HATÉKONY JAVA PROGRAMOZÁS

3. forduló

MSCI

A kategória támogatója: MSCI

Ismertető a feladathoz

A 3.forduló feladatait a hosszú hétvége miatt kivételesen szerda (11.02.) éjfélig tudod megoldani!

Érdemes ebben a fordulóban is játszanod, mert a következő forduló kezdetekor, 11.03-án 18 órától kiosztjuk az 1.-2.-3. fordulóban megszerzett badgeket!

A verseny közben az alábbi teljesítményeket díjazzuk:

- fordulógyőztes
- átlagnál jobb időeredmény
- átlag feletti pontszám
- hibátlan forduló

Szeretnénk rá felhívni figyelmedet, hogy az egyszer megkapott badge-eket nem vonjuk vissza, akkor sem, ha esetleg az adott fordulóban a visszajelzések alapján változások vannak.

Jó játékot!

Felhasznált idő: 00:00/10:00 **Elért pontszám:** 0/11

1. feladat 0/2 pont

Az alábbi állítások közül melyek igazak a Java default metódusokkal kapcsolatban?

```
public interface MyInterface {

        default void defaultMethod() {
        // default implementacio
        }
    }
```

Válaszok

- [x] Minden, az interface-t megvalósító osztály rendelkezni fog a metódussal.
  Ez a válasz helyes, de nem jelölted meg.
- [x] Ha kiterjesztjük az interface-t, a metódus felülírható (override).
  Ez a válasz helyes, de nem jelölted meg.

- [ ] Ha kiterjesztjük az interface-t, a metódus elrejthető (shadowing).
- [x] Ha kiterjesztjük az interface-t, a default metódus felülírható egy implementáció nélküli abstract metódussal.
  Ez a válasz helyes, de nem jelölted meg.
- [ ] Default metódus nem dobhat kivételt.
- [ ] A default metódust az osztályhoz rendeljük hozzá amin definiálva van, és nem az osztályt megvalósító objektumpéldányhoz.

Magyarázat

Ha egy Java interface-ben a metódus előtt a **default** kulcsszót használjuk, lehetőségünk lesz megadni a metódus alapértelmezett implementációját. Minden osztály, amely az interface-t megvalósítja, rendelkezni fog ezzel a metódussal.

Ha kiterjeszteni szeretnénk az interface-t, a **default** metódussal kapcsolatban három lehetőségünk van:

- nem említjük meg, ilyenkor a kiterjesztő interface is megörökli a **default** metódust
- újra deklaráljuk, ezzel a metódus **abstract** lesz
- újra definiáljuk, ezzel felülírjuk az eredeti implementációt

A **default** metódus dobhat kivételt, de elrejteni nem lehet.

2. feladat 0/2 pont

Az alábbi állítások közül melyek igazak egy Java osztályt futtató paranccsal kapcsolatban Java 11-ben?

```
java --class-path=/some-path HelloWorld.java 1 2 3
```

Válasz

- ( ) A hívás nem működik, mert az osztályt előbb le kell fordítani.
- (•) A hívás csak megkötésekkel működik, például az indított program csak egy forrásfileból állhat.
  Ez a válasz helyes, de nem jelölted meg.
- ( ) A hívás csak megkötésekkel működik, például a java file már lefordított bytecode-ot kell tartalmazzon.
- ( ) A hívás csak megkötésekkel működik, például a java osztály nem módosíthatja a környezetet biztonsági okokból (nem ír lemezre, nem fordul a hálózathoz).

Magyarázat

-

3. feladat 0/3 pont

Mit ír ki az alábbi kódrészlet, ha meghívjuk a **main** függvényt?

```java
public class VirtualInit {

  private static class B {
    private List<String> ancestorNames = new ArrayList<>();

    public B() {
      init();
    }

    protected void init() {
      ancestorNames.add("foo");
    }

    public void print() {
      System.out.println(ancestorNames);
    }
  }

  private static class A extends B {

    private List<String> childNames = new ArrayList<>();

    public A() {
      init();
    }

    @Override
    protected void init() {
      childNames.add("bar");
    }

    @Override
    public void print() {
      super.print();
      System.out.println(childNames);
    }
  }

  public static void main(String[] args) {
    new B().print();
    new A().print();
  }
}
```

Válasz

- A program nem fordul.
- [foo]
  [bar]
  [bar]
- [foo]

[bar]

- [ ] [foo]

[foo]

[bar]

- [x] [foo]

  Majd a program kivételt dob.
  Ez a válasz helyes, de nem jelölted meg.

Magyarázat

Az "A" osztály felülírja a "B" osztály init metódusát. A "B" konstruktora meghívja a felülírt init metódust, de ilyenkor az "A"-ban még nincs inicializálva a childNames lista.

4. feladat 0/3 pont

Adott az alábbi record típus (Java 16 alatt). Válaszd ki az igaz állításokat!

```
public record Currency(String symbol, String code) {
        public Currency {
            if (symbol == null || code == null) {
                throw new IllegalArgumentException();
            }
        }
    }
```

Válaszok

- [x] A Currency osztályok szálbiztosak.
  Ez a válasz helyes, de nem jelölted meg.
- [ ] A Currency osztályok megvalósítják a singleton tervezési mintát.
- [ ] A Currency osztályok cache-elve vannak (256 peldányig mint az Integer cache esetén).
- [x] A Currency osztálynak van saját, nem az Object-tól örökölt, equals és hashcode metódusa.
  Ez a válasz helyes, de nem jelölted meg.
- [x] A Currency osztálynak van saját, nem az Object-tól örökölt, equals, hashcode és toString metódusa.
  Ez a válasz helyes, de nem jelölted meg.
- [x] Példányosításkor validálja, hogy a két attribútuma ne kaphasson null értéket.
  Ez a válasz helyes, de nem jelölted meg.

Magyarázat

A Java 14-ben bevezetett **Record** osztály esetében csak a mezőket kell felsorolni, a konstruktor, a getter-ek, az equals, a hashcode és a toString metódusok automatikusan generálódnak.

Az osztály immutable, tehát szálbiztos is.

A fenti példában a konstruktor explicit módon van definiálva, és kivételt dob, ha a **symbol** vagy a **code** változóknak null éréket adnánk.

5. feladat 0/1 pont

Mivel egészítsuk ki az alábbi osztályt, hogy két őstől származzon Java 14-ben?

```
class MultiCall ... {

        @Override
        public void printA() {
          System.out.println("new A");
        }

        @Override
        public void printB() {
           System.out.println("new B");
        }
    }

    private static Class A {

       public void printA() {
         System.out.println("A");
       }

    }

    private static Class B {

       public void printB() {
         System.out.println("B");
       }

    }
```

Válasz

- ○ extends A extends B
- ○ extends A, B
- ○ extends both A, B
- ○ implements A default printA, B default printB
- ● Továbbra sincs többes leszármazas, csak interface-ből lehet többet implementálni
  Ez a válasz helyes, de nem jelölted meg.
- ○ record A, B

Magyarázat

-

Megjelenés

Világos