Objektovo-orientované programovanie **B**

Ing. Valentino Vranić, PhD., ÚISI FIIT STU

Skúška — 23. jún 2006 (opravný termín)

Priezvisko:	
Meno:	

1b	
2b	
3b	

	a	b	c	d	e
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					
17					
18					

Test trvá 120 minút.

V otázkach 1–18 je len jedna možnosť správna. Vyznačte svoju odpoveď krížikom do veľkej tabuľky (malú tabuľku nevypĺňajte). Hodnotia sa len odpovede vyznačené v tabuľke.

V prípade opravy jasne vyznačte ktorú odpoveď vyberáte. Každá správna odpoveď má hodnotu vyznačenú v otázke. Nesprávna odpoveď, vyznačenie viac odpovedí alebo nejednoznačné vyznačenie má hodnotu 0 bodov. Postup riešenia sa pre otázky 1–18 nehodnotí.

Odpovede na otázky 19 a 20 píšte na list s poslednými tromi otázkami. Na ňom tiež uveďte meno.

1. (1 b) V triede ktorá dedí od rozhrania je možné zadefinovať

(a) len ďalšie polia
(b) len to čo predpisuje rozhranie
(c) len konkrétne metódy
(d) len metódy ktoré predpisuje rozhranie
(e) aj iné metódy než predpisuje rozhranie

2. (1 b) Iterátory v Java API uľahčujú

(a) rušenie prvkov v zoskupeniach
(b) prechádzanie zoskupeniami
(c) pridávanie prvkov do zoskupení
(d) opakovanie vykonávania kódu
(e) volania abstraktných metód

3. (1 b) Abstraktná metóda v Jave

(a) nemôže byť volaná
(b) nemôže mať argumenty
(c) môže mať telo
(d) nemôže mať návratovú hodnotu
(e) môže sa vyskytovať v hocijakej triede

4. (1 b) V jazyku C++ inline funkcie

(a) predstavujú výlučne jednoriadkové funkcie
(b) umožňujú polymorfizmus
(c) nemajú telo
(d) prekladač nahrádza kódom na mieste ich volaní
(e) nemôžu obsahovať volania iných funkcií

5. (2 b) Aby bolo možné využiť polymorfizmus v jazyku C++, zodpovedajúce metódy určené na prekonávanie musia byť

(a) abstraktné
(b) virtuálne
(c) verejné
(d) konštantné
(e) statické

6. (2 b) Nech A je trieda ktorá poskytuje verejnú metódu **int** m(). Trieda C je definovaná takto:

```
class C {
   int i = (new A()).m();
}
```

Táto definícia je

(a) nekorektná
(b) korektná jedine ak je metóda m() finálna
(c) korektná jedine ak je metóda m() synchronizovaná
(d) korektná jedine ak je metóda m() statická
(e) korektná

7. (2 b) Agregácia v objektovo-orientovanom programovaní

(a) stanovuje kritéria pre použitie abstraktných tried
(b) predstavuje skrytie implementácie objektu
(c) predstavuje kritérium pre použitie dedenia
(d) predstavuje spájanie objektov do väčších celkov
(e) umožňuje, aby sa objekt uplatnil namiesto objektu jeho nadtypu

8. (2 b) Prístup **protected** je vhodné použiť pri takých prvkoch triedy ku ktorým chceme pristupovať len

(a) v triedach toho istého balíka
(b) v odvodených triedach toho istého balíka
(c) v odvodených triedach
(d) v odvodených triedach a v triedach toho istého balíka
(e) v danej triede

9. (2 b) Diagram prípadov použitia v jazyku UML znázorňuje

(a) štruktúru systému
(b) postupnosť správ prenášaných medzi objektmi
(c) funkcionalitu z pohľadu používateľa alebo iného systému
(d) vzťahy medzi inštanciami tried v určitom okamihu vykonávania programu
(e) triedy a vzťahy medzi nimi

10. (2 b) V jazyku AspectJ bodové prierezy (pointcuts) slúžia na

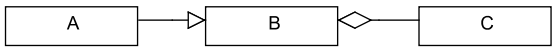
(a) pretínanie tried
(b) doplnenie nových prvkov do tried
(c) modifikáciu vykonávania programu
(d) zachytenie bodov spájania
(e) definovanie nových metód

11. (2 b) Aspekty v aspektovo-orientovanom programovaní

(a) modifikujú štruktúru a vykonávanie programu
(b) definujú miesta vo vykonávaní programu
(c) definujú rozdelenie kódu do komponentov
(d) robia kód zapleteným a roztrúseným
(e) urýchľujú vykonávanie programu

12. (3 b) Návrhový vzor Observer slúži na

(a) pridávanie operácií nad objektmi daných tried bez ich zmeny
(b) zabránenie vytváraniu viac než jednej inštancie danej triedy
(c) definovanie závislosti stavu viacerých objektov od ďalšieho objektu
(d) zabránenie rozširovania kódu
(e) pridávanie vzťahov medzi triedami bez ich zmeny



Obrázok 1: Diagram pre otázku 18.

