JAVA PROGRAMOZÁS

3. forduló

A kategória támogatója: Click Clock by BCS

Ismertető a feladatlaphoz

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

> Amennyiben olyan kategóriában játszol, ahol van csatolmány, de hibába ütközöl a letöltésnél, ott valószínűleg a vírusirtó korlátoz, annak ideiglenes kikapcsolása megoldhatja a problémát. (Körülbelül minden 3000. letöltésnél fordul ez elő.)

> Helyezéseket a 4. forduló után mutatunk, százalékos formában: adott kategóriában a TOP 20-40-60%-hoz tartozol.
>
> A feltűnően rövid idő alatt megoldott feladatlapok kizárást vonnak maguk után, bármilyen más gyanús esetben fenntartjuk a jogot a forduló érvénytelenítésére!

A Java 17 LTS verzióként érkezett amelyre ha a fejlesztők csak a Java 11 után frissítettek akkor bizonyos újdonságok kimaradhattak, nézzük meg ezeket most közelebbről.

A fordulóban elhangzó 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

Jó versenyzést kívánunk!

1. feladat 3 pont

Mi lesz a következő program futásának a végeredménye?

```java
import java.util.StringJoiner;

interface Customer {

}

class NormalCustomer implements Customer {

  private String id;
  private String username;

  public NormalCustomer(String id, String username) {
    this.id = id;
    this.username = username;
  }

  public String getId() {
    return id;
  }

  public String getUsername() {
    return username;
  }

  @Override
  public String toString() {
    return new StringJoiner(", ",
      NormalCustomer.class.getSimpleName() + "[", "]")
      .add("id=" + getId())
      .add("username=" + getUsername())
      .toString();
  }

}


class PremiumCustomer extends NormalCustomer {

  private long rank;

  PremiumCustomer(String id,
                  String username,
                  long rank) {
    super(id, username);
    this.rank = rank;
  }

  public long getRank() {
    return rank;
  }

  @Override
  public String toString() {
    return new StringJoiner(", ",
      PremiumCustomer.class.getSimpleName() + "[", "]")
```

```
56        .add("id=" + getId())
57        .add("username=" + getUsername())
58        .add("rank=" + rank)
59        .toString();
60    }
61 }
62
63 class SeasonalCustomer extends NormalCustomer {
64    private String season;
65
66    SeasonalCustomer(String id,
67                     String username,
68                     String season) {
69      super(id, username);
70      this.season = season;
71    }
72
73    public String getSeason() {
74      return season;
75    }
76
77    @Override
78    public String toString() {
79      return new StringJoiner(", ",
80        SeasonalCustomer.class.getSimpleName() + "[", "]")
81        .add("id=" + getId())
82        .add("username=" + getUsername())
83        .add("season=" + season)
84        .toString();
85    }
86 }
87
88 class InstanceOfDemo {
89
90    public static void main(String[] args) {
91      var customer = createInstance();
92
93      if (!(customer instanceof PremiumCustomer premiumCustomer)) {
94        System.out.println("Not premium customer");
95      } else {
96        System.out.println(premiumCustomer);
97      }
98
99    }
100
101   @SuppressWarnings("unchecked")
102   private static <T extends Customer> T createInstance() {
103     return (T) new PremiumCustomer("123",
104       "test-premium-customer", 100L);
105   }
106 }
107
```

Válasz

- ○ Fordítási hiba - A `premiumCustomer` nevű lokális változó nem érhető el abban a blokkban

- Futás idejű hiba - `ClassCastException` kivétel fog dobódni.
- PremiumCustomer[id=123, username=test-seasonal-customer, rank=100]
- Not premium customer

2. feladat 3 pont

Adott a következő kódrészlet, mely állítás igaz?

```java
sealed interface Flower permits Narcissus, Rose, Tulip {
}

enum Rose implements Flower {
  RED,
  WHITE,
  YELLOW;
}

record Tulip(String origin) implements Flower {
}

non-sealed class Narcissus implements Flower {
  private String owner;

  Narcissus(String owner) {
    this.owner = owner;
  }

  public String getOwner() {
    return owner;
  }
}

class WhiteNarcissus extends Narcissus {

  WhiteNarcissus(String owner) {
    super(owner);
  }
}
```

Válasz

- A `Rose` típus helytelen.
- A `Tulip` típus helytelen.
- A `Narcissus` típus helytelen.
- A kód fordítási hiba nélkül lefordul.

3. feladat 3 pont

Mely állítások igazak a sealed (lezárt) típusokra?

Válaszok

- [ ] A lezárt típusoknak muszáj rendelkeznie egy `permits` szekcióval.
- [ ] Egy lezárt osztálynak muszáj, hogy legyen legalább 1 alosztálya.
- [ ] Egy lezárt interfésznek muszáj, hogy legyen legalább 1 közvetlen implementációja.
- [ ] Egy lezárt típus lehet enum.
- [ ] Bizonyos körülmények között lehet olyan eset amelyben egy lezárt típusnak nem kell felsorolnia az engedélyezett leszármazottak listáját.
- [ ] Egy lezárt típus lehet rekord.

4. feladat 3 pont

Mit definiálhatunk egy rekord törzsében?

Válaszok

- [ ] Egy példány szintű változót.
- [ ] Egy példány szintű metódust.
- [ ] Egy inicilizáló blokkot.
- [ ] Egy argumentumok nélküli konstuktort.

5. feladat 2 pont

Mi lesz a következő program futásának a végeredménye ?

```java
import java.util.Arrays;

class CompareAndMismatch {
  public static void main(String[] args) {
    int[] a1 = {12, 3, 8, 7, 23};
    int[] a2 = {12, 4, 6, 7, 18};
    int[] a3 = {1, 8, 3, 24, 17};

    System.out.print(Arrays.compare(a1, a2));
    System.out.print(Arrays.compare(a2, a3));

    System.out.print(Arrays.mismatch(a1, a2));
    System.out.print(Arrays.mismatch(a2, a3));
  }
}
```

Válasz

- 1110
- -1110
- 11-10
- 1-112

6. feladat 2 pont

Mi lesz a következő program futásának a végeredménye ?

```java
import java.text.NumberFormat;
import java.text.NumberFormat.Style;
import java.util.Locale;

class FormatTask {

  public static void main(String[] args) {
    var numberFormat = NumberFormat
      .getCompactNumberInstance(Locale.UK, Style.SHORT);
    System.out.print(numberFormat.format(1000.123));
    System.out.print("-");
    System.out.print(numberFormat.format(100_000_000));
  }

}

```

Válasz

- 1000-100000000
- 1.123K-100M
- 1K-100M
- 1K-100000K

7. feladat 2 pont

Mi lesz a következő program futásának a végeredménye ?

```java
import java.time.LocalDateTime;
import java.time.ZoneId;
import java.time.format.DateTimeFormatter;
import java.util.Locale;

class DateTimeFormatterTask {

  public static void main(String[] args) {
    var formatter = DateTimeFormatter.ofPattern("B")
      .localizedBy(Locale.forLanguageTag("HU"));
    var dateTime = LocalDateTime.of(2023, 6, 25, 12, 37, 0);
    dateTime.plusHours(6);
    System.out.println(formatter.format(dateTime
      .atZone(ZoneId.of("UTC"))));
  }

}

```

Válasz

- délelőtt
- délután
- este
- éjjel

Megoldások beküldése