



TANULÁSI ÚTMUTATÓ

Objektumorientált programozás, tanulási útmutató informatikus felsőoktatási szakképzési szakosoknak

Készítette: Békési József Dávid Balázs Hajdu László

SZTE JGYPK

Informatika Alkalmazásai Tanszék

Lektorálta: Tóth Attila

Jelen tananyag a Szegedi Tudományegyetemen készült az Európai Unió támogatásával. Projekt azonosító: EFOP-3.4.3-16-2016-00014.

Alprojekt azonosító: AP2 – Komplex képzés- és szolgáltatásfejlesztés

Altéma azonosító: AP2_JGYPK5 Magyar és idegen nyelvű képzések oktatási innovációja az MTMI területen és tanártovábbképzés



TARTALOMJEGYZÉK

BEVEZETÉS	3
1. A TANTÁRGY LEÍRÁSA	4
1.1. TANTÁRGY TANÍTÁSÁNAK CÉLJA	4
1.2. A TANTÁRGY TANULÁSI EREDMÉNYEI.....	4
1.3. A TANTÁRGY TANULMÁNYI ELŐFELTÉTELE(I), PÁRHUZAMOSSÁGA(I)	5
1.4. A TANTÁRGY TANANYAGTARTALMA (FŐBB TÉMAKÖRÖK) – TEMATIKUS EGYSÉGEK.....	5
1.5. A TANANYAGTARTALOM FELDOLGOZÁSÁNAK IDŐTERVE	1
1.6. AZ ADOTT TUDÁSELEMEK ÁTADÁSÁT ILLETVE ELSAJÁTÍTÁSÁT SEGÍTŐ MUNKAFORMÁK	1
1.7. AZ ADOTT TUDÁSELEMEK ÁTADÁSÁT ILLETVE ELSAJÁTÍTÁSÁT SEGÍTŐ MUNKAMÓDSZEREK	1
1.8. ÉVKÖZI TANULMÁNYI KÖVETELMÉNYEK.....	1
1.9. A MEGSZERZETT TUDÁS ÉS KOMPETENCIÁK ELLENŐRZÉSE ÉS ÉRTÉKELÉSE	1
1.10. A TANTÁRGY TANÍTÁSÁNAK-TANULÁSÁNAK TÁRGYI FELTÉTELEI	2
1.11. A TANTÁRGY MINŐSÉGFEJLESZTÉSI MÓDSZEREI ÉS FEJLESZTÉSI POLITIKÁJA	2
2. A TANTÁRGY TEMATIKUS EGYSÉGEI	1
2.1. A NETBEANS KÖRNYEZET HASZNÁLATA.....	1
2.1.1. TANULÁSI FELADATOK.....	1
2.1.2. ÖNELLENŐRZŐ FELADATOK	1
2.1.3. MEGOLDÓKULCS AZ ÖNELLENŐRZŐ FELADATOKHOZ.....	1
2.2. AZ OBJEKTUMORIENTÁLTSAĞ ALAPJAI	2
2.2.1. TANULÁSI FELADATOK.....	2
2.2.2. ÖNELLENŐRZŐ FELADATOK	1
2.2.3. MEGOLDÓKULCS AZ ÖNELLENŐRZŐ FELADATOKHOZ.....	1
2.3. A FÜGGVÉNYKITERJESZTÉS.....	2
2.3.1. TANULÁSI FELADATOK.....	2
2.3.2. ÖNELLENŐRZŐ FELADATOK	1
2.3.3. MEGOLDÓKULCS AZ ÖNELLENŐRZŐ FELADATOKHOZ.....	1
2.4. LÁTHATÓSÁG, CSOMAGOK.....	2
2.4.1. TANULÁSI FELADATOK.....	2
2.4.2. ÖNELLENŐRZŐ FELADATOK	1
2.4.3. MEGOLDÓKULCS AZ ÖNELLENŐRZŐ FELADATOKHOZ.....	1
2.5. STATIKUS METÓDUSOK, STATIKUS ADATTAGOK, FINAL KULCSSZÓ	2
2.5.1. TANULÁSI FELADATOK.....	2
2.5.2. ÖNELLENŐRZŐ FELADATOK	1
2.5.3. MEGOLDÓKULCS AZ ÖNELLENŐRZŐ FELADATOKHOZ.....	1
2.6. BEÁGYAZOTT OSZTÁLYOK	2
2.6.1. TANULÁSI FELADATOK.....	2



2.6.2.	ÖNELLENŐRZŐ FELADATOK	1
2.6.3.	MEGOLDÓKULCS AZ ÖNELLENŐRZŐ FELADATOKHOZ.....	1
2.7.	ÖRÖKLŐDÉS.....	2
2.7.1.	TANULÁSI FELADATOK	2
2.7.2.	INFORMÁCIÓS LAP.....	1
2.7.3.	ÖNELLENŐRZŐ FELADATOK	1
2.7.4.	MEGOLDÓKULCS AZ ÖNELLENŐRZŐ FELADATOKHOZ.....	1
2.8.	ABSZTRAKT OSZTÁLYOK	2
2.8.1.	TANULÁSI FELADATOK.....	2
2.8.2.	ÖNELLENŐRZŐ FELADATOK	1
2.8.3.	MEGOLDÓKULCS AZ ÖNELLENŐRZŐ FELADATOKHOZ.....	1
2.9.	INTERFÉSZEK	2
2.9.1.	TANULÁSI FELADATOK.....	2
2.9.2.	ÖNELLENŐRZŐ FELADATOK	1
2.9.3.	MEGOLDÓKULCS AZ ÖNELLENŐRZŐ FELADATOKHOZ.....	1
2.10.	KIVÉTELKEZELÉS	2
2.10.1.	TANULÁSI FELADATOK.....	2
2.10.2.	ÖNELLENŐRZŐ FELADATOK	1
2.10.3.	MEGOLDÓKULCS AZ ÖNELLENŐRZŐ FELADATOKHOZ.....	1
2.11.	STREAMEK, ADAT IO.....	2
2.11.1.	TANULÁSI FELADATOK.....	2
2.11.2.	ÖNELLENŐRZŐ FELADATOK	1
2.11.3.	MEGOLDÓKULCS AZ ÖNELLENŐRZŐ FELADATOKHOZ.....	1
2.12.	KOLLEKCIÓK	2
2.12.1.	TANULÁSI FELADATOK.....	2
2.12.2.	ÖNELLENŐRZŐ FELADATOK	1
2.12.3.	MEGOLDÓKULCS AZ ÖNELLENŐRZŐ FELADATOKHOZ.....	1





Bevezetés

Az Objektumorientált Programozás kurzus célja, hogy megismertesse a hallgatókat napjaink egyik legelterjedtebb programozási paradigmájával. Ennek központi elemei az úgy nevezett objektumok, amik lényegében adatszerkezetek és metódusok kombinációjával alkotott, újrahasznosítható programozási elemek. Egy objektumorientált programot ilyen objektumok összessége épít fel, annak működése ezek kommunikációján és kölcsönhatásain alapszik. Mivel egy informatikus képzésben részt vevő hallgató számára általános tudásanyagnak kell tekinteni a kurzus anyagát, így az a tanszék összes képzésének tanrendjében szerepel.

A kurzus oktatása során párhuzamosan ismerkednek meg a hallgatók a szükséges elméleti és gyakorlati tudásanyaggal, az elhangzott elméleti témakörök minden esetben gyakorlati feladatokkal vannak prezentálva. A félév során mindkét ismeretanyag számonkérésre kerül: az elméleti témakörökből a legfontosabb fogalmak, míg gyakorlati esetben előre kiadott feladatokat, valamint a gyakorlati anyaghoz kapcsolódó, de ismeretlen programkiírásokat is képesnek kell lennie megvalósítani a hallgatónak.

A kurzushoz szükséges, főleg gyakorlati tananyag digitálisan megosztásra kerül a résztvevőkkel Coospace-en keresztül is, ami azonban nem elegendő a kurzus teljesítéséhez. Az órai feladatok követése és a megoldásukban való részvétel, valamint a kiadott gyakorló feladatok megoldása és a rendszeres gyakorlás kritikus részei a tantárgy teljesítésének. Gyakorlati rutin nélkül nehezen teljesíthető a számonkérés.

