

HATÉKONY JAVA PROGRAMOZÁS

2. forduló

MSCI

A kategória támogatója: MSCI

Ismertető a feladathoz

Útmutató:

- A **radio button-os kérdésekre** egy helyes válasz van.
- Ha lejár a feladatlap ideje, a rendszer **AUTOMATIKUSAN** beküldi azt az addig megjelölt válaszokkal.
- Az **adatbekérős feladatokra NEM jár részpontszám**, csak a feleletválasztósakra.
- **Badge-ke**t a 4.forduló után kapsz majd először.
- Az **adatbekérős kérdéseknél** igyekeztünk minden variációt megadni (kisbetű, nagybetű, szóköz), de ha mégis eltérést tapasztalsz a megoldásokban, kérjük, jelezd felénk!

+1: Azért szólunk, hogy senkit ne a végén érjen meglepetés: a játék nem tipp-mix és csapatkategória sincs! Természetesen akinek nem inge...

Jó versenyzést kívánunk!

Felhasznált idő: 00:00/10:00

Elért pontszám: 0/15

1. feladat 0/2 pont

Mivel tér vissza az alábbi függvény és miért?

```
public int giveMeANumber() {  
    try {  
        return 1;  
    } finally {  
        return 2;  
    }  
}
```

Válasz

- ☐ A kód nem fordul. Nem lehet return a finally blokkban.
- ☐ 1 Az első return érték.

- ☒ 2 Az utolsó return érték.
Ez a válasz helyes, de nem jelölted meg.

- ☐ Kivétel dobódik, mert a callstack már tartalmaz egy visszatérési értéket.

Magyarázat

A második return felülírja az első.

2. feladat 0/2 pont

Mivel egészítsük ki a kódrészletet, hogy az a lehető leggyorsabban lefusson?

```
public List<Integer> generate() {  
    List<Integer> test = ...;  
    for (int i = 0; i < 1_000_000; i++) {  
        test.add(0, i);  
    }  
    return test;  
}
```

Válasz

- ☐ `new ArrayList<>()`
- ☐ `new ArrayList<>(1_000_000)`
- ☐ `new LinkedList<>(1_000_000)`
- ☒ `new LinkedList<>()`

Ez a válasz helyes, de nem jelölted meg.

Magyarázat

Első helyre beszúrás költsége $O(1)$ LinkedList esetén. Olyan konstruktora, ami megadja a méretét, nincs.

3. feladat 0/2 pont

Mivel tér vissza az alábbi függvény?

```
public int giveMeANumber() {  
    try {  
        throw new IllegalStateException("I am not giving a number");  
    } finally {  
        return 2;  
    }  
}
```

Válasz

- ☐ A kód nem fordul. Nem lehet return a finally blokkban
- ☐ IllegalStateException dobódik
- ☒ 2
Ez a válasz helyes, de nem jelölted meg.
- ☐ Új kivétel dobódik, mert egy visszatérési érték nem adható vissza, ha előtte már kivételt dobtunk.

Magyarázat

A kivételt felülírja a return.

4. feladat 0/3 pont

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

```
public class SummCounter {  
  
    static {  
        init();  
    }  
  
    private static int sum;  
  
    public SummCounter() {  
        init();  
    }  
  
    public static int getSum() {  
        init();  
        return sum;  
    }  
  
    private static volatile boolean initialized = false;  
  
    private static synchronized void init() {  
        if(!initialized) {  
            for(int i = 0; i < 5; ++i) {  
                sum += i;  
            }  
            initialized = true;  
        }  
    }  
}
```


Magyarázat

Csak az első két megoldás működik, és a **replaceAll** reguláris kifejezést vár, bár az is működik, jobb, ha a **replace**-t használjuk **sim** **string**-gel.

6. feladat 0/4 pont

Melyek állítások igazak a JIT/HotSpot vs. AOT/GraalVM fordítókkal kapcsolatban?

Válaszok

- ☐ AOT/GraalVM csak Java nyelven írt programokat tud lefordítani.
- ☒ A program indítási idő AOT esetében alacsonyabb, a memóriaigény alacsonyabb, ezért ez előnyt jelenthet cloud környezetben.
Ez a válasz helyes, de nem jelölted meg.
- ☒ Tetszőleges osztályok betöltése reflection-nel csak JIT/HotSpot fordító esetében működhet.
Ez a válasz helyes, de nem jelölted meg.
- ☒ AOT/GraalVM elő tud állítani natívan futtatható programokat is, ilyen esetben nincs szükség előre telepített futtató környezetre a célgépen.
Ez a válasz helyes, de nem jelölted meg.
- ☐ JIT/HotSpot esetében az alkalmazás általában kevesebb időt tölt garbage collection-nel, mert csak a JIT/HotSpot-nál van többszálú GC implementáció.

Magyarázat

GraalVM több programozási nyelvet is támogat mint a Python vagy a JavaScript.

G1 már GraalVM esetében is elérhető (megkötésekkel), ezért az utolsó válasz hibás.

A többi válasz helyes.



[Legfontosabb tudnivalók](#)

[Kapcsolat](#)

[Versenyszabályzat](#)

[Adatvédelem](#)

© 2023 Human Priority Kft.

KÉSZÍTETTE **cone**

Megjelenés

Világos