

**Priezvisko:**

**Meno:**

1b	
2b	

Test trvá 35 minút.

V otázkach 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 nehodnotí. Len celistvý list bude akceptovaný.

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

**1. (1 b)** Liskovej princíp substitúcie stanovuje podmienky

- (a) vytvárania objektov
- (b) existencie vzťahu typ–podtyp
- (c) nahradenia prametrizovaných typov
- (d) uplatnenia objektovo-orientovaného programovania
- (e) nahradenia programu iným programom

**2. (2 b)** Daný je nasledujúci kód v Java:

```
class A {
    void m() {
        System.out.print("a ");
    }
    static void s() {
        System.out.print("sa ");
    }
}

class B extends A {
    void m() {
        System.out.print("b ");
    }
    static void s() {
        System.out.print("sb ");
    }
}
```

Čo sa vypíše po vykonaní týchto príkazov:

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

- (a) a sb sb
- (b) a sa sb
- (c) b sa sb
- (d) b sb sb
- (e) sa b sa

**3. (1 b)** Je dobré z hľadiska objektovo-orientovaného prístupu rozsiahle používanie statických metód?

- (a) Áno, lebo sa vykonávajú rýchlejšie.
- (b) Nie, lebo neumožňujú polymorfizmus.
- (c) Nie, lebo nepodporujú zapuzdrenie.
- (d) Nie, lebo sa nededia.
- (e) Áno, lebo sa dajú prekonať.

**4. (1 b)** Pomocou kľúčového slova **super** sa v Java dá zavolať

- (a) ľubovoľná metóda nadtypu
- (b) prekonávajúca metóda z prekonanej na ľubovoľnej úrovni dedenia
- (c) prekonaná metóda z prekonávajúcej na ľubovoľnej úrovni dedenia
- (d) bezprostredne prekonaná metóda z prekonávajúcej
- (e) bezprostredne prekonávajúca metóda z prekonanej

**5. (1 b)** Rozhranie v Java definuje

- (a) správanie bez implementácie
- (b) vzťahy dedenia medzi inými rozhraniami
- (c) implementáciu metód, ktorá sa dá prekonať
- (d) vzťahy dedenia medzi inými triedami
- (e) implementáciu metód, ktorá sa nedá prekonať

**6. (1 b)** Pokus o preklad a vykonanie nasledujúcej triedy v Java:

```
public class A {
    public static int f(int i) {
        return 2 * i;
    }
    public static void main(String[] args) {
        int[] a = new int[f(5)];
    }
}
```

- (a) pri vykonávaní vznikne výnimka pre nekonzistenciu typov
- (b) prebehne korektne
- (c) prekladáč hlási chybu pre nekonzistenciu typov
- (d) pri vykonávaní vznikne výnimka pre neznámu veľkosť poľa a
- (e) prekladáč hlási chybu pre neznámu veľkosť poľa a

**7. (1 b)** Iterátory v Java API uľahčujú

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

8. (2b) Daný je nasledujúci kód v Jave:

```
class MyException extends Exception {}

class A {
    A() throws MyException {
        . . .
    }
}

class B {
    void m() {
        A a = new A();
    }
}
```

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

9. (1b) Daný je nasledujúci kód v Jave:

```
while (getObject(o)) {
    if (o instanceof A)
        ((A)o).opa();
    else if (o instanceof B)
        ((B)o).opb();
    else
        ;
}
```

Tento kód porušuje

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

10. (1b) Daný je kód v Jave:

```
List<Axis> a = new ArrayList<Axis>();
```

Axis predstavuje

- (a) názov zoznamu
- (b) reťazec znakov
- (c) typ údajov uchovávaných v zozname
- (d) referenciu na údaje uchovávané v zozname
- (e) neznámu premennú

11. (2b) K triedam na obr. 1 je daný nasledujúci kód v Jave:

```
(new Q()).m();
```

Tento kód

- (a) sa vykoná úspešne a vypíše 0
- (b) sa nebude dať preložiť
- (c) vypíše 0 a vyhodí výnimku
- (d) nevypíše nič a vyhodí výnimku
- (e) sa vykoná úspešne a nevypíše nič

```
abstract class P {
    int f = 0;
    void m() {
        System.out.println(f);
    }
}

class Q extends P {
    void m(int f) {
        this.f = f;
        m();
    }
}
```

Obr. 1: Kód pre otázky 11 a 12.

12. (1b) Ak predpokladáme existenciu tried na obr. 1, príkaz new P(); v Jave

- (a) spôsobí zrušenie operačného systému
- (b) spôsobí zrušenie virtuálneho stroja Javy
- (c) bude preložený a vykonaný bez chýb
- (d) spôsobí chybu pri preklade
- (e) spôsobí výnimku

15 b

**1** b

**2** c

**3** b

**4** d

**5** a

**6** b

**7** a

**8** d

**9** e

**10** c

**11** a

**12** d