A kurzus ismeretanyaga átfedésben van Angster Erzsébet: Objektumorientált tervezés és programozás c. könyvével, aminek tanulmányozása ajánlott a kurzus hallgatóinak számára.



1. A tantárgy leírása

A tantárgy megnevezése: Objektumorientált Programozás		A tantárgy kódja: SZAKKR4
A tantárgy kredit-értéke:	2	
A tantárgy teljesítési formája:	gyakorlati jegy	
A tantárgy típusa:	szeminárium	
A tantárgy jellege:		
A tantárgy oktatásának ajánlott féléve:	2.	
A tantárgy meghirdetésének gyakorisága:	évente	
A tantárgy óraszám: - kontakt: - egyéni:	2 kontakt óra 2 óra egyéni munka	
A tantárgy heti óraszám:	2	
A tantárgy oktatásának nyelve:	magyar	
A tantárgyat meghirdető tanszék/ szakcsoport:	Informatika Alkalmazásai Tanszék	
A tantárgy felelőse és elérhetősége:	Békési József	
A tantárgyelem oktatója és elérhetősége:	bekesi@jgypk.u-szeged.hu	

1.1. Tantárgy tanításának célja

A kurzus folyamán a hallgató elsajátítja az objektumorientált tervezés és programozás fontosabb elemeit és témaköreit. A feladatok gyakorlati megvalósítása a Java programozási nyelv segítségével történik, a NetBeans fejlesztői környezetben belül. A hallgató képes lesz a megoldandó feladathoz szükséges összetett objektumok megtervezésére, létrehozására, és egy ezeken alapuló program elkészítésére.

1.2. A tantárgy tanulási eredményei

Azoknak az **előírt szakmai kompetenciáknak, kompetencia-elemeknek** (tudás, képesség stb., KKK 7. pont) a felsorolása, amelyek kialakításához a tantárgy jellemzően, érdemben hozzájárul:

A tantárggyal kialakítandó konkrét tanulási eredmények:

Tudás	Képesség	Attitűd	Autonómia/felelősség
Ismeri a legfontosabb vezérlőszerkezeteket, így a szelekciókat és az iterációkat.	Szelekciós és iterációs utasításokat alkalmaz az elkészített kódban.	Törekszik a szelekciók és iterációk pontos megvalósítására.	Képes önállóan szelekciós és iterációs utasítások használatára.
Ismeri a metódusokat, a metódusok felépítését, azok lokális változóinak	Rekurzív metódusokat alkalmaz.	Igyekszik a rekurzív metódusok pontos használatára.	Betartja a rekurzív metódushívás szabályait.

szerepét, valamint a rekurziót.			
Legyen tisztában az osztályok valamint az objektumok használatával, készítésével (referenciátípus, objektum létrehozása, deklaráció)	Egyszerűbb osztályokat és objektumokat készít, objektumokat deklarál.	Törekszik az objektumok és az osztályok precíz használatára.	A példatár segítségével önállóan ellenőrzi a munkáját.
Ismeri az osztály fogalmát, felépítését, deklarációit, valamint az osztálytag és példánytag fogalmát.	Osztály- és példánytagokat tartalmazó osztályokat hoz létre.	Nyitott az objektumok különböző módszerekkel történő létrehozására, az objektumok különböző célú alkalmazására.	Önállóan megtalálja és javítja a program hibáit.
Ismeri az azonosító, hivatkozási kör, takarás, alapértelmezés szerinti kezdeti érték, this objektumreferencia, konstruktor valamint az inicializálók fogalmát.	Fejlesztőrendszerben objektumokat inicializál, konstruktorokat létrehoz, objektumokat a referenciájukkal kezel.	Elkötelezett az objektumok és referenciáik pontos használata iránt.	Önállóan végez elemi objektumokkal kapcsolatos műveleteket, paraméterátadást és ezeket önállóan, pontosan kódolja.
Tisztában van az öröklődés, a típuskonverziók, az utódosztály adatai és kapcsolatai, a metódus felülírása, a dinamikus és statikus kötés, valamint a konstruktorláncolás technikájával.	Megérti és használja az osztályok öröklötését, a típuskonverzió technikáját. Felismeri a dinamikus és statikus kötetést. Alkalmazza a konstruktorok láncolását.	Törekszik az objektumok adatainak pontos meghatározására. Szem előtt tartja, hogy az öröklődés milyen esetekben alkalmazható.	Önállóan meghatározza az egyes objektumok konstruktorait.

1.3. A tantárgy tanulmányi előfeltétele(i), párhuzamossága(i)

Előfeltétel(ek): Programozás alapjai

Párhuzamosság(ok): nincs

1.4. A tantárgy tananyagtartalma (főbb témakörök) – tematikus egységek

- NetBeans környezet használata
- Objektumorientáltság alapjai
- Függvénykiterjesztés.
- Láthatóság, csomagok.
- Statikus metódusok, statikus adattagok, final kulcsszó
- Beágyazott osztályok
- Öröklődés, osztályhierarchia, polimorfizmus, super kulcsszó használata.
- Absztrakt osztályok.
- Interfészek.
- Kivételkezelés.



EFOP-3.4.3-16-2016-00014

SZÉCHENYI 2020

- Streamek, Adat IO.
- Kollektciók.

Szegedi Tudományegyetem
Cím: 6720 Szeged, Dugonics tér 13.
www.u-szeged.hu
www.szechenyi2020.hu

SZÉCHENYI 2020



Európai Unió
Európai Szociális
Alap



BEFECTETÉS A JÖVŐBE



EFOP-3.4.3-16-2016-00014

SZÉCHENYI

2020

1.5. A tananyagtartalom feldolgozásának időterve

Kontaktóra			Egyéni óra	
Hét	Óra	Tartalom	Óra	Tartalom
1	2	NetBeans környezet használata	2	A NetBeans fejlesztő környezet telepítése, projekt létrehozása, alapvető programozási elemek használata a környezetben belül, automatikus kiegészítés
2	2	Objektumorientáltság alapjai	2	Osztályok, adattagok, metódusok, konstruktorok, this kulcsszó használata
3	2	Függvénykiterjesztés.	2	Overloading használata konstruktorok és metódusok esetén
4	2	Láthatóság, csomagok.	2	Különböző láthatósági szintek megismerése, osztályok csomagokba rendezése
5	2	Statikus metódusok, statikus adattagok, final kulcsszó	2	Statikus osztálymetódusok és osztálymezők használata, a final kulcsszó használata, final és static kombinálása
6	2	Beágyazott osztályok	2	Beágyazott osztályok készítése
7	2	Öröklődés, osztályhierarchia, polimorfizmus, super kulcsszó használata.	2	Szülő és gyermekosztályok létrehozása, adattagok és metódusok öröklődése, metódus overriding
8	2	Absztrakt osztályok.	2	Az abstract kulcsszó használata, abstract osztályok és metódusok készítése, abstract metódusok implementálása leszármazott osztályban
9	2	Interfészek.	2	Interfészesk létrehozása, interfész implementálása osztályokban, interfész metódusainak megvalósítása
10	2	Kivételkezelés.	2	Fontosabb kivétel típusok vizsgálata, kivételek kezelése: try-catch blokk, illetve kivétel specifikáció segítségével, saját kivétel típusok deklarálása, kivételek dobása
11	2	Streamek, Adat IO.	2	Adatfolyamok fontosabb jellemzői, input-output FileReader és FileWriter segítségével



EFOP-3.4.3-16-2016-00014

SZÉCHENYI 2020

12	2	Kollekciók.	2	Collection interface, annak leszármazott interface-ei, valamint a megvalósító osztályok megismerése, List és Set interface-ek használata
----	---	-------------	---	--



1.6. Az adott tudáselemek átadását illetve elsajátítását segítő munkaformák

A hallgató kétféle módon találkozik a tárgyhoz kapcsolódó tananyaggal: kontaktóra keretében, illetve otthoni gyakorlás során. Kontaktóra folyamán az oktatóval közösen történik a tananyag feldolgozása, míg az otthoni gyakorlást a hallgató önállóan végzi.

