Objektovo-orientované programovanie **B**

Ing. Valentino Vranić, PhD., ÚISI FIIT STU

Skúška — 5. jún 2006

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 čistý list, ktorý ste dostali. Na ňom uveďte meno a skupinu.

1. (1 b) Pri dedení od ďalšej triedy je možné v danej triede zadefinovať

(a) len ďalšie polia
(b) len ďalšie metódy
(c) len metódy, ktorých názvy nejestvujú v nadtriede
(d) ďalšie polia a metódy
(e) len metódy, ktorých názvy jestvujú v nadtriede

2. (1 b) Iterátory v Java API uľahčujú

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

3. (1 b) Abstraktná trieda v Jave

(a) nemôže mať metódy
(b) môže mať len abstraktné metódy
(c) nemôže mať polia
(d) nemôže mať prekonávajúce metódy
(e) môže mať metódy

4. (1 b) Na rozdiel od Javy jazyk C++

(a) má explicitné deštruktory
(b) automaticky vytvára objekty
(c) automaticky ruší nereferencované objekty
(d) má explicitné konštruktory
(e) má implicitné deštruktory

5. (2 b) Prístup **private** je vhodné použiť pri takých prvkoch triedy ku ktorým chceme pristupovať len

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

6. (2 b) Nech o je objekt triedy ktorá poskytuje verejnú metódu **int** m(). Reťazec r je definovaný takto:

Object r[] = **new** Object[o.m()];

Táto definícia je

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

7. (2 b) V porovnaní s objektovo-orientovaným programovaním aspektovo-orientované programovanie umožňuje

(a) oddelenie pretínajúcich záležitostí
(b) rýchlejšie vykonávanie programu
(c) tvorbu modulov
(d) rozdelenie kódu do komponentov
(e) zlúčenie pretínajúcich záležitostí

8. (2 b) Diagram sekvencií v jazyku UML znázorňuje

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

9. (2 b) Generickým triedam v Jave v jazyku C++ zodpovedajú

(a) virtuálne triedy
(b) abstraktné triedy
(c) šablóny (template)
(d) štruktúry (struct)
(e) iterátory

10. (2 b) V jazyku AspectJ videnia (advices) slúžia na

(a) zachytenie bodov spájania
(b) doplnenie nových prvkov do tried
(c) definovanie bodov spájania
(d) definovanie nových metód
(e) modifikáciu vykonávanie programu v bodoch spájania

11. (2 b) Zapuzdrenie v objektovo-orientovanom programovaní

(a) stanovuje kritéria pre použitie abstraktných tried
(b) predstavuje skrytie implementácie objektu
(c) predstavuje spôsob tvorenia hierarchie
(d) predstavuje kritérium pre použitie agregácie
(e) umožňuje, aby sa objekt uplatnil namiesto objektu jeho nadtypu

12. (3 b) Na sledovanie kliknutia na tlačidlo v rámci Swing by ste

(a) vytvorili niť udalosti a registrovali ju pre tlačidlo
(b) v ďalšej niti sledovali v slučke, či používateľ neklikol keď kurzor bol na tlačidle
(c) registrovali tlačidlo v niti na odosielanie udalosti
(d) vytvorili prijímač udalosti (ActionListner) a registrovali ho pre tlačidlo
(e) použili rozloženie FlowLayout

13. (3 b) Návrhový vzor Visitor slúži na

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

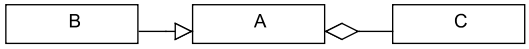
14. (3 b) Daný je nasledujúci kód v Jave:

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



Obrázok 1: Diagram pre otázku 14.

15. (3 b) V jazyku C++ ste implementovali komplexné čísla vlastnou triedou. Na zjednodušenie zápisu matematických operácií s komplexnými číslami je vhodné použiť

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

16. (3 b) Daný je nasledujúci kód v Jave:

```
if (o.class == "Kruh")
   ((Kruh)o).nakresli();
else if (o.class == "Stvorec")
   ((Stvorec)o).nakresli();
else
   ;
```

Tento kód porušuje

(a) princíp zapuzdrenia
(b) pravidlá dedenia
(c) princíp otvorenosti a uzavretosti
(d) Liskovej princíp substitúcie
(e) pravidlá polymorfizmu

17. (3 b) Daný je kód v Jave na obr. 2. Vykonaním týchto príkazov:

```
A o = new Y();
o.m();
((X)o).m();
((A)o).m();
```

(a) vznikne chyba v poslednom riadku
(b) nevypíše sa nič
(c) vypíše sa yx
(d) vypíše sa yy
(e) vypíše sa yyy

18. (3 b) Ku kódu v Jave na obr. 2 je daná nasledujúca trieda:

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

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

Pri jej vykonaní

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

```
abstract class A {
   public abstract void m();
}

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

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

Obrázok 2: Kód pre otázky 17 a 18.

19. (6 b) Vysvetlite Liskovej princíp substitúcie. Uveďte príklad (slovne a so základom kódu).

20. (10 b) Predstavte si, že vytvárate program na prácu s grafickými útvarmi. V programe je okrem iného možné meniť veľkosť grafických útvarov. Bolo by vhodné odvodiť štvorec od obdĺžnika? Zdôvodnite svoju odpoveď a uveďte základ kódu.

Objektovo-orientované programovanie **B**
Ing. Valentino Vranić, PhD., ÚISI FIIT STU
Skúška — 5. jún 2006

1 d

2 a

3 e

4 a, e

—

5 e

6 d

7 a

8 b

9 c

10 e

11 b

—

12 d

13 a

14 uznáva sa bez ohľadu na odpoveď (správne je B, A, A, C)

15 b

16 c

17 e

18 d

55