munka
			
			Megbeszélés

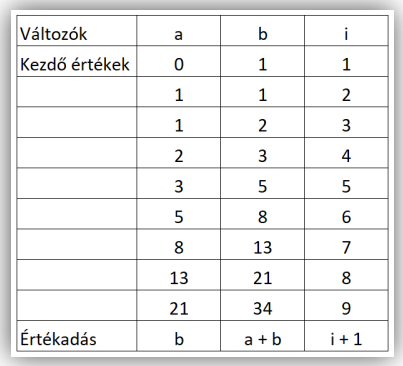
IDŐKERET	A FOGLALKOZÁS MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK

43 – 50’	Kód:
	`print("\n\n27. feladat\n")`
	`print("Véletlenszámot generálunk az [1; 12[ intervallumban.")`
	`while True:`
	`  veletlenEgesz = random.randrange(1, 12)`
	`  print("A véletlen egész szám", veletlenEgesz, ", a hármas maradéka:", veletlenEgesz % 3)`
	`  ch = input("Szeretne egy újabb véletlenszámot? ")`
	`  while not(ch == 'i' or ch == 'n'):`
	`  ch = input("Nem megfelelő karaktert ütött le! Adjon meg 'i' vagy 'n' karaktert!")`
	`  if ch == "n":`
	`  break`
	`print("VÉGE!")`
	
	ch != 'i' and ch != 'n' De Morgan azonosság
	https://hu.wikipedia.org/wiki/De_Morgan-azonoss%C3%A1gok
	
	Futási eredmény
		Programkód írása
		Szintaktika követése
		Precizitás, pontosság
		Strukturáltság
		
		Alkotó képesség
		
		De-Morgen azonosságok alkalmazása
		
		Önellenőrzés
			Egyéni munka
			
			Ha szükséges:
			Frontális munka
			
			Internet

50 – 54’	3. feladat - Python_07_het_foglalkozas.pdf 2/2. feladata Rajzoljon le egy 8 * 8-as sakktáblát csillagokból és szóközökből! 	Algoritmikus, absztrakt. logikus gondolkodás Esztétikai érzék	Python_07_het_foglalkozas.pdf SEGEDanyagok_Gyakorlat_07.xlsx Sakk munkalap Projektor

<table>
<tr><th>IDŐKERET</th><th>A FOGLALKOZÁS MENETE</th><th>CÉLOK ÉS FELADATOK</th><th>NEVELÉSI-OKTATÁSI STRATÉGIA
MEGJEGYZÉSEK</th></tr>
<tr><td>54 – 60’</td><td>Kód:
<pre>
for i in range(4):
    for j in range(4):
        print("* ", end="")
    print()
    for j in range(4):
        print(" *", end="")
    print()
print()
</pre>
Futási eredmény</td><td>Programkód készítése
Szintaktika követése
Strukturáltság

Önellenőrzés → Hibajavítás</td><td>Egyéni munka
vagy
Frontális munka</td></tr>
<tr><td>60 – 62’</td><td>4. feladat - Python_07_het_foglalkozas.pdf 2/12. feladata
Program, amely kiírja az első N Fibonacci számot. Fibonacci-számok: Az első két elem 0 és 1, a további elemeket az előző kettő összegeként kapjuk.
Képletben: Ha $F_0 = 0$ és $F_1 = 1$ akkor a következő elemeket úgy kapjuk, hogy $F_n = F_{n-1} + F_{n-2}$
Az első néhány Fibonacci-szám: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34</td><td>Feladat kijelölése</td><td>Python_07_het_foglalko-zas.pdf</td></tr>
<tr><td>62 – 65’</td><td>
<table>
<tr><th>0.</th><th>1.</th><th>2.</th><th>3.</th><th>4.</th><th>5.</th><th>6.</th><th>7.</th><th>8.</th><th>9.</th></tr>
<tr><td>0</td><td>1</td><td>1</td><td>2</td><td>3</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>a</td><td>b</td><td>a + b</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td>a</td><td>b</td><td>a + b</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td>a</td><td>b</td><td>a + b</td><td></td><td></td><td></td><td></td><td></td></tr>
</table></td><td>Algoritmikus gondol-kodás</td><td>SEGEDanyagok_Gyakor-lat_07.xlsx
Fibonacci munkalap
Szemléltetés</td></tr>
</table>

<table>
<tr><th>IDŐKERET</th><th>A FOGLALKOZÁS MENETE</th><th>CÉLOK ÉS FELADATOK</th><th>NEVELÉSI-OKTATÁSI STRATÉGIA
MEGJEGYZÉSEK</th></tr>
<tr><td>65 – 67’</td><td>
<table>
<tr><td>Változók</td><td>a</td><td>b</td><td>i</td></tr>
<tr><td>Kezdő értékek</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td></td><td>1</td><td>1</td><td>2</td></tr>
<tr><td></td><td>1</td><td>2</td><td>3</td></tr>
<tr><td></td><td>2</td><td>3</td><td>4</td></tr>
<tr><td></td><td>3</td><td>5</td><td>5</td></tr>
<tr><td></td><td>5</td><td>8</td><td>6</td></tr>
<tr><td></td><td>8</td><td>13</td><td>7</td></tr>
<tr><td></td><td>13</td><td>21</td><td>8</td></tr>
<tr><td></td><td>21</td><td>34</td><td>9</td></tr>
<tr><td>Értékadás</td><td>b</td><td>a + b</td><td>i + 1</td></tr>
</table></td><td>Algoritmus felisme-rése, megértése
Algoritmus alkotása</td><td>Projektor
Megbeszélés
SEGEDanyagok_Gyakor-lat_07.xlsx
Fibonacci munkalap</td></tr>
<tr><td>67 – 69’</td><td>Számok cseréje:
<pre>a, b = 0, 1
print(a, b)
a, b = b, a
print(a, b)</pre></td><td>Visszacsatolás
Rögzítés</td><td>Frontális munka</td></tr>
<tr><td>69 – 72’</td><td>Kód írása:
<pre>n = int(input("Hány Fibonacci számot szeretne kiíratni? "))
a, b, i = 0, 1, 0
while i < n :
    print(a, end=" ")
    a, b, i = b, a + b, i + 1</pre></td><td>Alkalmazás</td><td>Egyéni munka</td></tr>
<tr><td>72 –74’</td><td>Néhány érdekesség:
math modul factorial függvénye</td><td>Új ismeret közlése</td><td></td></tr>
</table>