1.7. Az adott tudáselemek átadását illetve elsajátítását segítő munkamódszerek

Az oktató minden óra elején ismerteti az adott témakört, illetve a hozzá tartozó fontosabb fogalmakat. Ezt követően egy példán keresztül szemlélteti ezen tudás gyakorlatba való átültetését, amelynek megvalósítását lépésről-lépésre, a hallgatókkal közösen végzi. A feladat megvalósítása során a hallgatóknak lehetőségük van kérdések feltételére a tananyag részeit illetően.

Minden tanórához kapcsolódóan gyakorló feladatok kerülnek kiosztásra, melyek megvalósítását a hallgatók otthon, önálló munkaként végzik. Az esetlegesen felmerülő kérdésekkel kapcsolatban itt is lehetőség van az oktatóval való egyeztetésre, konzultációs óra keretében.

1.8. Évközi tanulmányi követelmények

A félév során a kontaktórákról két alkalom esetében engedélyezett a hiányzás, kihagyott óra pótlására vagy kiváltására nincs lehetőség. A zárthelyi dolgozatok időpontjában mindenképpen kötelező a részvétel.

A kontaktórához kapcsolódóan az óra során otthoni gyakorló feladatok (óránként legalább 1) kerülnek kiosztásra a hallgatóknak, melyek elkészítése opcionális. A feladatok megoldása kontaktórákon nem kerül elő, de konzultációs óra keretében bármikor megbeszélhetők az oktatóval a félév során.

1.9. A megszerzett tudás és kompetenciák ellenőrzése és értékelése

A félév során a kontaktórákon 2 db zárthelyi dolgozat kerül megírásra a 7. és a 13. héten a gyakorlat anyagából, melyek egyenként 20-20 pontosak. A gyakorlat teljesítésének feltétele mindkét zárthelyi dolgozatnál egyenként minimum 10 pont elérése. Amennyiben valamelyik dolgozatnál nincs meg a minimális pontszám, úgy a félév végén javító dolgozat írására van lehetőség a teljes féléves anyagból. Csak egy megírt dolgozat javítható ilyen módon. Ha valamelyik dolgozat pontszáma ezek után sem éri el a minimális 10 pontot, úgy a gyakorlat értékelése elégtelen.

Nem megírt dolgozat pótlására nincs lehetőség.

A félév végi értékelés a két dolgozat pontszáma alapján történik:

40-36: jeles (5)

35-31: jó (4)

30-26: közepes (3)

25-20: elégséges (2)

19- 0: elégtelen (1)

1.10. A tantárgy tanításának-tanulásának tárgyi feltételei

A tantárgy oktatása számítógépes tanteremben történik, ahol minden munkaállomásra telepíteni kell a Java JDK aktuális verzióját, valamint az aktuálisan legfrissebb verziójú NetBeans fejlesztői környezetet.

Az otthoni gyakorló feladatok megoldásához szintén számítógépre van szüksége a hallgatónak, melyekre ugyanúgy szükséges a telepített Java JDK és NetBeans. Ezek az internetről ingyenesen letölthetők és telepíthetők bárki számára.

1.11. A tantárgy minőségfejlesztési módszerei és fejlesztési politikája

- Az oktató mindig naprakész a kurzus tananyagához kapcsolódó tudását és technológiákat illetően.
- Az oktató folyamatosan egyeztet az előfeltétel kurzus, valamint a ráépülő kurzusok oktatóival a tananyaggal kapcsolatban. Az esetlegesen felmerülő igények alapján módosítja a kurzus tematikáját és tartalmát.
- Az oktató állandó kapcsolatban van a szakmai gyakorlati helyek képviselőivel, és az igényeik, valamint a hallgatókra vonatkozó visszajelzéseik alapján módosítja a kurzus tematikáját és tartalmát.
- Az oktató meghallgatja a hallgatók visszajelzéseit, amik alapján esetleges módosításokat eszközölhet a kurzus tempóján, beosztásán, valamint az általa alkalmazott tanítási módszereken.

2. A tantárgy tematikus egységei

2.1. A NetBeans környezet használata

2.1.1. Tanulási feladatok

Tartalom:**A NetBeans telepítése és használata**

- NetBeans letöltése és telepítése
- Projektek létrehozása és kezelése
- Példaprogramok készítése és futtatása
- Automatikus kiegészítés használata

A tematikus egység tanulási eredményei:

A hallgató legyen képes

- fejlesztői környezetet telepíteni és beállítani,
- programozási feladatait projektekbe rendezni
- a fejlesztés során magabiztosan használja a környezet nyújtotta kényelmi lehetőségeket.

Szükséges eszközök, anyagok:

- A hallgatók felkészüléséhez felhasználható szakirodalom (jegyzet, tankönyv, egyéb források és segédanyagok:

Kötelező:

Ajánlott: Coospace-n elérhető tananyag

- Egyebek:



Tanóra (Kontaktóra) (1 kontaktóra = 45 perc)			Egyéni hallgatói munkaóra óra (1 egyéni hallgatói munkatóra = 60 perc)		
Ren- delke- zésre álló idő- tartam	Tanulási tevékenység	Különleges instrukciók	Ren- delke- zésre álló idő- tartam	Tanulási tevékenység	Különleges instrukciók
3 óra	Töltse le és telepítse a NetBeans fejlesztői környezetet! Ismerje meg a projektek létrehozásának módszerét! Készítsen egyszerű programot a környezet kipróbálására! Ismerje meg az automatikus kiegészítés funkciót!	Az automatikus kiegészítés funkció alapbeállításként előhívható bármikor a ctrl+space billentyűkombinációval..	3 óra	Ültesse át korábbi kurzusokról származó programjait NetBeans környezetbe! Írja újra a programkódot, és a megvalósítás során próbálja minél többet használni az automatikus kiegészítést!	Projekt létrehozásakor figyeljen arra, hogy automatikusan generálja a main metódust tartalmazó osztályt is. Az osztálynév és a projekt neve eltérhetnek! Ügyeljen rá, hogy kódja szintaktikailag helyes legyen!

2.1.2. Önellenőrző feladatok

Ismertesse a projekt létrehozásának folyamatát!

Adjon új osztályt a projekthez!

Készítsen programot, melybe egy szelekciós és egy ismétléses vezérlési szerkezet illeszt be automatikus kiegészítés segítségével!

Fordítsa és futtassa a fenti programot!

2.1.3. Megoldókulcs az önellenőrző feladatokhoz

Képes a projektet elnevezni, a main osztályt létrehozni, annak külön nevet adni. (30 pont)

Képes új osztály létrehozására. (10 pont)

Képes tetszőleges elágazást generálni automatikus kiegészítéssel. (20 pont)

Képes tetszőleges ciklust generálni automatikus kiegészítéssel. (20 pont)

Képes fordítani és futtatni a programot. (20 pont)

Az önellenőrzés értékelése:

Maximálisan elérhető pontszám: 100 pont. A sikeres teljesítéshez legalább 51%-os (51 pont) teljesítés szükséges.

50 pontig: elégtelen (1)

62 pontig: elégséges (2)

75 pontig: közepes (3)

88 pontig: jó (4)

100 pontig: jeles (5)

Hallgatói teljesítményértékelő lap

4Ellenőrizze, hogy elvégezte-e a tematikus egység valamennyi feladatát! Minden kérdésnél tegyen egy X-et a leginkább megfelelő rovatba, tehát értékelje saját maga a feladat végrehajtását. Ha a felsoroltak közül valamelyik feladat teljesítése nem történt meg vagy lehetetlen volt a teljesítése, tegyen X-et a "Nem" oszlopba.

		Nem	Igen
1.	Képes projektet létrehozni, illetve kezelni.		
2.	Képes fordítani és futtatni a kész projektet.		
3.	Képes használni az automatikus kiegészítés funkciót.		

