

| | |
|-------------|-----------------|
| Priezvisko: | tlačeným písmom |
| Meno: | |

| | |
|----|--|
| 1b | |
| 2b | |

| | |
|----|--|
| 1 | |
| 2 | |
| 3 | |
| 4 | |
| 5 | |
| 6 | |
| 7 | |
| 8 | |
| 9 | |
| 10 | |

Test trvá 30 minút.

Odpovede na otázky vpíšte do tabuľky. Hodnotia sa len odpovede v tabuľke. Pri otázkach s výberom odpovede je len jedna možnosť správna.

V prípade opravy jasne vyznačte odpoveď, ktorá platí. Každá správna odpoveď má hodnotu vyznačenú v otázke. Nesprávna, nejednoznačná alebo neúplná odpoveď má hodnotu 0 bodov. Postup riešenia sa nehodnotí.

Poškodený list nebude uznaný.

1. (1 b) Pri agregácii v objektovo-orientovanom programovaní

- (a) agregujúci objekt obsahuje agregovaný objekt
- (b) agregujúci objekt je typu agregovaného objektu
- (c) agregovaný objekt obsahuje agregujúci objekt
- (d) agregovaný objekt je typu agregujúceho objektu
- (e) nevzniká žiaden vzťah medzi agregujúcim a agregovaným objektom

2. (1 b) Daná je trieda

```
public class C implements Serializable {
    public C l;
    public C p;
}
```

Uvedte, ktoré z objektov a, b, c, d a e sa zaznamenajú do súboru f.out po vykonaní nasledujúceho kódu (uvedte príslušné písmená; na poradí nezáleží):

```
C a = new C();
C b = new C();
C c = new C();
C d = new C();
C e = new C();
```

```
a.l = c;
a.p = d;
b.l = d;
b.l = c;
c.l = e;
d.l = e;
e.l = a;
```

```
OutputStream out =
    new ObjectOutputStream(
        new FileOutputStream("f.out"));
out.writeObject(e);
out.close();
```

3. (1 b) K triede sa mimo balíka, ktorého je súčasťou, sa bez zavedenia (importu) tohto balíka

- (a) dá prístup rovnaným spôsobom ako v tomto balíku, ale len k jej **public** častiam
- (b) dá prístup za účelom vytvorenia jej inštancií, ale nedajú sa spúšťať jej metódy
- (c) dá prístup cez jej plne kvalifikované meno
- (d) nedá prístup
- (e) dá prístup rovnaným spôsobom ako v tomto balíku

4. (1 b) Rámce na zabezpečenie grafického používateľského rozhrania, akými sú JavaFX a Swing, typicky vyžadujú

- (a) pripojenie potrebnej funkcionality k oknu
- (b) implementáciu sledovania operácií nad prvkami grafického používateľského rozhrania mimo triedy, ktorá implementuje príslušné okno
- (c) implementáciu sledovania operácií nad prvkami grafického používateľského rozhrania v triede, ktorá implementuje príslušné okno
- (d) implementáciu sledovania operácií nad prvkami grafického používateľského rozhrania v triede, ktorá implementuje hlavné okno
- (e) pripojenie potrebnej funkcionality k príslušným prvkom grafického používateľského rozhrania

5. (1 b) Anonymná trieda v Jave sa vytvára

- (a) priamo na mieste tvorby objektu
- (b) pomocou lambda výrazu
- (c) použitím kľúčového slova **class** bez modifikátora prístupu **public**
- (d) pomocou odkazu na metódu
- (e) použitím kľúčového slova **anonymous**

6. (1 b) Ktorý návrhový vzor implementuje tento kód v Jave (každá trieda a rozhranie vo vlastnom súbore)?

```
public interface A {
    void p(B d);
    void q(B d);
    void r();
}

public interface B {
    void t();
}

public class A1 implements A {
    private List<B> bs = new ArrayList<>();
    ...

    public void c() {
        ...
        r();
    }
    public void d() {
        ...
        r();
    }
    public void e() {
        ...
        r();
    }
    public void p(B d) {
        bs.add(d);
    }
    public void r() {
        for (B b : bs) {
            b.t();
        }
    }
    ...
}

public class B1 implements B {
    private A1 a;
    ...

    public B1(A1 a) {
        this.a = a;
    }
    public void t() {
        ...
    }
    ...
}
```

- (a) Composite
- (b) žiaden z uvedených
- (c) Visitor
- (d) Observer
- (e) Strategy

7. (1 b) Vo vykonávanom programe v Jave nič je reprezentovaná

- (a) konštruktorom
- (b) atribútom
- (c) objektom
- (d) procesom
- (e) metódou

8. (1 b) Prístup k atribútom a metódam sa obmedzuje pomocou modifikátorov prístupu, aby sa

- (a) zrýchliło vykonávanie programu
- (b) znížili závislosti medzi triedami
- (c) umožnilo dedenie
- (d) zabezpečilo utajenie údajov a kódu
- (e) ochránili autorské práva

9. (1 b) Čo sa vypíše po spustení nasledujúceho programu v Jave?

```
interface T {
    void m();
}

abstract class C implements T {
    public void m() {
        System.out.print("C");
    }
}

class X extends C {
    public void m() {
        super.m();
        System.out.print("X");
    }
}

class Y extends X {
    public void m() {
        System.out.print("Y");
    }
}

class M {
    public static void exe(T... o) {
        for (T e : o) {
            e.m();
            System.out.print("-");
        }
    }

    public static void main(String[] args) {
        Y e1 = new Y();
        X e2 = (X) new Y();
        T e3 = new X();
        X e4 = new Y();

        exe(e1, e2, e3, (T) e4);
    }
}
```

10. (1 b) Synchronizovať nestatickú metódu v Jave znamená zabezpečiť

- (a) nemennosť tejto metódy v odvodených triedach
- (b) rýchlejšie vykonávanie tejto metódy
- (c) rovnomerné prepínanie nití pri vykonávaní tejto metódy
- (d) zamknutie aktuálneho objektu (**this**) kým sa táto metóda nevykoná
- (e) zamknutie objektu triedy (**class**) kým sa táto metóda nevykoná

10 b

1 a

2 a, c, d, e

3 c

4 e

5 a

6 d

7 c

8 b; *a* je tiež akceptované, lebo môže byť interpretované ako skrývanie informácií pre chýbajúcu referenciu na programátorov (“zabezpečilo utajenie údajov a kódu *od programátorov*”)

9 Y-Y-CX-Y-

10 d