Miskolci Egyetem
Általános Informatikai Intézeti Tanszék

Nappali tagozat
Tavaszi félév

GEIAL313-B
Objektum orientált programozás
mérnök-, programtervező és gazdaságinformatikus alapszak (BSc)

Tantárgy előadója, leckekönyvi jegyzője: Dr. Baksáné Dr. Varga Erika, egyetemi docens
Tantárgy lezárásának módja: aláírás és kollokvium
Mintatanterv szerinti félév: 2
Kredit: 5
Kontakt órák száma / hét: 2 előadás, 2 labor gyakorlat

ÜTEMTERV

Hét	Előadás	Gyakorlat
1.	A Java nyelv története, jellemzői. Java fejlesztő környezetek, JDK. A Java nyelv alapelemei (egyszerű típusok, literálok, operátorok, tömb).	Laborszabályzat ismertetése. Eclipse fejlesztőkörnyezet megismerése. Első Java programok. Egydimenziós tömb használata. Parancssori futtatás.
2.	A Java nyelv utasításai, standard input / output.	A Java nyelv utasításainak gyakorlása. Procedurális programok írása. Ellenőrzött beolvasás. Kétdimenziós tömb, String adatok.
3.	Objektum orientált programozás fogalma, alapelvei.	A Java nyelv utasításainak gyakorlása. Procedurális programok írása. Input ellenőrzése, véletlenszám generálás, a Math osztály metódusai, tömbkezelés az Arrays osztály metódusaival.
4.	Osztály definíció, példányosítás, hivatkozások az osztály tagjaira. Referencia változók. Hozzáférési kategóriák. Konstruktor I. Szemétgyűjtő mechanizmus. Példány élettartam.	**1. programozás számonkérés: alapalgoritmusok megvalósítása Java nyelven** Osztálydefiníció és használat. Default konstruktor.
5.	Függvénytúlterhelés (overloading), paraméter átadás. Osztályszintű tagok és használatuk. This kulcsszó és final módosító. Inicializáló blokk. Csomagok (package), osztályok láthatósága, importálás.	Osztálydefiníció, referenciák. Konstruktor. Dátumkezelés.
6.	Öröklődés, polimorfizmus. Metódus felüldefiniálás (overriding). Referenciák statikus és dinamikus típusai.	Osztályszintű tagok, final tagok. Függvénytúlterhelés, this, csomagok.
7.	Absztrakt metódusok és osztályok. Interfészek.	Öröklődés, metódusok felüldefiniálása. Referenciák típusai.
8.	Tagosztályok, interfész tagok. Tömb és enum típus.	Absztrakt osztály és interfész definiálása, használata.

9.	Kivételkezelés.	Kivételkezelés.
10.	A java.lang csomag osztályai.	Beágyazott osztály. Enum osztály.
11.	Java generikusok, kollekciók. Class fájl.	Tömbrendezés. Dinamikus tömb. Enum osztály.
12.	Osztályok közötti kapcsolatok, osztálytervezési szempontok. Kódolási szabályok.	Felkészülés a féléves programozás beszámolóra.
13.	Fájlkezelés.	**2. programozás számonkérés: objektum orientált programozás Java nyelven**
14.	**Elővizsga**	**Programozás számonkérések pótlása**

Tananyag: www.iit.uni-miskolc.hu → Munkatársak: Baksáné V.E. → Oktatott tárgyak → OOP

Ajánlott irodalom:

1. Oracle Java dokumentáció, https://docs.oracle.com/javase/tutorial/
2. Kövesdán Gábor: Szoftverfejlesztés Java SE platformon, Második kiadás – Frissítve Java 11-es verzióhoz, ISBN: 9786150029337, 2018.
3. Antal Margit: Objektumorientált programozás (SAPIENTIA Erdélyi Magyar Tudományegyetem, Műszaki és Humán Tudományok Kar, Marosvásárhely), 2007.

Aláírás megszerzésének feltételei:

1. Aktív részvétel a gyakorlatok 70%-án.
2. A 2 programozás számonkérés sikeres teljesítése.

Vizsga menete: írásbeli és szóbeli

Objektum orientált programozás
1. programozás számonkérés mintafeladat

A program elkészítésére 30 perc áll rendelkezésére.
Semmilyen segédeszköz nem használható.
A program csak akkor elfogadható, ha a megadott feltételeknek megfelelően lett elkészítve, hiba nélkül fut, és a megadott tesztadatokra a megadott eredményt szolgáltatja.

