

## Objektovo-orientované programovanie 2017/18

doc. Ing. Valentino Vranić, PhD., ÚISI FIIT STU

Test – náhradný termín – 10. máj 2018

Priezvisko:	tlačeným písmom
Meno:	

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) Synchronizované metódy v Java

- (a) sa rovnomerne striedajú
- (b) predstavujú kritické regióny programu
- (c) sa spúšťajú a končia naraz
- (d) bránia uviaznutiu
- (e) predstavujú nite programu

### 2. (1 b) Vyhodenie výnimky v Java

- (a) opravuje vzniknutú chybu
- (b) signalizuje výnimočnú situáciu hlavnej metóde v programe
- (c) uvoľňuje pamäť od zbytočných objektov
- (d) posielajú správu vedúcemu programátorovi
- (e) signalizuje výnimočnú situáciu klientskemu kódu

### 3. (1 b) Metóda, ktorej v programe v Java predchádza kľúčové slovo **public**

- (a) sa dá zavolať len v rámci balíka, do ktorého patrí
- (b) sa dá zavolať len v hierarchii tried, do ktorej patrí
- (c) má na rozdiel od **private** metód jej zdrojový kód je verejne dostupný
- (d) sa dá zavolať odkiaľkoľvek
- (e) musí byť predpísaná rozhraním, ktoré trieda, do ktorej patrí, implementuje

### 4. (1 b) Zapuzdrenie v objektovo-orientovanom programovaní

- (a) predstavuje kritérium pre použitie agregácie
- (b) umožňuje spájanie objektov
- (c) predstavuje spôsob tvorenia hierarchie
- (d) umožňuje znížiť závislosť klientskeho kódu
- (e) umožňuje, aby sa objekt uplatnil namiesto objektu jeho nadtypu

### 5. (1 b) Pre ktorý návrhový vzor je tento kód charakteristický (každá trieda a rozhranie vo vlastnom súbore)?

```
public class A {
    private B b;
    public A(B b) {
        this.b = b;
    }
    public void m(...) {
        ...
        b.op();
        ...
    }
}

public interface B {
    ... op(...);
}

public class B1 implements B {
    public ... op(...) {
        ...
    }
}

public class B2 implements B {
    public ... op(...) {
        ...
    }
}
```

- (a) Observer
- (b) Visitor
- (c) MVC
- (d) Composite
- (e) Strategy

### 6. (1 b) Program v Java v obsahuje triedu M. Dá sa pristúpiť k objektu, ktorý predstavuje samotnú triedu, a ak áno, tak ako?

- (a) áno, použitím výrazu **M.class**
- (b) nie
- (c) áno, použitím výrazu **class(M)**
- (d) áno, použitím výrazu **new <M>()**
- (e) áno, použitím výrazu **new M()**

### 7. (1 b) V objektovo-orientovanom programe hlavná funkcionálna typicky

- (a) je obsiahnutá v konštruktoroch
- (b) vzniká dedením
- (c) vzniká v interakcii objektov
- (d) je zabezpečená statickými metódami
- (e) je obsiahnutá v metóde **main()**

### 8. (1 b) Inštancia rozhrania v Java

- (a) je polymorfná
- (b) je abstraktná
- (c) je generická
- (d) je statická
- (e) nejestvuje

9. (1 b) Daná je trieda v Jave:

```
class C implements Serializable {
    public String id;
    public List<C> l = new ArrayList<>();

    public C(String id) {
        this.id = id;
    }
    public static void main(String[] args)
        throws ClassNotFoundException, IOException {

        C x = new C("x");
        C y = new C("y");
        C z = new C("z");
        C w = new C("w");

        y.l.add(x);
        y.l.add(w);
        z.l.add(x);
        z.l.add(w);
        w.l.add(x);

        ObjectOutputStream out =
            new ObjectOutputStream(
                new FileOutputStream("f.out"));
        out.writeObject(y);
        out.close();
    }
}
```

Ktoré z objektov x, y, z a w sa zaznamenajú do súboru f.out (uveďte príslušné písmená; na poradí nezáleží)?

10. (1 b) Čo sa vypíše po spustení nasledujúceho programu v Jave?

```
abstract class A {
    public void m() {
        System.out.print("A");
    }
}
abstract class S extends A {
    public void m() {
        super.m();
        System.out.print("S");
    }
}
class D extends S {
    public void m() {
        System.out.print("D");
    }
}
class F extends D {
    public void m() {
        super.m();
        System.out.print("F");
    }
}
class M {
    public static void exe(A... o) {
        for (A e : o) {
            e.m();
            System.out.print("-");
        }
    }
    public static void main(String[] args) {
        A e1 = (S) new D();
        F e2 = new F();
        S e3 = new D();
        D e4 = new F();

        exe(e1, (D) e2, e3, (A) e4);
    }
}
```

**Objektovo-orientované programovanie 2017/18**

doc. Ing. Valentino Vranić, PhD., ÚISI FIIT STU

Test – náhradný termín – 10. máj 2018

10 b

1 b

2 e

3 d

4 d

5 e

6 a

7 c

8 e

9 x, y, w

10 D-DF-D-DF-