2.2. Az objektumorientáltság alapjai

2.2.1. Tanulási feladatok

Tartalom

Objektumok tervezése

- Mezők
- Metódusok
- Konstruktor
- this kulcsszó
- getter, setter, toString

Objektumok használata

- Példányosítás
- Példánymetódus hívása

A tematikus egység tanulási eredményei:

A hallgató képes legyen:

- Objektumot tervezni egyszerű feladat modellezésére
- Konstruktor segítségével paraméteren keresztül mezőket beállítani
- Metódust készíteni a feladat megoldására
- a szükséges getter, setter, toString metódusok létrehozására
- objektumok példányosítására és használatára

Szükséges eszközök, anyagok:

- A hallgatók felkészüléséhez felhasználható szakirodalom (jegyzet, tankönyv, egyéb források és segédanyagok):

Kötelező:

Ajánlott:

- Coospace-n elérhető tananyag
- Angster Erzsébet: Objektumorientált tervezés és programozás
- Egyebek:



Tanóra (Kontaktóra) (1 kontaktóra = 45 perc)			Egyéni hallgatói munkaóra óra (1 egyéni hallgatói munkatóra = 60 perc)		
Ren- delke- zésre álló idő- tartam	Tanulási tevékenység	Különleges instrukciók	Ren- delke- zésre álló idő- tartam	Tanulási tevékenység	Különleges instrukciók
2 óra	Ismerje az objektumok fogalmát, alapötletét, felépítésük főbb elemeit (mező, konstruktor, metódus)! Legyen képes objektumot tervezni, használja a speciális metódusokat (getter, setter, toString)! Tudjon programon belül objektumot példányosítani, és annak metódusait használni.	Példánymetódus hívásához először példányosított objektum szükséges!	2 óra	Válasszon egy tetszőleges dolgot való életből, amit objektumként modellez, és egy programon belül használ (példányosít, metódusokat hív, stb.)!	Azonos nevű mező és paraméter esetén figyeljen a this kulcsszó használatára! Ügyeljen rá, hogy kódja szintaktikailag helyes legyen!

2.2.2. Önellenőrző feladatok

Sorolja fel az objektumokat leíró osztályok főbb elemeit!

Sorolja fel azokat az alapvető metódusokat, amik az objektumról információt szolgáltatnak, vagy annak tulajdonságait módosítják.

Mondja ki, hogy mely kulcsszó segítségével lehet objektumot példányosítani.

2.2.3. Megoldókulcs az önellenőrző feladatokhoz

Megnevezte: mezők, konstruktorok, metódusok. (30 pont)

Megnevezte: getter, setter, toString. (60 pont)

Megnevezte: new (10 pont)

Az önellenőrzés értékelése:

Maximálisan elérhető pontszám: 100 pont. A sikeres teljesítéshez legalább 51%-os (51 pont) teljesítés szükséges.

50 pontig: elégtelen (1)

62 pontig: elégséges (2)

75 pontig: közepes (3)

88 pontig: jó (4)

100 pontig: jeles (5)

Hallgatói teljesítményértékelő lap

Ellenőrizze, hogy elvégezte-e a tematikus egység valamennyi feladatát! Minden kérdésnél tegyen egy X-et a leginkább megfelelő rovatba, tehát értékelje saját maga a feladat végrehajtását. Ha a felsoroltak közül valamelyik feladat teljesítése nem történt meg vagy lehetetlen volt a teljesítése, tegyen X-et a "Nem" oszlopba.

	Nem	Igen
1. Ismeri az objektumok használatát, képes objektumot tervezni.		
2. Képes objektumot példányosítani.		
3. Képest használni az objektumokat, azok példánymetódusait.		

2.3. A Függvénykiterjesztés.

2.3.1. Tanulási feladatok

Tartalom

Konstruktorok kiterjesztése

- konstruktor overloading
- this kulcsszó konstruktorokra

Metódusok kiterjesztése

- metódus overloading

A tematikus egység tanulási eredményei:

- Érti és alkalmazza a konstruktor overloading jelenségét, képes konstruktorokat láncolni a this kulcsszó segítségével
- Érti és alkalmazza a metódus overloading jelenségét

Szükséges eszközök, anyagok:

- A hallgatók felkészüléséhez felhasználható szakirodalom (jegyzet, tankönyv, egyéb források és segédanyagok:

Kötelező:

Ajánlott:

- Coospace-n elérhető tananyag
- Angster Erzsébet: Objektumorientált tervezés és programozás
- Egyebek:



Tanóra (Kontaktóra) (1 kontaktóra = 45 perc)			Egyéni hallgatói munkaóra óra (1 egyéni hallgatói munkatóra = 60 perc)		
Ren- delke- zésre álló idő- tartam	Tanulási tevékenység	Különleges instrukciók	Ren- delke- zésre álló idő- tartam	Tanulási tevékenység	Különleges instrukciók
2 óra	Az előző héten készített objektumhoz készítsen konstruktort, ami kiterjeszti a meglévő konstruktort! A konstruktorban this segítségével hívja meg a régi konstruktor! Az előző héten készített objektumhoz egy metódusához készítsen metódust, ami kiterjeszti a meglévő metódust! Használja a kiterjesztett konstruktort és metódust programon belül!		2 óra	Az előző héten otthon készített objektumhoz készítsen konstruktort, ami kiterjeszti a meglévő konstruktort! A konstruktorban this segítségével hívja meg a régi konstruktor! Az előző héten otthon készített objektumhoz egy metódusához készítsen metódust, ami kiterjeszti a meglévő metódust! Használja a kiterjesztett konstruktort és metódust programon belül!	Minden esetben ügyeljen a megfelelő paraméter-átadásra! Ügyeljen rá, hogy kódja szintaktikailag helyes legyen!



2.3.2. Önellenőrző feladatok

Készítsen osztályt, amiben szerepel kiterjesztett konstruktor és metódus is!

Példányosítsa ezt mindkét konstruktorral, és használja mindkét metódust.

2.3.3. Megoldókulcs az önellenőrző feladatokhoz

A programkód szintaktikailag helyes (30 pont)

A konstruktor, valamint annak kiterjesztése is megfelelően működik. (20 pont)

A metódus, valamint annak kiterjesztése is megfelelően működik. (20 pont)

A paraméterátadások minden esetben megfelelőek. (30 pont)

Az önellenőrzés értékelése:

Maximálisan elérhető pontszám: 100 pont. A sikeres teljesítéshez legalább 51%-os (51 pont) teljesítés szükséges.

50 pontig: elégtelen (1)

62 pontig: elégséges (2)

75 pontig: közepes (3)

88 pontig: jó (4)

100 pontig: jeles (5)

Hallgatói teljesítményértékelő lap

Ellenőrizze, hogy elvégezte-e a tematikus egység valamennyi feladatát! Minden kérdésnél tegyen egy X-et a leginkább megfelelő rovatba, tehát értékelje saját maga a feladat végrehajtását. Ha a felsoroltak közül valamelyik feladat teljesítése nem történt meg vagy lehetetlen volt a teljesítése, tegyen X-et a "Nem" oszlopba.

	Nem	Igen
1. Ismeri és tudja alkalmazni a konstruktor kiterjesztést.		
2. Ismeri és tudja alkalmazni a metódus kiterjesztést.		
3. Képes konstruktorok hívására a this kulcsszó segítségével.		

