

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.

1. (1 b) Abstraktná metóda v Jave

- (a) môže mať telo
- (b) nemôže mať argumenty
- (c) nemôže mať návratovú hodnotu
- (d) nemôže byť statická
- (e) môže sa vyskytovať v hocijakej triede

2. (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

3. (1 b) Atribút triedy, ktorému predchádza kľúčové slovo **private**

- (a) sa nezapíše do súboru pri serializácii objektu
- (b) je dostupný len v rámci jednej nite
- (c) je dostupný len v danej triede
- (d) je dostupný len v danej hierarchii tried
- (e) je chránený pred zápisom

4. (1 b) Prúd údajov (stream) v Java API sa otvára

- (a) príkazom System.open()
- (b) jeho konštruktorom
- (c) jeho prvým použitím
- (d) jeho metódou open()
- (e) príkazom IOStream.open()

5. (2 b) Mechanizmus friend v jazyku C++

- (a) nejestvuje
- (b) umožňuje poskytnúť prístup k neverejným prvkom triedy cudzím funkciám
- (c) sa používa na označenie user-friendly funkcionality
- (d) umožňuje poskytnúť prístup k neverejným prvkom triedy iným triedam
- (e) slúži na zapínanie a vypínanie polymorfizmu

6. (2 b) V diagrame sekvencií v jazyku UML horizontálne šípky s plnou čiarou označujú

- (a) vyvolanie operácie
- (b) zľava doprava vyvolanie operácie, a sprava doľava návrat hodnoty
- (c) prenos parametrov operácie v smere šípky
- (d) návrat hodnoty
- (e) tok údajov medzi objektmi

7. (2 b) Výraz call(abc*(..)) v jazyku AspectJ znamená

- (a) vyvolanie všetkých metód, ktorých názov začína na abc
- (b) vyvolanie metódy call() pred všetkými metódami, ktorých názov začína na abc
- (c) vyvolanie prvej metódy, ktorej názov začína na abc
- (d) zachytenie prvého volania metódy, ktorej názov začína na abc
- (e) zachytenie volania všetkých metód, ktorých názov začína na abc

8. (2 b) Synchronizácia statickej metódy

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

9. (2 b) Daný je nasledujúci kód:

```
List<R> z = new ArrayList<R>();
z.add(new X());
z.add(new Y());
```

Aby sa tento kód mohol preložiť a vykonať

- (a) R musí byť abstraktná trieda, a typy X a Y triedy od nej odvodené
- (b) typy X a Y musia byť triedy odvodené od typu R, ale X nesmie byť odvodené od Y alebo naopak
- (c) typy X a Y musia byť triedy priamo alebo nepriamo odvodené od typu R
- (d) typ X musí byť odvodený od typu Y alebo naopak
- (e) R musí byť rozhranie, a typy X a Y triedy, ktoré ho implementujú

10. (2 b) Metóda f() triedy A vyhadzuje výnimku MyException. Daná je trieda B:

```
class B {
    void m() { new A().f(); }
}
```

Metóda m() triedy B

- (a) musí vyhadzovať výnimku typu MyException
- (b) musí deklarovať, že vyhadzuje výnimku typu MyException
- (c) musí ošetrovať výnimku typu MyException
- (d) musí deklarovať alebo ošetrovať výnimku typu MyException
- (e) je korektná

11. (2 b) Kód

```
BufferedReader stdin =  
    new BufferedReader(  
        new InputStreamReader(System.in));
```

zabezpečuje

- (a) otvorenie štandardného vstupu
- (b) pohodlnejšiu prácu so štandardným vstupom
- (c) vytvorenie a otvorenie štandardného vstupu
- (d) čítanie zo štandardného vstupu
- (e) vytvorenie štandardného vstupu

12. (2 b) Návrhový vzor Strategy slúži na

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

13. (3 b) Pre architektonický vzor Model-View-Controller nie je významný ani jeden z návrhových vzorov

- (a) Visitor a Worker Object Creation
- (b) Observer a Strategy
- (c) Observer a Composite
- (d) Composite a Strategy
- (e) Composite a Worker Object Creation

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

```
class A {  
    public void m() { System.out.print("a"); }  
}  
  
class B extends A { }  
  
class C extends B {  
    public void m() { System.out.print("c"); }  
}
```

Čo sa vypíše po vykonaní týchto príkazov:

```
A b = new B();  
A c = new C();
```

```
b.m();  
((A)b).m();  
c.m();  
((B)c).m();
```

- (a) aaca
- (b) acc
- (c) aacc
- (d) aaaa
- (e) cc

15. (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) princíp otvorenosti a uzavretosti
- (c) Liskovej princíp substitúcie
- (d) pravidlá polymorfizmu
- (e) pravidlá dedenia

16. (3 b) Ku kódu v Jave na obr. 1 je daná nasledujúca trieda:

```
class M {  
    static void m(Class<? extends A> T, A... o) {  
        for (A e : o)  
            if (T.isInstance(e))  
                System.out.print("a");  
    }  
  
    public static void main(String... args) {  
        m(X.class, new A[]{new X(), new Y()});  
    }  
}
```

Jej vykonaním

- (a) vypíše sa a
- (b) vznikne výnimka
- (c) vypíše sa aaa
- (d) vypíše sa aa
- (e) nevypíše sa nič

17. (3 b) Daný je kód v Jave na obr. 1. 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 -1
- (c) atribút i nadobudne hodnotu 1
- (d) vznikne chyba v poslednom riadku
- (e) atribút i nadobudne hodnotu -3

```
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++; }  
}
```

Obr. 1: Kód pre otázky 16 a 17.

Priezvisko:

Meno:

18. (10 b) V informačnom systéme softvérovej firmy sa vedie evidencia o projektoch, oddeleniach a vývojových skupinách. Pre evidované projekty, oddelenia a vývojové skupiny systém umožňuje výpis niektorých spoločných charakteristík ako sú napríklad názov, zoznam pracovníkov, ktorých zahŕňajú, začiatok a koniec realizácie/existencie atď. V ďalších verziách možno vznikne potreba pridania nových spoločných operácií.

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 d

2 b

3 c

4 b

—

5 b

6 a

7 e

8 c

9 c

10 d

11 b

12 a

—

13 a

14 c

15 b

16 d

17 e

45