

Objektovo-orientované programovanie

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

Skúška (opravný termín) – 25. jún 2012



Priezvisko:

Meno:

1 b	
2 b	
3 b	

Skúška trvá 75 minút.

V otázkach 1–12 je len jedna možnosť správna. Vyznačte svoju odpoveď krížikom do tabuľky. Odpovede na otázky 13–15 vpíšte do tabuľky.

Pri otázkach 1–15 sa hodnotia len odpovede v tabuľke (bez postupu). V prípade opravy jasne vyznačte odpoveď, ktorá platí. Nesprávna odpoveď, vyznačenie viac odpovedí alebo nejednoznačné vyznačenie má hodnotu 0 bodov. Poškodený list nebude uznaný.

Odpoveď na otázku 18 píše výlučne na list, na ktorom sa nachádza jej znenie.

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

1. (1 b) Príkaz

`import java.util.*;`

- (a) fyzicky pripojí typy balíka `java.util` k programu bez typov v podbalíkoch
- (b) sprístupní priestor názvov všetkých typov balíka `java.util`, ale bez typov v podbalíkoch
- (c) fyzicky pripojí len skutočne použité typy balíka `java.util` k programu bez typov v podbalíkoch
- (d) sprístupní priestor názvov všetkých typov balíka `java.util` vrátane typov v podbalíkoch
- (e) fyzicky pripojí všetky typy balíka `java.util` k programu vrátane typov v podbalíkoch

2. (1 b) Daný je nasledujúci kód:

```
for (Object o : l) {  
    if (o.class == "StrasnyObor")  
        ((StrasnyObor)o).utoc();  
    else if (o.class == "PlachyObor")  
        ((PlachyObor)o).utec();  
    else  
        ;  
}
```

Tento kód porušuje

- (a) princíp polymorfizmu
- (b) Liskovej princíp substitúcie
- (c) princíp generalizácie a špecializácie
- (d) princíp otvorenosti a uzavretosti kódu
- (e) princíp abstrakcie

3. (1 b) Pri dedení je možné v odvodenej triede zdefinovať

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

4. (2 b) Na rozdiel od objektovo-orientovaného programovania aspektovo-orientované programovanie umožňuje

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

5. (2 b) Najdôležitejšie kritérium pre použitie dedenia je

- (a) miera použitia abstraktných tried
- (b) existencia vzťahu typ-podtyp
- (c) zachovanie zapuzdrenia
- (d) znovupoužitie atribútov a metód nadtriedy v podtriedach
- (e) agregácia tried

6. (2 b) Diagram sekvencií v jazyku UML znázorňuje

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

7. (2 b) Dedeniu typu `extend` medzi triedami v Jave v jazyku C++ zodpovedá dedenie typu:

- (a) `private`
- (b) `protected` s virtuálnymi triedami
- (c) `public` s virtuálnymi triedami
- (d) `protected`
- (e) `public`

8. (2 b) Synchronizácia statickej metódy

- (a) nie je možná
- (b) znamená uzamknutie objektu `this` pre hocikajký iný synchronizovaný prístup
- (c) znamená uzamknutie objektu `this` pre hocikajký iný prístup
- (d) znamená uzamknutie objektu triedy pre iný hocikajký synchronizovaný prístup
- (e) znamená uzamknutie objektu triedy pre hocikajký iný prístup

9. (2 b) Implementácie rozhraní prijímačov vo Swingu predstavujú

- (a) pozorovateľov vo vzore `Observer`
- (b) návštevníkov vo vzore `Visitor`
- (c) konkrétne stratégie vo vzore `Strategy`
- (d) prvky vo vzore `Visitor`
- (e) kontexty vo vzore `Strategy`

10. (2 b) V jednej z tried programu sa realizuje výpočet na základe dodaných koeficientov. V prípade ich nesúlady, ktorý sa dá zistiť ich porovnaním, výpočet nemá zmysel. Toto treba ošetriť

- (a) vyhodnotením výnimky typu vytvoreného špeciálne za týmto účelom a jej ošetrením v metóde, ktorá výpočet potrebovala
- (b) priamo v metóde, ktorá realizuje výpočet, zastavením programu pre chybu
- (c) vyhodnotením všeobecnej výnimky typu `Exception` a jej ošetrením v metóde, ktorá výpočet potrebovala
- (d) priamo v metóde, ktorá realizuje výpočet, pokusom o nápravu výsledku
- (e) vyhodnotením výnimky typu `RuntimeException`

11. (2 b) Pri rozhodovaní o použití dedenia je kľúčové sledovať

- (a) možnosti zovšeobecnenia podtypu nadtypom
- (b) možnosť uplatnenia objektov podtypu namiesto objektov nadtypu
- (c) možnosti abstrakcie podtypu nadtypom
- (d) počet spoločných atribútov a metód podtypu a nadtypu
- (e) mieru príbuznosti objektov modelovaných podtypom a nadtypom v realite

12. (2 b) Aký návrhový vzor by sa dal použiť na zastrešenie rôznych spôsobov riešenia toho istého problému?

13. (3 b) Čo sa stane po spustení kódu z obr. 1?

```
class X {
    public void m() { System.out.print("x"); }
}
class Y extends X {
    public void m() { System.out.print("y"); }
}
class A {
    public Runnable r(final X x) {
        return new Runnable(){
            public void run(){
                x.m();
            }
        };
    }
    public static void main(String[] args) {
        X ox = new Y();
        Y oy = new Y();

        Runnable a = (new A()).r(ox);
        Runnable b = (new A()).r(oy);
        Runnable c = (new A()).r((X)oy);
        a.run();
        b.run();
        c.run();
    }
}
```

Obr. 1: Kód pre otázky 13 a 14.

14. (3 b) Ku kódu v Jave na obr. 1 je daná nasledujúca trieda:

```
class M {
    static void m(Class<? extends X> T, X... o) {
        for (X e : o)
            if (T.isInstance(e))
                System.out.print("i");
    }

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

Čo sa stane po jej vykonaní?

15. (3 b) Medzi ktorými časťami architektonického vzoru Model-View-Controller figuruje návrhový vzor Observer a ktorá časť vystupuje v akej role?

Priezvisko:

Meno:

16. (10 b) V informačnom systéme fakulty sa okrem iného vedie evidencia o výskumných projektoch a výskumných skupinách. Systém poskytuje možnosť výpisu súhrnných informácií o projektoch ako sú napríklad počet účastníkov, začiatok a koniec realizácie/existencie, schválené prostriedky atď. V ďalších verziách možno vznikne potreba za ďalšími výpismi.

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

Objektovo-orientované programovanie

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

Skúška (opravný termín) – 25. jún 2012



1 b

2 d

3 a

—

4 a

5 b

6 e

7 c

8 d

9 c

10 a

11 b

12 Strategy

—

13 vypíše sa yyy

14 vypíše sa ii

15 Model a View; Model je Subject, a View je Observer