2.4. Láthatóság, csomagok

2.4.1. Tanulási feladatok

Tartalom:**A láthatósági szintek ismerete**

- private
- package-private
- protected
- public

Csomagok

- osztályok rendszerezése
- import

A tematikus egység tanulási eredményei:

A hallgató képes legyen:

- Ismeri a láthatósági szinteket
- Képes a helyzetnek megfelelő láthatóság kiválasztására
- Képes osztályait csomagokba rendezni
- Képes más csomagokban lévő osztályok használatára

Szükséges eszközök, anyagok:

- A hallgatók felkészüléséhez felhasználható szakirodalom (jegyzet, tankönyv, egyéb források és segédanyagok:

Kötelező:

Ajánlott:

- Coospace-n elérhető tananyag
- Angster Erzsébet: Objektumorientált tervezés és programozás
- Egyebek:



Tanóra (Kontaktóra) (1 kontaktóra = 45 perc)			Egyéni hallgatói munkaóra óra (1 egyéni hallgatói munkatóra = 60 perc)		
Ren- delke- zésre álló idő- tartam	Tanulási tevékenység	Különleges instrukciók	Ren- delke- zésre álló idő- tartam	Tanulási tevékenység	Különleges instrukciók
1 óra	Ismerje meg a különböző láthatósági szinteket! Példákon keresztül ismerje meg a szintek közötti különbséget!		2 óra	Készítsen programot, ami több egyszerű osztályt tartalmaz, külön csomagokba rendezve! Az egyes osztályok használják adattagként más csomagokba tartozó osztályokat! Mindig használja a megfelelő láthatóságot!.	Az import kulcsszó szükséges, ha más csomag elemeihez szeretnénk hozzáférni. Ügyeljen rá, hogy kódja szintaktikailag helyes legyen!
1 óra	Készítsen programot, ami csomagokba rendszerezi osztályait! Az egyes osztályok érjenek el más csomagban lévő osztályokat is!				



2.4.2. Önellenőrző feladatok

Készítsen programot, ami tartalmazza az összes órán tanult elemet!

Használjon legalább 3 láthatóságot a megvalósítás során!

Rendezze a létrehozott osztályait csomagokba!

2.4.3. Megoldókulcs az önellenőrző feladatokhoz

A programkód szintaktikailag helyes (20 pont)

Legalább három láthatósági szint szerepel a programban. (30 pont)

Legalább két csomagba rendezte osztályait. (40 pont)

Használja az import kulcsszót. (10 pont)

Az önellenőrzés értékelése:

Maximálisan elérhető pontszám: 100 pont. A sikeres teljesítéshez legalább 51%-os (51 pont) teljesítés szükséges.

50 pontig: elégtelen (1)

62 pontig: elégséges (2)

75 pontig: közepes (3)

88 pontig: jó (4)

100 pontig: jeles (5)

Hallgatói teljesítményértékelő lap

Ellenőrizze, hogy elvégezte-e a tematikus egység valamennyi feladatát! Minden kérdésnél tegyen egy X-et a leginkább megfelelő rovatba, tehát értékelje saját maga a feladat végrehajtását. Ha a felsoroltak közül valamelyik feladat teljesítése nem történt meg vagy lehetetlen volt a teljesítése, tegyen X-et a "Nem" oszlopba.

	Nem	Igen
1. Ismeri és használja a láthatósági szinteket.		
2. Ismeri és használja a csomagokat.		



2.5. Statikus metódusok, statikus adattagok, final kulcsszó

2.5.1. Tanulási feladatok

Tartalom:**Static kulcsszó**

- használata mezők esetén
- használata adatagok esetén

Final kulcsszó

- használata mezők esetén
- static és final együtt

A tematikus egység tanulási eredményei:

A hallgató képes legyen:

- Megfelelően alkalmazni a static kulcsszót mezők és metódusok esetén is
- Megfelelően használni mezők esetén a final kulcsszót, illetve a kettő kombinációját

Szükséges eszközök, anyagok:

- A hallgatók felkészüléséhez felhasználható szakirodalom (jegyzet, tankönyv, egyéb források és segédanyagok):

*Kötelező:**Ajánlott:*

- Coospace-n elérhető tananyag
- Angster Erzsébet: Objektumorientált tervezés és programozás
- Egyebek:



Tanóra (Kontaktóra) (1 kontaktóra = 45 perc)			Egyéni hallgatói munkaóra óra (1 egyéni hallgatói munkatóra = 60 perc)		
Ren- delke- zésre álló idő- tartam	Tanulási tevékenység	Különleges instrukciók	Ren- delke- zésre álló idő- tartam	Tanulási tevékenység	Különleges instrukciók
2 óra	Ismerje meg a különbséget osztályváltoz/metódus, valamint példányváltozó/metódus között! Ismerje meg a final kulcsszó működését, valamint a static kulcsszóval együtt történő használatát! Készítsen osztályt, amiben szerepelnek a fenti elemek! Használja őket!		2 óra	Készítsen olyan osztályt, amiben szerepel osztályváltozó és osztálymetódus is! Legyen az osztálynak egy static final mezője! Az osztálymetódus módosítsa az osztályváltozót a final mező értékének megfelelően!	Fontos, hogy az osztályszintű változók és metódusok eléréséhez nem szükséges objektumpéldány! Ügyeljen rá, hogy kódja szintaktikailag helyes legyen!

2.5.2. Önellenőrző feladatok

Készítsen osztályt, amiben szerepel osztályszintű metódus és adattag is!

Elkészült osztálya tartalmazzon egy static final mezőt!

A programban futás közben használja a fenti három elemet!

2.5.3. Megoldókulcs az önellenőrző feladatokhoz

A programkód szintaktikailag helyes. (20 pont)

Helyesen használja az osztályszintű adattagot. (20 pont)

Helyesen használja az osztályszintű metódust. (20 pont)

Csak egyszer ad értéket a static final mezőnek (20 pont)

Nem használ objektumpéldányt. (20 pont)

Az önellenőrzés értékelése:

Maximálisan elérhető pontszám: 100 pont. A sikeres teljesítéshez legalább 51%-os (51 pont) teljesítés szükséges.

50 pontig: elégtelen (1)

62 pontig: elégséges (2)

75 pontig: közepes (3)

88 pontig: jó (4)

100 pontig: jeles (5)

Hallgatói teljesítményértékelő lap

Ellenőrizze, hogy elvégezte-e a tematikus egység valamennyi feladatát! Minden kérdésnél tegyen egy X-et a leginkább megfelelő rovatba, tehát értékelje saját maga a feladat végrehajtását. Ha a felsoroltak közül valamelyik feladat teljesítése nem történt meg vagy lehetetlen volt a teljesítése, tegyen X-et a "Nem" oszlopba.

		Nem	Igen
1.	Érti a static kulcsszó tulajdonságait, használatát.		
2.	Érti a final kulcsszó tulajdonságait, használatát.		
3.	Képes a két kulcsszó kombinálására.		

2.6. Beágyazott osztályok

2.6.1. Tanulási feladatok

Tartalom:**Beágyazott osztályok**

- Osztályok létrehozása más osztályokon belül
- Példányosítás, elérhetőség vizsgálata

A tematikus egység tanulási eredményei:

A hallgató képes legyen:

- Beágyazott osztályok megvalósítására és használatára

Szükséges eszközök, anyagok:

- A hallgatók felkészüléséhez felhasználható szakirodalom (jegyzet, tankönyv, egyéb források és segédanyagok:

Kötelező:

Ajánlott:

- Coospace-n elérhető tananyag
- Angster Erzsébet: Objektumorientált tervezés és programozás
- Egyebek:



Tanóra (Kontaktóra) (1 kontaktóra = 45 perc)			Egyéni hallgatói munkaóra óra (1 egyéni hallgatói munkatóra = 60 perc)		
Ren- delke- zésre álló idő- tartam	Tanulási tevékenység	Különleges instrukciók	Ren- delke- zésre álló idő- tartam	Tanulási tevékenység	Különleges instrukciók
2 óra	Ismerje meg a beágyazott osztályok fogalmát, használatukat! Készítsen beágyazott osztályt egy másik osztályon belülre!		2 óra	Egy korábban elkészített osztályához készítsen beágyazott osztályt! Vizsgálja meg, milyen adatokhoz férnek hozzá ennek az osztálynak az elemei! Példányosítson beágyazott osztályt!	Fontos, hogy a beágyazott osztály csak akkor használható objektumpéldányként, ha már létezik a beágyazó objektumból is példány! Ügyeljen rá, hogy kódja szintaktikailag helyes legyen!



2.6.2. Önellenőrző feladatok

Készítsen olyan osztályt, ami rendelkezik beágyazott osztállyal!

Használja ezt a beágyazott osztályt, hogy a beágyazó mezőiben tárolt adatokat olvassa és módosítsa.