Készítse el az alábbi kiírásnak megfelelő programot Java programnyelven:

Olvasson be ellenőrzött módon egy mátrixba 5 darab koordináta párt a -25 ... 25 tartományban. Számítsa ki a koordináta párok távolságát a Pitagorasz-tétel alkalmazásával, és írja ki a képernyőre a következő alakban:

```
X=17, Y=-21, Távolság=27,018
```

Betartandó feltételek:

- A koordináták ellenőrzött beolvasására készítsen függvényt, amely csak egész számot fogad el a -25 – 25 tartományban.
- A távolság kiszámítására készítsen függvényt, amely paraméterként megkap egy koordináta párt.

Tehát a **main** metódus mellett további **két függvényt** kell elkészítenie.

Beolvasott adatok:

```
14, -5
20, 10
-8, 22
16, -2
8, 11
```

Kiírt adatok:

```
X=14, Y=-5, Távolság=14,866
X=20, Y=10, Távolság=22,360
X=-8, Y=22, Távolság=23,409
X=16, Y=-2, Távolság=16,124
X=8, Y=11, Távolság=13,601
```

Objektum orientált programozás
2. programozás számonkérés mintafeladat

A feladat megoldására 50 perc áll rendelkezésre.
A megoldás során semmilyen segédeszköz nem használható!
A feladatmegoldás során tartsa be az OOP alapelveket!
Az azonosítók megválasztása tetszőleges, de beszédes legyen.
Az értékelés szempontjai: A program akkor elfogadható, ha hiba nélkül fut és az elvárt eredményt adja, és a kód megfelel a feladatleírásnak.

Definiáljon saját csomagban **Torta** osztályt.
Adattagjai (csak az osztályban érhetők el): szeletek száma (int), íz (String), méret (enum: kicsi, közepes, nagy),
Konstruktora a paraméterként kapott 2 értékkel (méret és íz) inicializálja az adattagokat. A szeletek száma a torta méretétől függ. A nagy torta 20 szeletes, a közepes 16, a kicsi 8.
Metódusai:
- Visszaadja a szeletek számát
- Visszaadja az ízt
- Visszaadja a méretét
- toString: egy sztringbe összefűzve visszaadja a torta adatait
- Absztrakt metódus: a torta árát adja vissza (int) és nincs paramétere

Definiáljon ugyanebben a csomagban **SzulinapiTorta** osztályt, amely a Torta leszármazottja.
Adattagjai (csak az osztályban érhetők el): gyertyák száma (int),
Konstruktorai:
1. A paraméterben megadott 4 értékkel inicializálja az adattagokat
2. A paraméterben megadott 3 értékkel inicializálja az adattagokat (az torta méretét automatikusan nagyra állítja)
Metódusai:
- Az adattag értékét lekérdező getter metódus.
- Definiálja felül az ősosztály absztrakt metódusát:

```
ár = szeletek száma * 450 + gyertyák száma * 50
```

- toString: egy sztringbe összefűzve visszaadja a szülinapi torta minden adatát (az árral együtt)
- Összehasonlító metódus: igazat ad, ha a szülinapi torta ára nagyobb, mint a paraméterként kapott másik szülinapi tortáé.

Definiáljon másik csomagban **TortaTeszt** osztályt.
1. Külön metódusban ellenőrzött módon olvassa be a feladat sorszámát (1 és 6 közötti érték). Itt 3.
2. Hozzon létre egy 4 elemű Szulinapi torta tömböt és töltse fel az alábbi adatokkal.

```
SzeletekSzama=8, Iz=Karamellás, Meret=Kicsi, GyertyakSzama=5, Ar=3850
SzeletekSzama=16, Iz=Meggyes, Meret=Kozepes, GyertyakSzama=13, Ar=7850
SzeletekSzama=16, Iz=Csokis, Meret=Kozepes, GyertyakSzama=15, Ar=7950
SzeletekSzama=20, Iz=Erdeigyumolcs, Meret=Nagy, GyertyakSzama=30, Ar=10500
```

3. Írjon metódust, amelyik visszaadja annak a tortának a sorszámát, amelyiknek a legmagasabb az ára. Írassa ki ennek az adatait a main metódusból.
Megoldás:

```
SzeletekSzama=20, Iz=Erdeigyumolcs, Meret=Nagy, GyertyakSzama=30, Ar=10500
```

4. Írjon metódust, amelyik kiszámítja a szülinapi torták átlagárát.
Megoldás: 7537.5 Ft
5. Rendezze a tömböt a gyertyák száma szerint és listázza a rendezett tömb adatait. A tömb elemeinek kiírására írjon külön metódust.

