



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 píše na prídavný list. Na ňom tiež uveďte svoje priezvisko a meno.

1. (1 b) Pri dedení triedy od inej triedy v podtriede je možné zadefinovať

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

2. (1 b) Iterátory v Java API uľahčujú

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

3. (1 b) Abstraktná trieda v Jave

- (a) môže mať len abstraktné metódy
- (b) nemôže dediť
- (c) nemôže mať atribúty
- (d) môže mať statické metódy
- (e) nemôže mať prekonávajúce metódy

4. (1 b) Objekt v objektovo-orientovanom programovaní predstavuje

- (a) inštanciu triedy
- (b) modul
- (c) triedu
- (d) inštanciu triedy alebo rozhrania
- (e) typ

5. (2 b) Daný je nasledujúci kód:

```
class XException extends Exception {}  
class A {  
    void a() throws XException {  
        if (...) { ... }  
        else throw new XException();  
    }  
}
```

```
class B {  
    void b() throws XException { new A().a(); }  
}
```

Metóda b() triedy B

- (a) musí zachytávať výnimku typu XException
- (b) je korektná
- (c) musí ošetrovať výnimku typu XException
- (d) nesmie obsahovať klauzulu **throws**
- (e) musí vyhadzovať výnimku typu XException

6. (2 b) Agregácia v objektovo-orientovanom programovaní

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

7. (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

8. (2 b) Kliknutie na tlačidlo vo Swingu zachytáva a spracúva

- (a) príslušné okno, v ktorom sa tlačidlo nachádza
- (b) samotné tlačidlo svojou metódou listen()
- (c) alternátor tlačidla
- (d) metóda main() v nekonečnej slučke
- (e) prijímač registrovaný pre dané tlačidlo

9. (2 b) Na rozdiel od objektovo-orientovaného programovania aspektovo-orientované programovanie umožňuje

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

10. (2 b) Nech o je objekt triedy, ktorá poskytuje verejnú metódu **int** m(). Pole r je definované takto:

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

Táto definícia je

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

11. (2 b) Vzťah View–Controller je v architektonickom vzore Model–View–Controller implementovaný

- (a) bez použitia návrhových vzorov
- (b) návrhovým vzorom Strategy
- (c) návrhovým vzorom Observer
- (d) návrhovým vzorom Worker Object Creation
- (e) návrhovým vzorom Composite

12. (2 b) Aký prístup je vhodné použiť pri metóde, ktorá má byť prekonaná, ale nemá byť prístupná triedam mimo jej hierarchie dedenia?

- (a) inherited
- (b) public
- (c) protected
- (d) private
- (e) implicitný prístup (bez modifikátora)

13. (3 b) V programe v jazyku C++ trieda GumovaLopta dedí od triedy Lopta a implementuje špeciálnu verziu metódy skoc(). Daný je nasledujúci kód:

```
Lopta o;  
o = new GumovaLopta();  
o->skoc();
```

Zavolá sa týmto správna metóda skoc()?

- (a) áno, ale jedine ak je metóda skoc() statická
- (b) nie
- (c) áno
- (d) áno, ale jedine ak je trieda Lopta virtuálna
- (e) áno, ale jedine ak je metóda skoc() virtuálna

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

```
class M {  
    static int m(Class<? extends I> T, I... o) {  
        int i = 5;  
        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 B(), new A(), new B()}));  
    }  
}
```

Pri jej vykonaní

- (a) vznikne výnimka
- (b) vypíše sa 4
- (c) vypíše sa 3
- (d) vypíše sa 2
- (e) vypíše sa 5

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

16. (3 b) Daný je kód v Jave na obr. 1. Vykonaním týchto príkazov:

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

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

```
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"); }  
}
```

Obr. 1: Kód pre otázky 14 a 16.

17. (3 b) V aspektovo-orientovanej implementácii vzoru Observer

- (a) vyčleňuje sa do aspektu logika vzoru, ktorá je predovšetkým v časti Observer
- (b) vyčleňuje sa do aspektu logika vzoru, ktorá je predovšetkým v časti Subject
- (c) oddeľuje sa aplikačná logika od rozhrania aplikácie
- (d) vzniká tretia časť vzoru označovaná ako Mediator medzi časťami Subject a Observer
- (e) zaniká potreba implementovať Subject

18. (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 a vysvetlite ich úlohu. Ak je to vhodné, aplikujte niektorý z návrhových vzorov.



1 c

2 a

3 d

4 a

—

5 b

6 a

7 d

8 e

9 a

10 d

11 b

12 c

—

13 e

14 c

15 e

16 a

17 b

45