

Objektovo-orientované programovanie

Ing. Valentino Vranić, PhD., ÚISI FIIT STU

Skúška — 5. jún 2006

A

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 nevyplňajte). 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 píšete na čistý list, ktorý ste dostali. Na ňom uveďte meno a skupinu.

1. (1 b) Abstraktná trieda v Jave

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

2. (1 b) Pri dedení od ďalšej triedy je možné v danej triede zdefinovať

- (a) len ďalšie polia
- (b) ďalšie polia a metódy
- (c) len ďalšie metódy
- (d) len metódy, ktorých názvy nejstávajú v nadtriede
- (e) len metódy, ktorých názvy jstávajú v nadtriede

3. (1 b) Na rozdiel od Javy jazyk C++

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

4. (1 b) Iterátory v Java API uľahčujú

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

5. (2 b) V porovnaní s objektovo-orientovaným programovaním aspektovo-orientované programovanie umožňuje

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

6. (2 b) Generickým triedam v Jave v jazyku C++ zodpovedajú

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

7. (2 b) Nech o je objekt triedy ktorá poskytuje verejnú metódu `int m()`. Refazec r je definovaný takto:

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

Táto definícia je

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

8. (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 danej triede
- (d) v triedach toho istého balíka
- (e) v odvodených triedach toho istého balíka

9. (2 b) Diagram sekvencií v jazyku UML znázorňuje

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

10. (2 b) Zapuzdrenie v objektovo-orientovanom programovaní

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

11. (2 b) V jazyku AspectJ videnia (advice) slúžia na

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

12. (3 b) Návrhový vzor Visitor slúži na

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

13. (3 b) Ku kódu v Jave na obr. 1 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 0
- (e) vypíše sa 3

14. (3 b) Daný je kód v Jave na obr. 1. 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 yyy
- (d) vypíše sa yx
- (e) vypíše sa yy

```
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 1: Kód pre otázky 13 a 14.

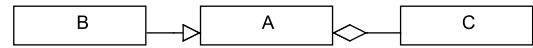
15. (3 b) Daný je nasledujúci kód v Jave:

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

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

Aby kód zodpovedal diagramu na obr. 2, 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 2: Diagram pre otázku 15.

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) pravidiel dedenia
- (c) Liskovej princíp substitúcie
- (d) princíp otvorenosti a uzavretosti
- (e) pravidiel polymorfizmu

17. (3 b) V jazyku C++ ste implementovali komplexné čísla vlastnou triedou. Na zjednodušenie zápisu matematických operácií s komplexnými číslami je vhodné použiť

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

18. (3 b) Na sledovanie kliknutia na tlačidlo v rámci Swing by ste

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

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.

1 b

2 b

3 d, e

4 c

—

5 d

6 e

7 b

8 c

9 e

10 a

11 d

—

12 b

13 e

14 c

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

16 d

17 a

18 b

55