1. Milyen láthatóságot módosító kulcsszavakat ismersz? Ismertesd az egyes láthatóságok szerepét! (3 pont)



2. Mire jó a static kulcsszó? (3 pont)



3. Sorold fel azokat az eseteket, ahol a final kulcsszót használhatjuk! Mit érünk el vele az egyes eseteknél? (3 pont)



4. Mi a különbség az interface és az abstract class között? (5 pont)



5. Írhatunk-e egy osztály számára több konstruktort? Ha igen, miért? Ha nem, miért nem? (3 pont)



6. Mi a különbség a “==” operátorral, illetve a .Equals() függvény segítségével történő összehasonlítás között? (3 pont)



7. Lehet-e absztrakt metódust létrehozni olyan osztályban, amely nem absztraktként lett megjelölve? Ha igen, miért? (3 pont)



8. Adott az alábbi kiragadott kódrészlet egy programból. A Vegrehajt() metódus működését nem fontos ismernünk. Milyen kivételek keletkezésének esetében futnak le az egyes ágakban található kódok? Mi történik, ha a Vegrehajt() nem dob kivételt? (4 pont)

static void myFunc()

{

               try{

                              MyType t = new MyType();  
                              t.Vegrehajt();

               }

               catch(MyException){  
                              System.out.println(“MyException keletkezett!”);

               }

               catch(Exception){

                              System.out.println(“Kivétel keletkezett!”);

}

               finally{

                              System.out.println(“Finally!”);

               }

}



9. Mit ír ki az alábbi program? Válaszod indokold! (5 pont)

public class SwitchCase {

public static void main( String[] args ) {

int ertek = 1;

switch( ertek ) {

case 1: System.out .print( "1" );

case 2:

case 3: System.out .print( "2" );

break;

case 4: System.out .print( "4" );

default: System.out .print( "D" );

}

}

}



10. Írj példát egy osztályra, ami az Abstrc absztrakt osztályból származik, és megvalósítja az Intrf interfészt! (5 pont)



11. Adott az alábbi Nev osztály, ami vezetékneveket és keresztneveket tartalmaz, illetve egy lista amiben példányok vannak az osztályból. A Java 8 stream api segítségével keresd meg és add vissza az első Sándor keresztnevű személy vezetéknevét. Amennyiben nem létezik adj vissza null-t. A feladatot egyetlen függvényhívási lánccal old meg. (6 pont)

public class Nev {

private String vezeteknev;

private String keresztnev;

public Nev(String vezeteknev, String keresztnev) {

this.vezeteknev = vezeteknev;

this.keresztnev = keresztnev;

}

public String getKeresztnev() {

return keresztnev;

}

public String getVezeteknev() {

return vezeteknev;

}

}



12. Mire való a throws kulcsszó egy függvény fejlécében? (4 pont)



13. Írj unit testet az alábbi osztályhoz! A teszt ellenőrizze, hogy a függvény valóban a megfelelő értéket írja ki, tehát két szám szorzatát! (8 pont)

public MyClass

{

               public int szorzat(int a, int b)

               {

                              return a\*b;

}

}



14. Írjunk programot, amely bekér a felhasználótól N db számot, majd kiírja az összegüket, az átlagukat, és eldönti, hogy van-e köztük két azonos szám! (15 pont)



15. Készítsünk egy virtuális könyvtárat, ahol különböző szoftvereket tárolhatunk! Készítsünk osztályhierarchiát az egyes programtípusok számára. A közös ősük egy absztrakt osztály legyen! Például: Software -> UtilitySoftware, VideoGame, SmallApplications. Készítsünk egy ITelepitheto interfészt, amely egyetlen metódust tartalmaz, a Telepit() metódust. A virtuális könyvtárunkba legyen lehetőségünk új szoftver felvitelére, valamint annak törlésére! A felvitelt és a törlést név szerint jelezzük a konzolon, például: “A \*név\* szoftver hozzáadásra/törlésre került!”. A felhasználó választhasson ki egy szoftvert a listából, és ha megvalósítja az ITelepitheto interfészt, akkor telepíthesse azt! Ezt jelezzük a konzolon névvel ellátva: “A \*név\* szoftver települ!” (20 pont)