2.6.3. Megoldókulcs az önellenőrző feladatokhoz

A kód szintaktikailag helyes. (20 pont)

A program tartalmaz beágyazott osztályt. (20 pont)

A beágyazott osztály példányosításra kerül. (20 pont)

A beágyazott osztály olvassa a beágyazó mezőit. (20 pont)

A beágyazott osztály írja a beágyazó mezőit. (20 pont)

Az önellenőrzés értékelése:

Maximálisan elérhető pontszám: 100 pont. A sikeres teljesítéshez legalább 51%-os (51 pont) teljesítés szükséges.

50 pontig: elégtelen (1)

62 pontig: elégséges (2)

75 pontig: közepes (3)

88 pontig: jó (4)

100 pontig: jeles (5)

Hallgatói teljesítményértékelő lap

Ellenőrizze, hogy elvégezte-e a tematikus egység valamennyi feladatát! Minden kérdésnél tegyen egy X-et a leginkább megfelelő rovatba, tehát értékelje saját maga a feladat végrehajtását. Ha a felsoroltak közül valamelyik feladat teljesítése nem történt meg vagy lehetetlen volt a teljesítése, tegyen X-et a "Nem" oszlopba.

		Nem	Igen
1.	Érti a beágyazott osztályok szerepét, tulajdonságait.		
2.	Képes beágyazott osztályok használatára.		



2.7. Öröklődés

2.7.1. Tanulási feladatok

Tartalom:**Öröklődés**

- osztályhierarchia kialakítása
- polimorfizmus
- super kulcsszó használata

A tematikus egység tanulási eredményei:

A hallgató képes legyen:

- Szülő és gyermekosztályok létrehozására
- Metódusok felülírására

Szükséges eszközök, anyagok:

- A hallgatók felkészüléséhez felhasználható szakirodalom (jegyzet, tankönyv, egyéb források és segédanyagok:

Kötelező:

Ajánlott:

- Coospace-n elérhető tananyag
- Angster Erzsébet: Objektumorientált tervezés és programozás
- Egyebek:



Tanóra (Kontaktóra) (1 kontaktóra = 45 perc)			Egyéni hallgatói munkaóra óra (1 egyéni hallgatói munkatóra = 60 perc)		
Ren- delke- zésre álló idő- tartam	Tanulási tevékenység	Különleges instrukciók	Ren- delke- zésre álló idő- tartam	Tanulási tevékenység	Különleges instrukciók
2 óra	Ismerje meg az öröklődés fogalmát, annak fontosságát! Készítsen leszármazott osztályt, használja a super kulcsszót példányosításhoz! Figyelje meg a polimorfizmus jelenségét különböző leszármazott osztályok között! Használjon overridingot metódusok felülírására!		2 óra	Tetszőlegesen választott példára valósítsa meg az öröklődés jelenségét több osztály segítségével! A leszármazott osztályok legalább egy konstruktorában használja a super kulcsszót! Írjon felül metódusokat mindegyik leszármazott osztályban!	Figyeljen, hogy a super kulcsszó konstruktor hívása esetén csak első utasításként szerepelhet! Ügyeljen rá, hogy kódja szintaktikailag helyes legyen!

2.7.2. Információs lap**2.7.3. Önellenőrző feladatok**

Készítsen osztályt, melyből több másik osztályt származtasson le!

A leszármazott osztályok konstruktorában használja fel az őosztályok konstruktorait!

A leszármazott osztályok minden esetben rendelkezzenek új adattagokkal és metódusokkal!

Leszármazott osztályból legalább egyszer hívja meg az őosztály egy metódusát.

Használjon legalább két esetben metódus overridingot!

2.7.4. Megoldókulcs az önellenőrző feladatokhoz

A kód szintaktikailag helyes. (20 pont)

A super kulcsszó használva van a leszármazott osztályok konstruktoraiban. (20 pont)

A leszármazott osztályok tartalmazznak új adattagokat és metódusokat. (20 pont)

Legalább két esetben megvalósításra kerül metódus overriding. (40 pont)

Az önellenőrzés értékelése:

Maximálisan elérhető pontszám: 100 pont. A sikeres teljesítéshez legalább 51%-os (51 pont) teljesítés szükséges.

50 pontig: elégtelen (1)

62 pontig: elégséges (2)

75 pontig: közepes (3)

88 pontig: jó (4)

100 pontig: jeles (5)

Hallgatói teljesítményértékelő lap

Ellenőrizze, hogy elvégezte-e a tematikus egység valamennyi feladatát! Minden kérdésnél tegyen egy X-et a leginkább megfelelő rovatba, tehát értékelje saját maga a feladat végrehajtását. Ha a felsoroltak közül valamelyik feladat teljesítése nem történt meg vagy lehetetlen volt a teljesítése, tegyen X-et a "Nem" oszlopba.

	Nem	Igen
1. Érti az öröklődés koncepcióját, képes alkalmazni azt.		
2. Érti a super kulcsszó működését, és használja azt.		
3. Képes metódusok felülírására.		
4. Érti a polimorfizmus koncepcióját.		

2.8. Absztrakt osztályok

2.8.1. Tanulási feladatok

Tartalom:**Absztrakt osztályok**

- Az abstract kulcsszó használata
- Absztrakt osztályok és metódusok készítése
- Absztrakt metódusok implementálása leszármazott osztályban

A tematikus egység tanulási eredményei:

A hallgató képes legyen:

- Absztrakt osztályok tervezésére és létrehozására
- Absztrakt osztályból történő leszármaztatásra
- Absztrakt metódusok definiálására és implementálására

Szükséges eszközök, anyagok:

- A hallgatók felkészüléséhez felhasználható szakirodalom (jegyzet, tankönyv, egyéb források és segédanyagok:

Kötelező:

Ajánlott:

- Coospace-n elérhető tananyag
- Angster Erzsébet: Objektumorientált tervezés és programozás
- Egyebek:



Tanóra (Kontaktóra) (1 kontaktóra = 45 perc)			Egyéni hallgatói munkaóra óra (1 egyéni hallgatói munkatóra = 60 perc)		
Ren- delke- zésre álló idő- tartam	Tanulási tevékenység	Különleges instrukciók	Ren- delke- zésre álló idő- tartam	Tanulási tevékenység	Különleges instrukciók
2 óra	Ismerje meg az absztrakt osztályok koncepcióját! Értse a különbséget rendes osztály és absztrakt osztály között! Legyen képes absztrakt osztályokat létrehozni, és ősosztályként használni őket! Legyen képes absztrakt metódusok implementálására!		2 óra	Az előző héten elkészített feladatát alakítsa át úgy, hogy az tartalmazzon absztrakt osztályt! Definiáljon legalább egy absztrakt metódust, amit egy leszármazott osztályban valósítson meg!	Figyeljen oda, hogy absztrakt osztályból nem lehet objektumot példányosítani! Ügyeljen rá, hogy kódja szintaktikailag helyes legyen!

2.8.2. Önellenőrző feladatok

Készítsen absztrakt osztályt, melyben szerepel legalább egy absztrakt metódus!

Származtasson le több osztályt az absztrakt osztályból, minden esetben megvalósítva az esetleges absztrakt metódusokat!

Példányosítson minden lehetséges osztályhoz legalább egy objektumot!

2.8.3. Megoldókulcs az önellenőrző feladatokhoz

A kód szintaktikailag helyes. (20 pont)

Létezik legalább két leszármazott osztály. (30 pont)

Minden leszármazott osztályban megvalósításra került az őosztály absztrakt metódusa (40 pont)

Csak nem absztrakt osztályból példányosít objektumot (10 pont)

Az önellenőrzés értékelése:

Maximálisan elérhető pontszám: 100 pont. A sikeres teljesítéshez legalább 51%-os (51 pont) teljesítés szükséges.

50 pontig: elégtelen (1)

62 pontig: elégséges (2)

75 pontig: közepes (3)

88 pontig: jó (4)

100 pontig: jeles (5)

Hallgatói teljesítményértékelő lap

Ellenőrizze, hogy elvégezte-e a tematikus egység valamennyi feladatát! Minden kérdésnél tegyen egy X-et a leginkább megfelelő rovatba, tehát értékelje saját maga a feladat végrehajtását. Ha a felsoroltak közül valamelyik feladat teljesítése nem történt meg vagy lehetetlen volt a teljesítése, tegyen X-et a "Nem" oszlopba.

