

JAVA PROGRAMOZÁS

1. forduló

cl'ck

A kategória támogatója: Click Clock by BCS

Ismertető a feladatlaphoz

Kérjük, hogy a feladatlapon indítása előtt mindenképp olvasd el az alábbi útmutatót:

MINDEN kérdésre van helyes válasz.

Olyan kérdés NINCS, amire az összes válasz helyes, ha mégis az összes választ bejelölöd, arra a feladatra automatikusan 0 pont jár.

Több válaszlehetőség esetén a helytelen válasz megjelölése mínusz pontot ér.

A radio button-os kérdésekre egy helyes válasz van.

Ha lejár a feladatlapon ideje, a rendszer AUTOMATIKUSAN beküldi azt az addig megjelölt válaszokkal.

Azokat a feladatlaponkat, amelyekhez csatolmány tartozik, javasoljuk NEM mobilon elindítani, erre az érintett feladatlaponk előtt külön felhívjuk a figyelmet.

Az adatbekérős feladatokra NEM jár részpontoszám, csak a feleletválasztósakra.

Helyezéseket a 4. forduló után mutatunk, százalékos formában: adott kategóriában a TOP 20-40-60%-hoz tartozol.

Ha egyszerre több böngészőből, több ablakban vagy több eszközről megnyitod ugyanazt a feladatlaponkat, nem tudjuk vállalni az adatmentéssel kapcsolatban esetlegesen felmerülő anomáliákért a felelősséget!

A ChatGPT használata nem tiltott, de az arra való hivatkozással észrevételt NEM fogadunk el!

A feltűnően rövid idő alatt megoldott feladatlaponk kizárását vonnak maguk után, bármilyen más gyanús esetben fenntartjuk a jogot a fordulón érvénytelenítésére!

Jó versenyzést kívánunk!



A fordulóban feltett kérdések mindegyike Java 17 alapokon értelmezendő. Ha valamelyik kérdés JDK-ban használt eszközre kérdez rá, ott mindig az OpenJDK-t vegyük alapul.

Az említett verzió a következő linken tölthető le: <https://jdk.java.net/java-se-ri/17>

1. feladat 5 pont

Válaszd ki azokat az állításokat, amelyek igaz értékkel térnek vissza?

Válaszok

- ☐ `"OITM" == "OITM"`
- ☐ `"OITM" == new String("OITM")`
- ☐ `"OITM" == new String("OITM").intern()`
- ☐ `"OITM " + "2023" == "OITM 2023"`
- ☐ `(Long)125L == (Long)125L`
- ☐ `(Long)128L == (Long)128L`
- ☐ `Integer.valueOf(-128) == Integer.valueOf(-128)`
- ☐ `Boolean.valueOf("false") == Boolean.valueOf("hamis")`

2. feladat 3 pont

Melyik metódusokat használhatjuk arra, hogy a megadott listából a páros számokat kitöröljük?

```
import java.util.*;
import java.util.stream.*;

class DeleteFromList {
    public static void main(String[] args) {
        List<Integer> list = IntStream.range(0, 10)
            .boxed()
            .collect(Collectors.toList());
        // System.out.println(withFor(new ArrayList<>(list)));
        // System.out.println(withEnhancedFor(new ArrayList<>(list)));
        // System.out.println(withRemoveIf(new ArrayList<>(list)));
        // System.out.println(withForEach(new ArrayList<>(list)));
    }

    // A
    private static List<Integer> withFor(List<Integer> list) {
        for (int i = 0; i < list.size(); i++) {
            if (list.get(i) % 2 == 0) {
                list.remove(i);
            }
        }
        return list;
    }

    // B
    private static List<Integer> withEnhancedFor(List<Integer> list) {
        for (Integer i : list) {
            if (i % 2 == 0) {
                list.remove(i);
            }
        }
        return list;
    }

    // C
    private static List<Integer> withRemoveIf(List<Integer> list) {
        list.removeIf(i -> i % 2 == 0);
        return list;
    }

    // D
    private static List<Integer> withForEach(List<Integer> list) {
        list.forEach(integer -> {
            if (integer % 2 == 0) {
                list.remove(integer);
            }
        });
        return list;
    }
}
```

