

<b>Priezvisko:</b>		<b>Body</b>
<b>Meno:</b>		

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

V otázkach je len jedna možnosť správna. Vyznačte svoju odpoveď krížikom do tabuľky uvedenej nižšie. Hodnotia sa len odpovede vyznačené v tabuľke. V prípade opravy jasne vyznačte ktorú odpoveď vyberáte.

Každá správna odpoveď má hodnotu 1 bod. Nesprávna odpoveď, vyznačenie viac odpovedí alebo nejednoznačné vyznačenie má hodnotu 0 bodov. Postup riešenia sa nehodnotí.

1. V programe v Jave sa príkaz

```
package my;
```

- (a) môže vyskytovať len v jednom súbore
- (b) môže vyskytovať len v súbore ktorý obsahuje len jednu triedu alebo rozhranie
- (c) môže vyskytovať vo viacerých súboroch toho istého programu len ak sú v tom istom adresári
- (d) nemôže vyskytovať v súboroch
- (e) môže vyskytovať vo viacerých súboroch toho istého programu

2. Abstraktná trieda v Jave

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

3. Nech  $n$  je pole triedy. Reťazec  $r$  je v jednej z metód tej istej triedy definovaný takto:

```
Integer r[] = new Integer[n];
```

Táto definícia je v Jave

- (a) nekorektná
- (b) korektná jedine ak je pole  $n$  statické
- (c) korektná jedine ak je pole  $n$  finálne
- (d) korektná
- (e) korektná jedine ak je pole  $n$  synchronizované

4. Tok údajov (stream) v Java API sa zatvára

- (a) príkazom `System.close()`
- (b) zavolaním jeho deštruktora
- (c) jeho metódou `close()`
- (d) prečítaním posledného údajov v ňom
- (e) príkazom `IOStream.close()`

5. Daný je kód v Jave na obr. 1. Čo sa vypíše po vykonaní týchto príkazov:

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

- (a) bab
- (b) aaa
- (c) baa
- (d) bbb
- (e) bba

```
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"); }  
}
```

Obrázok 1: Kód pre otázky 5, 7 a 6.

6. K triedam z obr. 1 je daný nasledujúci kód:

```
List<I> z = new ArrayList<A>();  
z.add(new B());
```

Tento kód sa

- (a) nepreloží, lebo do zoznamu list sa dajú vkladať len objekty typu A
- (b) preloží, ale padne počas vykonávania s výnimkou `ClassCastException`
- (c) preloží a vykoná korektné
- (d) nepreloží, lebo trieda `ArrayList` nie je generická
- (e) nepreloží, lebo typ referencie  $z$  nezodpovedá typu priradeného objektu

7. Daný je kód v Jave na obr. 1. Dá sa z metódy `m()` triedy B zavolať rovnomenná metóda triedy A?

- (a) áno, príkazom `super.m()`;
- (b) nie
- (c) áno, príkazom `m()`;
- (d) áno, príkazom `A.m()`;
- (e) áno, príkazom `this.m()`;

8. Trieda v objektovo-orientovanom programovaní predstavuje

- (a) typ objektu
- (b) objekt
- (c) inštanciu
- (d) rozhranie objektu
- (e) interakciu objektov

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

10. Dá sa v Jave urobiť inštancia rozhrania?

- (a) nie
- (b) áno
- (c) áno, ale len ak neobsahuje metódy
- (d) áno, ale nebudú sa dať zavolať metódy
- (e) áno, ale len ak obsahuje výlučne statické prvky

11. Daný je nasledujúci kód v Jave obsiahnutý v jednej z metód:

```
abstract class A implements I { }
interface I {
    void m();
}
A[] a = new I[5];
```

Tento kód sa

- (a) preloží a vykoná korektne
- (b) nepreloží, lebo posledný riadok je zlý
- (c) nepreloží, lebo nie je možné definovať triedy a rozhrania v metódach
- (d) preloží, ale vznikne chyba pri vykonávaní posledného riadku
- (e) nepreloží, lebo trieda A neimplementuje metódu m()

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

13. Dá sa v Jave niť vytvoriť implementáciou rozhrania Runnable?

- (a) nie
- (b) áno, ale len atomická
- (c) áno, ale len ako daemon
- (d) áno, ale na spustenie nite sa musí použiť trieda Thread
- (e) áno, ale len synchronizované

14. Daný je nasledujúci kód:

```
class E1 extends Exception { }
class E2 extends E1 { }
```

```
class A {
    void m() throws E1 {
        ...
    }
}
```

```
class B extends A {
    void m() throws E2 {
        ...
    }
}
```

Metóda m() triedy B

- (a) musí deklarovať že vyhadzuje výnimku typu E1
- (b) nesmie deklarovať že vyhadzuje žiadnu výnimku
- (c) musí ošetrovať výnimku typu E1
- (d) musí vyhadzovať výnimku typu E1
- (e) je korektná

15. Liskovej princíp substitúcie

- (a) hovorí o zapuzdrení
- (b) stanovuje kritéria pre použitie abstraktných tried
- (c) predstavuje kritérium pre použitie dedenia
- (d) predstavuje kritérium pre použitie agregácie
- (e) odčleňuje správanie objektu od jeho implementácie

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