13. (3 b) Vytvorili ste tlačidlo t ako komponent rámca Swing a zviditeľneli ste okno, ktoré ho obsahuje. Potrebujete zmeniť označenie (label) tlačidla na text Abc. Urobite to volaním

(a) t.changeText("Abc")
(b) t.invokeLater(t.changeText("Abc");}});
(c) (**new** Runnable(SwingUtilities.invokeLater(
t.changeText("Abc");}}))).run();
(d) SwingUtilities.invokeLater(**new** Runnable() {
public void run() {w.changeText("Novy text");}});
(e) SwingUtilities.run(**new** Runnable() {
public void invokeLater() {
w.changeText("Novy text");}});

14. (3 b) V jazyku C++ implementujete operácie nad komplexnými číslami. V čase implementácie ešte neviete či zložky komplexných čísel budú môcť byť len celočíselné alebo aj decimálne. Na vyriešenie tohto problému použijete

(a) virtuálne funkcie
(b) preťaženie funkcií
(c) šablóny
(d) preťaženie operátorov
(e) statické funkcie

15. (3 b) Daný je kód v Jave na obr. 2. Vykonaním týchto príkazov:

```
I i = new B();
i.m();
((A)i).m();
((I)i).m();
```

(a) vypíše sa bbb
(b) vznikne chyba v poslednom riadku
(c) vypíše sa aaa
(d) vypíše sa bb
(e) nevypíše sa nič

16. (3 b) Ku kódu v Jave na obr. 2 je daná nasledujúca trieda:

```
class M {
   static int m(Class<? extends I> T, I... o) {
      int i = 0;
      for (I e : o)
         if (T.isInstance(e))
            i++;
      return i;
   }

   public static void main(String... args) {
      System.out.println(
        m(B.class, new I[]{new A(), new B()}));
   }
}
```

Pri jej vykonaní

(a) vznikne výnimka
(b) vypíše sa 2
(c) vypíše sa 0
(d) vypíše sa 1
(e) vypíše sa 3

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

class A implements I {
   public void m() { System.out.print("a"); }
}

class B extends A {
   public void m() { System.out.print("b"); }
}
```

Obrázok 2: Kód pre otázky 15 a 16.

17. (3 b) Daný je nasledujúci program v Jave:

```
class C {
   C x;
   void m() {
      x = new SubC();
      **1**.f();
   }
   public static void main(String[] args) {
      **2**.m();
   }
}

class SubC **3** {
   void f() {
      System.out.println("f");
   }
}
```

Ktoré fragmenty kódu treba v tomto programe doplniť, aby sa pri jeho vykonaní vypísalo f?

(a) **1**: x **2**: new C()
3: extends C
(b) **1**: ((SubC)x) **2**: new C()
3: extends C
(c) **1**: x **2**: new C()
3: implements C
(d) **1**: SubC **2**: C
3: extends C
(e) **1**: ((SubC)x) **2**: new C()
3: nič

Objektovo-orientované programovanie **B**
Ing. Valentino Vranić, PhD., ÚISI FIIT STU
Skúška — 23. jún 2006 (opravný termín)

Priezvisko:	
Meno:	

18. (3 b) Daný je nasledujúci kód v Jave:

```
class **1** extends **2** { . . . }

class **3** {
    **4** o;
    . . .
}
```

Aby kód zodpovedal diagramu na obr. 1, fragmenty kódu označené číslami 1–4 v danom poradí treba nahradiť identifikátormi:

(a) B, A, A, C
(b) A, B, B, C
(c) B, A, C, A
(d) C, B, B, A
(e) B, C, A, B

19. (6 b) Vysvetlite princíp otvorenosti a uzavretosti kódu. Uveďte príklad (slovne a so základom kódu).

20. (10 b) V programe na prácu s grafickými útvarmi zobrazenie útvarov môže byť v rôznej kvalite (napr. len hrany, s výplňou apod.). Naznačte hierarchiu útvarov. Aplikujte idióm double dispatch na zobrazenie útvarov: uveďte základ kódu s vysvetlením. Uveďte príklad použitia idiómu double dispatch pri vyvolaní zobrazenia.

Objektovo-orientované programovanie **B**
Ing. Valentino Vranić, PhD., ÚISI FIIT STU
Skúška — 23. jún 2006 (opravný termín)

1 e

2 b

3 a

4 d

—

5 b

6 e

7 d

8 d

9 c

10 d

11 a

—

12 c

13 d

14 c

15 a

16 d

17 b

18 b

55