		Nem	Igen
1.	Érti az absztrakt osztályok koncepcióját.		
2.	Képes absztrakt osztályok definiálására.		
3.	Képes absztrakt metódusokat definiálni és megvalósítani.		

2.9. Interfészek

2.9.1. Tanulási feladatok

Tartalom:**Interfészek**

- Interfészesk létrehozása
- Interfész implementálása osztályokban
- Interfész metódusainak megvalósítása
- default és static

A tematikus egység tanulási eredményei:

A hallgató képes legyen:

- interfészek definiálására
- interfészek megvalósítására osztályokban
- a default és static kulcsszavak használatára interfészek esetén

Szükséges eszközök, anyagok:

- A hallgatók felkészüléséhez felhasználható szakirodalom (jegyzet, tankönyv, egyéb források és segédanyagok:

Kötelező:

Ajánlott:

- Coospace-n elérhető tananyag
- Angster Erzsébet: Objektorientált tervezés és programozás
- Egyebek:



Tanóra (Kontaktóra) (1 kontaktóra = 45 perc)			Egyéni hallgatói munkaóra óra (1 egyéni hallgatói munkatúra = 60 perc)		
Ren- delke- zésre álló idő- tartam	Tanulási tevékenység	Különleges instrukciók	Ren- delke- zésre álló idő- tartam	Tanulási tevékenység	Különleges instrukciók
2 óra	Ismerje meg az interfészek fogalmát, és az absztrakt osztályoktól való különbségeiket! Valósítsa meg interfészeket osztályok segítségével! Ismerje meg a default és static kulcsszavak használatát interfészek esetén! Legyen képes interfészek kiterjesztésére!		2 óra	Az előző heti gyakorlófeladatot bővítse tovább interfészek bevezetésével! Legalább egy interfészt terjesszen ki! Legalább egy interfészben szerepeljen default metódus!	Ügyeljen rá, hogy kódja szintaktikailag helyes legyen!

2.9.2. Önellenőrző feladatok

Készítsen programot, amiben legalább egy absztrakt osztály, és legalább egy interfész szerepel!

Készítsen olyan osztályt, ami kizárólag az absztrakt osztályból származik le

Készítsen olyan osztályt, ami kizárólag az interfészt valósítja meg

Készítsen olyan osztályt, ami egyidőben származik le az absztrakt osztályból, valamint valósítja meg az interfészt!

2.9.3. Megoldókulcs az önellenőrző feladatokhoz

A kód szintaktikailag helyes. (20 pont)

Elkészítette az első osztályt. (20 pont)

Elkészítette a második osztályt. (30 pont)

Elkészítette a harmadik osztályt. (30 pont)

Az önellenőrzés értékelése:

Maximálisan elérhető pontszám: 100 pont. A sikeres teljesítéshez legalább 51%-os (51 pont) teljesítés szükséges.

50 pontig: elégtelen (1)

62 pontig: elégséges (2)

75 pontig: közepes (3)

88 pontig: jó (4)

100 pontig: jeles (5)

Hallgatói teljesítményértékelő lap

Ellenőrizze, hogy elvégezte-e a tematikus egység valamennyi feladatát! Minden kérdésnél tegyen egy X-et a leginkább megfelelő rovatba, tehát értékelje saját maga a feladat végrehajtását. Ha a felsoroltak közül valamelyik feladat teljesítése nem történt meg vagy lehetetlen volt a teljesítése, tegyen X-et a "Nem" oszlopba.

		Nem	Igen
1.	Érti az interfészek koncepcióját, valamint tudja őket alkalmazni.		
2.	Érti a különbséget absztrakt osztály és interfész között.		
3.	Képes absztrakt osztályt és interfészt egyidőben alkalmazni.		
4.	Ismeri a default és static kulcsszavak használatát interfészek esetén.		

2.10. Kivételkezelés

2.10.1. Tanulási feladatok

Tartalom:**Kivételkezelés**

- Fontosabb kivételtípusok megismerése,
- Kivételek kezelése: try-catch blokk, kivételspecifikáció
- finally kulcsszó
- Saját kivételtípusok deklarálása
- Kivételek dobása

A tematikus egység tanulási eredményei:

A hallgató képes legyen:

- kivételeket azonosítani és kezelni
- saját kivételtípust készíteni
- szükség esetén kivételeket dobni

Szükséges eszközök, anyagok:

- A hallgatók felkészüléséhez felhasználható szakirodalom (jegyzet, tankönyv, egyéb források és segédanyagok:

Kötelező:

Ajánlott:

- Coospace-n elérhető tananyag
- Angster Erzsébet: Objektumorientált tervezés és programozás
- Egyebek:



Tanóra (Kontaktóra) (1 kontaktóra = 45 perc)			Egyéni hallgatói munkaóra óra (1 egyéni hallgatói munkatóra = 60 perc)		
Ren- delke- zésre álló idő- tartam	Tanulási tevékenység	Különleges instrukciók	Ren- delke- zésre álló idő- tartam	Tanulási tevékenység	Különleges instrukciók
2 óra	Ismerje meg a kivételek különböző típusait! Legyen tisztában a kivételek dobódásának folyamatával! Készítsen try-catch blokkot kivétel kezelésére! Kezeljen kivételt specifikáció segítségével! Használja a finally blokkot! Készítsen és dobjon saját kivételt!		2 óra	Vizsgálja meg az eddig készített gyakorló feladatait, és azonosítsa azokat a kódrészeket, ahol dobódhatnak kivételek! Minden ilyen esetben egészítse ki az adott kódon úgy, hogy az esetlegesen dobódó kivételek le legyenek kezelve!	Minden lekezelendő kivételtípushoz legyen külön catch blokk! Ügyeljen rá, hogy kódja szintaktikailag helyes legyen!

2.10.2. Önellenőrző feladatok

Készítsen programot, ahol bemenettől függően előfordulhatnak az alábbi kivételek: IndexOutOfBoundsException, NumberFormatException, ArithmeticException!

Minden fent előforduló kivételt kezeljen le!

Deklaráljon saját kivételt a programhoz!

Külön metódusban dobja a saját kivételt, amit szintén kezeljen le a program!

2.10.3. Megoldókulcs az önellenőrző feladatokhoz

A kód szintaktikailag helyes. (20 pont)

Elő tudja idézni a kért kivételeket. (30 pont)

Le tudja kezelni a kért kivételeket (30 pont)

Képes saját kivétel deklarálására (10 pont)

Képes saját kivétel dobására és lekezelésére (10 pont)

Az önellenőrzés értékelése:

Maximálisan elérhető pontszám: 100 pont. A sikeres teljesítéshez legalább 51%-os (51 pont) teljesítés szükséges.

50 pontig: elégtelen (1)

62 pontig: elégséges (2)

75 pontig: közepes (3)

88 pontig: jó (4)

100 pontig: jeles (5)

Hallgatói teljesítményértékelő lap

Ellenőrizze, hogy elvégezte-e a tematikus egység valamennyi feladatát! Minden kérdésnél tegyen egy X-et a leginkább megfelelő rovatba, tehát értékelje saját maga a feladat végrehajtását. Ha a felsoroltak közül valamelyik feladat teljesítése nem történt meg vagy lehetetlen volt a teljesítése, tegyen X-et a "Nem" oszlopba.

	Nem	Igen
1. Ismeri és érti a kivételek koncepcióját.		
2. Képes kivételeket azonosítani és kezelni.		
3. Képes saját kivételeket készíteni.		
4. Képes kivételek dobására.		

2.11. Streamek, Adat IO

2.11.1. Tanulási feladatok

Tartalom:**Streamek, Adat IO**

- Adatfolyamok különböző típusai
- Karakterfolyamok
- FileReader
- FileWriter
- BufferedReader
- BufferedWriter

A tematikus egység tanulási eredményei:

- A hallgató ismerje a folyamatok koncepcióját, alapvető használatukat.
- A hallgató legyen képes információ formázott beolvasására fájlból.
- A hallgató legyen képes információ formázott fájlba írására.

