

# HATÉKONY JAVA PROGRAMOZÁS

3. forduló

MSCI

A kategória támogatója: MSCI

RENDELKEZÉSRE ÁLLÓ IDŐ:

10:00

## Ismertető a feladathoz



### Fontos információk

Ezután a forduló után automatikusan jár a [kitartóknak szóló garantált ajándékunk](#), érdemes kitöltened a feladatlapot! :)

Ha kifutsz az adott feladatlap kitöltésére rendelkezésre álló időből, a felület **automatikusan megpróbálja beküldeni** az addig megadott válaszokat.

A kérdésekre **mindig van helyes válasz**, olyan kérdés viszont nincs, amelyre az összes válasz helyes!

Egyéb információkat a [versenyszabályzatban](#) találsz!

Felhasznált idő: 02:08/10:00

Elért pontszám: 0/5

## 1. feladat 0/1 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

- ☒ A Currency típus immutable tulajdonsággal rendelkezik
- ☐ Az equals és a hashCode implementációja az Object őssztályával egyenértékű
- ☐ Az attribútumokhoz tartozó getter metódusok package private láthatósággal rendelkeznek
- ☐ Az attribútumokhoz tartozó setter metódusok package private láthatósággal rendelkeznek
- ☐ Örökléssel bővíthető
- ☒ Példányosításkor validálja, hogy a két attribútuma ne kaphasson null értéket

### Magyarázat

A válaszok a JEP 395: Records specifikáció alapján egyértelműek.

## 2. feladat 0/1 pont

A var kulcsszó alábbi használatai közül melyik felel meg a nyelv fordítási követelményeinek?

### Válasz

☐

```
var version = "1.0";  
version = 1.1;
```

☐

```
var list = new ArrayList<>();  
list = new LinkedList<>();
```

☐

```
class BestClass {  
    var a = "String";  
}
```

☐

```
var x = null;
```

☐

egyik sem

### Magyarázat

A specifikáció egyértelműen tiltja az összes bemutatott példát: ellentmondó típusok, null és attribútum nem támogatott.

### 3. feladat 0/1 pont

Mi lesz a listában az alábbi kódrészlet lefutása után?

```
List<Integer> list = ThreadLocalRandom.current()  
    .ints(1, 91)  
    .distinct()  
    .sorted()  
    .limit(5)  
    .boxed()  
    .collect(Collectors.toList());
```

#### Válasz

☐

5 db egyedi véletlen szám 1 és 90 között (határokat is beleértve)

☐

5 db egyedi véletlen szám 1 és 91 között (határokat is beleértve)

☐

egész számok 1-től 5-ig sorrendben

☐

egész számok 1-től 90-ig véletlen sorrendben

☐

a kódrészlet valamilyen okból nem futhat le (pl. fordítási hiba, kivétel, végtelen ciklus)

### Magyarázat

A kódrészlet egy végtelen streamet próbál rendezni, ami végtelen futást eredményez.

#### 4. feladat 0/1 pont

Mi a különbség a final és a sealed osztályok között?

##### Válasz

- ☐ a sealed osztályok kódját nem lehet módosítani a verziókezelő rendszerben, a final osztályt nem lehet örökléssel bővíteni
- ☒ mindkettő az öröklést korlátozza, de sealed osztályok esetében meghatározható a leszármazott osztályok listája
- ☐ a sealed osztályokkal való műveletek sokkal hatékonyabbak (mint a final osztályokkal végzettek), mert az attribútumok nem módosíthatók
- ☐ a final kulcsszó túl sokféle szemantikával rendelkezett különböző helyzetekben, ezért bevezették a sealed kulcsszót is, mely ugyanazt jelenti

##### Magyarázat

A Java nyelv specifikáció egyértelműen megadja a helyes választ.

#### 5. feladat 0/1 pont

Mit ír ki az alábbi kódrészlet?

```
Stream
    .of(5.1, 3.3, 4.0, 55f, 2.1)
    .parallel()
    .map(a -> String.valueOf(a))
    .mapToDouble(a -> Double.valueOf(a))
    .mapToInt(a -> Long.valueOf(Math.round(a)).intValue())
    .filter(a -> a % 2 == 0)
    .forEach(System.out::print);
```

##### Válasz

- ☐ 53
- ☐ 4
- ☐ 24
- ☐ 5355
- ☐ 42
- ☐ 534552

- ☐ semmit
- ☒ nem egyértelműen meghatározható

## Magyarázat

A parallel párhuzamosítás miatt a foreach sorrendje nem meghatározható, így a számokat bármilyen sorrendben kiírhatja.

[Legfontosabb tudnivalók](#)

[Kapcsolat](#)

[Versenyszabályzat](#)

[Adatvédelem](#)

© 2022 Human Priority Kft.

KÉSZÍTETTE

Megjelenés

 Világos 