Objektum orientált programozás
Írásbeli vizsga minta

Az írásbeli vizsga időtartama: 30 perc
Értékelés: 0-7 pont sikertelen
8-15 pont sikeres

1. Mi nem lehet osztály tagja, ha nem akarjuk megsérteni az OOP alapelveket? 1 pont
   - tagosztály
   - public metódus
   - public adattag
   - private konstruktor

Megoldás: public adattag

2. Adja meg egy végrehajtható Java program belépési pontját tartalmazó kódsort. Mit jelentenek itt a módosítók? 4 pont

Megoldás:
public static void main(String[] args)
public - hozzáférési módosító, a metódus nyilvános, hatására a metódust a JVM is látja (e nélkül az osztály nem lenne futtatható)
static – osztályszintű metódus (akkor is végrehajtható, ha az osztálynak nincsenek példányai)

3. Hogyan valósítható meg többszörös öröklés Java-ban (szintaktika, magyarázat) ? 3 pont

Megoldás:
Egy osztálynak csak egy közvetlen őse lehet, de tetszőleges számú interfészt implementálhat. Az ősosztály adattagjait, metódusait örökli. Az implementált interfészek absztrakt metódusait meg kell valósítania.
public class Car extends Vehicle implements Steerable, Chargeable { … }

4. Hol használjuk és mire a `throw` és a `throws` kulcsszavakat ? 2 pont

Megoldás:
throws – Metódus fejlécében jelzi, hogy a metódus milyen kivételeket dobhat.
public int read() throws IOException {...}
throw – Kivétel megdobására használt utasítás.
throw new myException();

5. Írja le a kivételkezelő utasítás szintaktikáját, működésének szabályait. 5 pont

`` try { utasítások (try blokk) } catch (ExceptionType name) { utasítások } catch (ExceptionType name) { utasítások } //tetszőleges számú catch blokk finally { //finally blokk elmaradhat utasítások } ```	try – a normál működéshez tartozó kód, ami kivételt dobhat catch – a paraméterben megadott kivételt lekezelő kódrész finally – opcionális; a try kódrész után mindig lefut, erőforrások felszabadításához használjuk

Objektum orientált programozás
Szóbeli vizsga tételsor

1. Az objektum orientált programtervezés alapelvei.
2. Java jellemzői, java platform, java program felépítése, fordítás, futtatás menete. Csomagok. Import.
3. Osztálydefiníció, adattag, metódustag definíció. Egységbezárás alapelvének implementálása.
4. Osztálydefiníció, adattag, metódustag definíció. Információ rejtés alapelvének implementálása.
5. Hivatkozás típusú (referencia) változók fogalma, jellemzői. Elemi tipusú és referencia változók összehasonlítása.
6. Műveletek referencia változókkal. Referencia statikus és dinamikus típusa. Referencia konverziók.
7. Példányosítás szintaktikája, menete. Példány élettartama. Szemétgyűjtő. A final módosító.
8. Konstruktorok fogalma, definíciója, használatának szabályai.
9. A this fogalma, szerepe. Osztályszintű adattagok, metódusok.
10. Öröklődés fogalma, használatának előnyei, hátrányai, jellemzői. Öröklődés szintaktikája. Tagok öröklődése.
11. Polimorfizmus megjelenési formái. Függvény/operátor túlterhelés (overloading), metódus felüldefiniálás (overriding).
12. Absztrakt metódusok, osztályok. Interface fogalma szerepe. Interface definíció, implementálás.
13. Speciális szintaktikával definiálható osztályok: tömb, enum. Generikus osztályok.
14. Kivételkezelés.
15. A lang csomag osztályai: Object, Comparable, System, String, StringBuilder, csomagoló osztályok.
16. Osztályok közötti kapcsolatok.