Válaszok

- ☐ A
- ☐ B
- ☐ C
- ☐ D

3. feladat 3 pont

Mi lesz a kimenete a következő program futásának?

```

import java.util.ArrayList;
import java.util.Collections;
import java.util.List;

class Unmodifiable {

    public static void main(String[] args) {
        List<String> oitmEvek = new ArrayList<>();
        oitmEvek.add("OITM 2023");
        oitmEvek.add("OITM 2022");
        oitmEvek.add("OITM 2021");
        oitmEvek.add("OITM 2020");
        oitmEvek.add("OITM 2019");
        oitmEvek.add("OITM 2018");
        oitmEvek.add("OITM 2017");
        ListHolder listHolder = new ListHolder(oitmEvek);
        List<String> protectedList = listHolder.getList();
        oitmEvek.add("OITM 2016");
        System.out.println(protectedList.contains("OITM 2016"));
    }
}

class ListHolder {
    private final List<String> list;

    public ListHolder(List<String> list) {
        this.list = list;
    }

    public List<String> getList() {
        return Collections.unmodifiableList(list);
    }
}

```

Válasz

- ☐ Fordítási hiba
- ☐ Futási hiba - UnsupportedOperationException
- ☐ true
- ☐ false

4. feladat 3 pont

Mi lesz a kimenete következő kód futásának?

```
1 import java.util.*;
2 import java.util.stream.*;
3
4 class Feladat {
5     public static void main(String[] args) {
6         List<Integer> intList = IntStream.range(0, 10)
7             .boxed()
8             .toList();
9         var list = new ArrayList<Integer>(intList) {
10             List<Integer> doubleValues() {
11                 return stream()
12                     .map(value -> value * 2)
13                     .toList();
14             }
15         };
16         System.out.println(list.doubleValues());
17     }
18 }
19 }
20
```

Válasz

- ☐ Fordítási hiba
- ☐ Futási hiba
- ☐ [0, 2, 4, 6, 8, 10, 12, 14, 16, 18]
- ☐ [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
- ☐ [0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20]
- ☐ [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

5. feladat 3 pont

Mi lesz a kimenete a következő program futásának?

```

@FunctionalInterface
interface PossibleFunctionalInterface {

    String toString();

    String calculateString();

}

class FunctionalInterfaceDemo {
    public static void main(String[] args) {
        doYourThing(() -> "Hello OITM");
    }

    static void doYourThing(
        PossibleFunctionalInterface instance
    ) {
        System.out.println(instance.calculateString());
    }
}
```

Válasz

- ☐ Fordítási hiba - Egy funkcionális interfész csak egy darab absztrakt metódussal rendelkezhet.
- ☐ Futási hiba - Bár a program lefordul, de futási időben a futtató nem tudja eldönteni, hogy a két String-et visszaadó metódus közül melyiket kell lefuttatni.
- ☐ Hello OITM
- ☐ null - Mert a toString van hamarabb definiálva az interfészben, és ebben az esetben azt valósítjuk meg, így a calculateString null-t ad vissza.

6. feladat 2 pont

Mi lesz a következő program lefutásának a végeredménye?

```
1 import java.time.*;
2
3 class DateTimeManipulation {
4     public static void main(String[] args) {
5         var dateTime = LocalDateTime.of(2023, 5, 23, 15, 30);
6         dateTime = dateTime.plusDays(10)
7             .minusHours(5);
8         var tokyoDateTime = dateTime.atZone(ZoneId.of("Asia/Tokyo"));
9         var atUtc = tokyoDateTime.withZoneSameLocal(ZoneId.of("UTC"));
10        System.out.println(atUtc.getHour());
11    }
12
13 }
```

Válasz

Megoldások beküldése