Objektovo-orientované programovanie
doc. Ing. Valentino Vranić, PhD., ÚISI FIIT STU
Skúška (opravný termín) – 17. jún 2011

Priezvisko:

Meno:

1 b	
2 b	
3 b	

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

Skúška trvá 75 minút.
V otázkach 1–17 je len jedna možnosť správna. Vyznačte svoju odpoveď krížikom do tabuľky. Hodnotia sa len odpovede v tabuľke.
V prípade opravy jasne vyznačte odpoveď, ktorá platí. 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 1–17 nehodnotí. Poškodený list nebude uznaný. Odpoveď na otázku 18 píšte na prídavný list.

1. (1 b) Prúd údajov (stream) v Java API sa otvára
(a) príkazom System.open()
(b) jeho prvým použitím
(c) jeho metódou open()
(d) jeho konštruktorom
(e) príkazom IOStream.open()

2. (1 b) Modifikátory prístupu v Jave slúžia na
(a) obmedzenie viditeľnosti prvkov tried, pričom **private** je najvyššie obmedzenie
(b) ochranu údajov, pričom **private** je najvyšša ochrana
(c) ochranu prvkov triedy pred úpravou
(d) obmedzenie viditeľnosti prvkov tried, pričom **protected** je najvyššie obmedzenie
(e) ochranu údajov, pričom **protected** je najvyšša ochrana

3. (1 b) Dedenie triedy od rozhrania v Jave predstavuje dedenie
(a) správania
(b) rozšírenia
(c) štruktúry
(d) implementácie
(e) agregácie

4. (1 b) Z hľadiska objektovo-orientovaného prístupu rozsiahle používanie statických metód
(a) je žiaduce, lebo sa vykonajú rýchlejšie
(b) nie je žiaduce, lebo sa nededia
(c) nie je žiaduce, lebo nepodporujú zapuzdrenie
(d) je žiaduce, lebo umožňujú polymorfizmus
(e) nie je žiaduce, lebo sa nedajú prekonať

5. (2 b) Synchronizácia statickej metódy
(a) znamená uzamknutie objektu **this** pre hocijaký iný synchronizovaný prístup
(b) znamená uzamknutie objektu triedy pre hocijaký iný synchronizovaný prístup
(c) znamená uzamknutie objektu **this** pre hocijaký iný prístup
(d) znamená uzamknutie objektu triedy pre hocijaký iný prístup
(e) nie je možná

6. (2 b) V diagrame sekvencií jazyka UML horizontálne šípky s plnou čiarou označujú
(a) zľava doprava vyvolanie operácie, a sprava doľava návrat hodnoty
(b) tok údajov medzi objektmi
(c) prenos parametrov operácie v smere šípky
(d) vyvolanie operácie
(e) návrat hodnoty

7. (2 b) Jeden z rozdielov medzi abstraktnou triedou a rozhraním v Jave je ten, že
(a) je možné vytvárať inštancie abstraktných tried, ale nie aj rozhraní
(b) konkrétne triedy môžu dediť od abstraktnej triedy, ale nie od rozhrania
(c) rozhranie môže dediť od abstraktnej triedy, ale nie naopak
(d) je možné vytvárať inštancie rozhraní, ale nie aj abstraktných tried
(e) abstraktná trieda môže dediť od rozhrania, ale nie naopak

8. (2 b) Výraz call(abc∗(..)) v jazyku AspectJ znamená
(a) zachytenie prvého volania metódy, ktorej názov začína na abc
(b) vyvolanie metódy call() pred všetkými metódami, ktorých názov začína na abc
(c) zachytenie volania všetkých metód, ktorých názov začína na abc
(d) vyvolanie všetkých metód, ktorých názov začína na abc
(e) vyvolanie prvej metódy, ktorej názov začína na abc

9. (2 b) V jazyku C++ ste implementovali sčítavanie a odčítavanie farieb. Na zjednodušenie zápisu týchto operácii vhodné je použiť
(a) preťaženie operátorov
(b) statické funkcie
(c) virtuálne funkcie
(d) virtuálne operátory
(e) preťaženie funkcií

10. (2 b) Ak je potrebné pracovať s V/V systémom Javy, príkaz

import java.io.∗;

(a) nie je nevyhnutný v programe, ale pri kompilácii musí byť zadaný parameter java.io.∗
(b) je zbytočný
(c) je nevyhnutný
(d) nie je nevyhnutný
(e) nie je nevyhnutný, ale potrebné triedy V/V systému musia byť skopírované do daného programu

11. (2 b) Okno v rámci Swing sa dá vytvoriť zavolaním
(a) konštruktora triedy JFrame
(b) statickej metódy EventQueue.newWindow()
(c) konštruktora triedy SwingWindow
(d) metódy newWindow() triedy JFrame
(e) statickej metódy SwingUtils.newWindow()