Szükséges eszközök, anyagok:

- A hallgatók felkészüléséhez felhasználható szakirodalom (jegyzet, tankönyv, egyéb források és segédanyagok:

Kötelező:

Ajánlott:

- Coospace-n elérhető tananyag
- Angster Erzsébet: Objektumorientált tervezés és programozás
- Egyebek:



Tanóra (Kontaktóra) (1 kontaktóra = 45 perc)			Egyéni hallgatói munkaóra óra (1 egyéni hallgatói munkatóra = 60 perc)		
Ren- delke- zésre álló idő- tartam	Tanulási tevékenység	Különleges instrukciók	Ren- delke- zésre álló idő- tartam	Tanulási tevékenység	Különleges instrukciók
2 óra	Ismerje meg az adatfolyamok alapfogalmát valamint működési elvét! Ismerje meg a karakterfolyamokat, mint a folyamatok egy speciális fajtáját! Készítsen fájlból olvasó, illetve fájlba író metódusokat a megfelelő folyamatok használatával!	Fájlból olvasásnál a BufferedReader osztály egyszerűsíti a folyamatot.	2 óra	Vizsgálja meg az eddig készített gyakorló feladatait, és azonosítsa azokat a kódrészeket, ahol külső bemenetet olvasott a program, vagy információt írt ki! Minden ilyen esetben változtassa meg a beolvasás és kiírás folyamatát: a bemenet beolvasása formázottan történjen fájlból, míg a kimenetet szintén formázottan külön állományba írja!	Folyamok használata esetén kimondottan fontos a kivételkezelés! Használat után ne felejtse el üríteni és lezárni a folyamatokat! (flush és close) Ügyeljen rá, hogy kódja szintaktikailag helyes legyen!

2.11.2. Önellenőrző feladatok

Készítsen programot, ami képes egy tetszőleges, formázott szöveges fájl beolvasására és feldolgozására!

A szöveges fájl elérési útját parancssori argumentumból olvassa!

A fájlt az előre meghatározott formázás szerint dolgozza fel!

A feldolgozott adatról kapott információkat kimeneti fájlba írja!

Figyeljen a kivételkezelésre minden esetben!

2.11.3. Megoldókulcs az önellenőrző feladatokhoz

A kód szintaktikailag helyes. (20 pont)

Parancssori bemenet helyes használata. (20 pont)

A FileReader helyes használata (20 pont)

A FileWriter helyes használata (20 pont)

Megfelelő a kivételkezelés (20 pont)

Az önellenőrzés értékelése:

Maximálisan elérhető pontszám: 100 pont. A sikeres teljesítéshez legalább 51%-os (51 pont) teljesítés szükséges.

50 pontig: elégtelen (1)

62 pontig: elégséges (2)

75 pontig: közepes (3)

88 pontig: jó (4)

100 pontig: jeles (5)

Hallgatói teljesítményértékelő lap

Ellenőrizze, hogy elvégezte-e a tematikus egység valamennyi feladatát! Minden kérdésnél tegyen egy X-et a leginkább megfelelő rovatba, tehát értékelje saját maga a feladat végrehajtását. Ha a felsoroltak közül valamelyik feladat teljesítése nem történt meg vagy lehetetlen volt a teljesítése, tegyen X-et a "Nem" oszlopba.

	Nem	Igen
1. Érti az adatfolyamok általános koncepcióját és működési elvét.		
2. Érti a karakterfolyamok működési elvét.		
3. Képes karakterfolyam segítségével fájl beolvasására.		
4. Képes karakterfolyam segítségével fájlba írásra.		
5. Képes adatot formázottan beolvasni és kiírni.		
6. Képes a folyamatokhoz kapcsolódó kivételkezelés elvégzésére		

2.12. Kollekción

2.12.1. Tanulási feladatok

Tartalom:**Kollekción**

- Collection interfész általánosan
- Kiterjesztett interfészek
- A kollekciót megvalósító osztályok általánosan
- List és Set interfészek, valamint a megvalósító osztályok használata

A tematikus egység tanulási eredményei:

- A hallgató ismerje a kollekciók fogalmát és működési elvét
- A hallgató legyen képes kiválasztani a feladathoz megfelelő kollekciót.
- A hallgató legyen képes különböző műveleteket végrehajtani kollekción.

Szükséges eszközök, anyagok:

- A hallgatók felkészüléséhez felhasználható szakirodalom (jegyzet, tankönyv, egyéb források és segédanyagok:

*Kötelező:**Ajánlott:*

- Coospace-n elérhető tananyag
- Angster Erzsébet: Objektumorientált tervezés és programozás
- Egyebek:



Tanóra (Kontaktóra) (1 kontaktóra = 45 perc)			Egyéni hallgatói munkaóra óra (1 egyéni hallgatói munkatóra = 60 perc)		
Ren- delke- zésre álló idő- tartam	Tanulási tevékenység	Különleges instrukciók	Ren- delke- zésre álló idő- tartam	Tanulási tevékenység	Különleges instrukciók
2 óra	Ismerje meg az általános Collection interfészt, és annak metódusait! Ismerje meg a kiterjesztő interfészeket, és azok adatszerkezet-specifikus metódusait! Ismerje meg a megvalósító osztályokat, és a különböző metódusok osztályszerkezet-specifikus működését! Készítsen programot, ahol részletesen elemzi a List és Set interfészek különböző implementációit!		2 óra	Vizsgálja meg az eddig készített gyakorló feladatait, és azonosítsa azokat a kódrészeket, ahol a különböző tárolók (tömbök) helyettesíthetők megfelelő kollekciókkal! Minden ilyen esetben változtassa meg az adattárolás módját úgy, hogy a program a feladatnak legjobban megfelelő kollekciót alkalmazza!	Figyeljen arra, hogy egy adott kollekció különböző implementációi más hatékonysággal végezhetik egy ugyanazt a funkciót! Ügyeljen rá, hogy kódja szintaktikailag helyes legyen!

2.12.2. Önellenőrző feladatok

Készítsen programot, ami formázottan olvas be adatot egy bemeneti fájlból!

A beolvasott adatot típustól függően helyezze el különböző List és Set kollekciókba!

A kollekciókban tárolt adatokon végezzen el szűréseket, töröljön adatot kollekciókból, illetve képezze kollekciók unióját!

A kollekciókban található adatokat rendezetten írja fájlba!

2.12.3. Megoldókulcs az önellenőrző feladatokhoz

A kód szintaktikailag helyes. (20 pont)

A fájl formázott írása és olvasása megfelelően működik. (20 pont)

Képes List kollekciók példányosítására és feltöltésére. (20 pont)

Képes Set kollekciók példányosítására és feltöltésére. (20 pont)

Képes kollekciókban tárolt adat bejárására és feldolgozására (20 pont)

Az önellenőrzés értékelése:

Maximálisan elérhető pontszám: 100 pont. A sikeres teljesítéshez legalább 51%-os (51 pont) teljesítés szükséges.

50 pontig: elégtelen (1)

62 pontig: elégséges (2)

75 pontig: közepes (3)

88 pontig: jó (4)

100 pontig: jeles (5)

Hallgatói teljesítményértékelő lap

Ellenőrizze, hogy elvégezte-e a tematikus egység valamennyi feladatát! Minden kérdésnél tegyen egy X-et a leginkább megfelelő rovatba, tehát értékelje saját maga a feladat végrehajtását. Ha a felsoroltak közül valamelyik feladat teljesítése nem történt meg vagy lehetetlen volt a teljesítése, tegyen X-et a "Nem" oszlopba.

	Nem	Igen
1. Érti a kollekciók általános koncepcióját és használatát.		
2. Ismeri az egyes kollekciók közötti különbségeket.		
3. Képes a feladathoz legjobban illő kollekció kiválasztására.		
4. Képes az adott kollekcióhoz tartozó legjobb implementáció kiválasztására.		
5. Képes kollekciókat adattal feltölteni.		
6. Képes a kollekciókban tárolt adat bejárására és feldolgozására.		