IDŐKERET	A FOGLALKOZÁS MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
	`n = int(input("Kérem, adjon meg egy egész számot a faktoriálishoz! "))` `print(math.factorial(n))`		Ha marad idő, akkor bemutatom, ha nem, akkor önálló feldolgozásra adom ki az alábbiakat. Bemutatás

74 – 76’	Törtek kezelése
	fractions modul Fraction osztálya
	`print("Kérem, adjon meg két egész számot, a tört számlálóját és nevezőjét!")`
	`a = int(input())`
	`b = int(input())`
	`tort = fractions.Fraction(a, b)`
	`print(tort)`
		Új ismeret közlése	Bemutatás

76 – 79’	Tanultak összefoglalása	Összefoglalás Rendszerezés	Frontális munka
79 – 80’	Értékelés, házi feladat kiadása	Értékelés A házi feladat kijelölése	Beszélgetés, elbeszélés Frontális munka

IDŐKERET	A FOGLALKOZÁS MENETE	CÉLOK ÉS FELADATOK	NEVELÉSI-OKTATÁSI STRATÉGIA MEGJEGYZÉSEK
	Tatalék feladat - Python_07_het_foglalkozas.pdf 28. feladata Készítsen programot, amely bekér egész számokat mindaddig, amíg nem adja meg a 0-t! A program határozza meg és írja ki a beadott egész számok közül a legkisebbet és a legnagyobbat! (A 0-t ne számítsa bele a beadott számokba, ez csak a bevitel végét jelzi, ez az ún. végjel!) A számok beolvasását a 0 végjelig ciklus (while True) segítségével valósítsa meg!	Feladat kitűzése	Python_07_het_foglalko-zas.pdf

	Feladat megoldása:
	`min = sys.maxsize`
	`# A sys.maxint konstans eltávolításra került, mivel az egész számok értékére`
	`# már nincs korlátozás. A sys.maxsize azonban használható egész számként.`
	`max = -sys.maxsize`
	`while True:`
	`  egeszSzam = int(input("Kérem, adjon meg egy egész számot! "))`
	`  if egeszSzam < min:`
	`  min = egeszSzam`
	`  elif egeszSzam > max:`
	`  max = egeszSzam`
	`  elif egeszSzam == 0:`
	`  break`
		Új ismeret közlése
		Algoritmikus gondol-kodás
		Programkód készítése
		Szintaktika követése
			Frontális munka
			Egyéni munka

Feladatok

0. Alapfeladat:

   Olvasson be egész számokat mindaddig, amíg 0-t nem adnak meg! Ha 0-t adnak, írja ki, hogy hány darab szám került beolvasásra a 0 kivételével, és vége a feldolgozásnak!

   Más megfogalmazásban: Olvasson be egész számokat 0 végjelig, majd írja ki a beolvasott értékek számát! A végjel nem eleme a sorozatnak!

1. Írjon programot, ami bekér a felhasználótól egy pozitív számot! Ha a felhasználó nem a feltételnek megfelelő számot ad meg, ismételje meg a bekérést! Ha a szám megfelelő, írja ki a képernyőre a szám 5-ös maradékát!
2. Generáljon le egy véletlen egész számot az [1; 12[intervallumban, és írja ki a képernyőre a számot, valamint a szám 3-as maradékát! Ezek után kérdezze meg a felhasználót, hogy szeretne-e újabb számot! Ha igen, ismételje meg az előzőt, ha nem, lépjen ki a programból! A felhasználó választását (i/n) egy változóban tárolja el!
3. Rajzoljon le egy 8 * 8-as sakktáblát csillagokból és szóközökből!
4. Program, amely kiírja az első N Fibonacci számot. Fibonacci-számok: Az első két elem 0 és 1, a további elemeket az előző kettő összegeként kapjuk. Képletben: Ha $F_0 = 0$ és $F_1 = 1$ akkor a következő elemeket úgy kapjuk, hogy $F_n = F_{n-1} + F_{n-2}$

   Az első néhány Fibonacci-szám: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34

5. Tartalék feladat:

   Készítsen programot, amely bekér egész számokat mindaddig, amíg nem adja meg a 0-t! A program határozza meg és írja ki a beadott egész számok közül a legkisebbet és a legnagyobbat! (A 0-t ne számítsa bele a beadott számokba, ez csak a bevitel végét jelzi, ez az ún. végjel!) A számok beolvasását a 0 végjelig ciklus (while True) segítségével valósítsa meg.

Megjegyzés

Eddig digitális munkarendben folyt az oktatás, egyetlen kontakt órám volt az osztállyal, ha a haladás nem a terv szerint megy, akkor a kimaradt feladatot - a megoldás csatolásával - önálló feldolgozásra adom ki az osztálynak.

Amennyiben látom, hogy a 0. alapfeladat nem okoz nehézséget, akkor az algoritmizálás folyamatát lerövidítem. Pl.: elhagyható a PowerPoint megoldása, elegendő csak a kész algoritmus kivetítése.