12. (2 b) V jazyku UML hrana znázornená prerušovanou čiarou so šípkou v smere od triedy k rozhraniu a označením «use» znamená že
(a) trieda vytvára inštanciu rozhrania
(b) trieda implementuje metódy predpísané rozhraním
(c) trieda volá niektorú z metód rozhrania
(d) trieda volá všetky metódy rozhrania
(e) trieda ovplyvňuje rozhranie

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. Urobíte to volaním

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

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

```
abstract class A {
   int i = 0;
   public abstract void m();
}

class X extends A {
   public void m() { i--; }
}

class Y extends X {
   public void m() { i++; }
}
```

Vykonaním týchto príkazov:

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

(a) atribút i nadobudne hodnotu 3
(b) atribút i nadobudne hodnotu -3
(c) vznikne chyba v poslednom riadku
(d) atribút i nadobudne hodnotu 1
(e) atribút i nadobudne hodnotu -1

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

```
public class El<T> {
   private T data;
   private El**1** next;
   public El(**2**) { data = d; }
   public void add(**2**) { next = new **3**; }
}
```

Čím treba nahradiť označené časti kódu z obr. **??**, aby metóda add() pripájala ďalší prvok so zadaným údajom data k aktuálnemu prvku pre rôzne typy údajov?

(a) `**1**`: <T> `**2**`: T d `**3**`: El<T>(d)
(b) `**1**`: `**2**`: T d `**3**`: El(d)
(c) `**1**`: <T> `**2**`: <T> d `**3**`: El(<T> d)
(d) `**1**`: `**2**`: d `**3**`: El<d>(d)
(e) `**1**`: <data> `**2**`: data d `**3**`: El<data>(d)

16. (3 b) V programe je každý druh geometrického útvaru reprezentovaný triedou. Rozmery útvarov sú reprezentované atribútmi a ich čítanie a zmena zabezpečené prostredníctvom metód. Je vhodné odvodiť triedu, ktorá reprezentuje štvorec, od triedy, ktorá reprezentuje obdĺžnik?

(a) áno
(b) áno, ale len ak sú príslušné atribúty **private** a metódy **public**
(c) nie, lebo to porušuje princíp otvorenosti a uzavretosti
(d) nie, lebo to porušuje Liskovej princíp substitúcie
(e) áno, ale prekonaním metód musí byť zabezpečené, aby sa pri štvorci výška menila zároveň so šírkou

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

```
interface I {
   void op();
}
class A implements I {
   public void op() { System.out.print("a"); }
}
class B extends A {
   public void op() { System.out.print("b"); }
}
class C implements I {
   protected A a;
   public C(A a) { this.a = a; }
   public void op() {
      System.out.print("c");
      a.op();
   }
}
class D extends C {
   public D(A a) { super(a); }
   public void op() {
      System.out.print("d");
      a.op();
   }
}
class M {
   public void opi(I... i) {
      for (I x : i)
         x.op();
   }
   public static void main(String[] args) {
      A o1 = new B();
      B o2 = new B();
      C o3 = new C(o1);
      D o4 = new D(o2);
      new M().opi(o1, o2, o3, o4);
   }
}
```

Po spustení tohto kódu sa vypíše

(a) abcd
(b) abcadb
(c) bbcbdb
(d) bbcd
(e) abcbdb

Priezvisko:

Meno:

18. (10 b) V informačnom systéme banky sa okrem iného vedie evidencia o bežných účtoch a úveroch klienta. Systém poskytuje možnosť výpisu stavu všetkých účtov a úverov pre zvoleného klienta vzhľadom na rôzne kritériá ako sú celková dlžná suma (pre účet sa berie reálny debet, a ak nie je, tak nula) a schválený úver (pre účet sa berie povolený debet). Dá sa očakávať, že v budúcnosti bude potrebné podporiť ďalšie kritériá pre výpis stavu.

Nakreslite diagram tried s najvýznamnejšími vzťahmi, operáciami a atribútmi, ktoré vyplývajú z uvedeného opisu. Napíšte zodpovedajúci kód v Jave. Aplikujte mechanizmy objektovo-orientovaného programovania a vysvetlite ich úlohu. Ak je to vhodné, aplikujte niektorý z návrhových vzorov.

Objektovo-orientované programovanie
doc. Ing. Valentino Vranić, PhD., ÚISI FIIT STU
Skúška (opravný termín) – 17. jún 2011

1 d

2 a

3 a

4 e

—

5 b

6 d

7 e

8 c

9 a

10 d

11 a

12 c

—

13 b

14 b

15 a

16 d

17 c

45