

Priezvisko:

1b	
2b	

Meno:

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) Rozhranie v Jave definuje

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

2. (1 b) Ak predpokladáme existenciu tried na obr. 1, príkaz `new P();` v Jave

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

3. (1 b) Pomocou kľúčového slova `super` sa v Jave dá zavolať

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

4. (2 b) K triedam na obr. 1 je daný nasledujúci kód v Jave:

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

Tento kód

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

```
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 2 a 4.

5. (1 b) Pokus o preklad a vykonanie nasledujúcej triedy v Jave:

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

6. (1 b) Liskovej princíp substitúcie stanovuje podmienky

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

7. (1 b) Je dobré z hľadiska objektovo-orientovaného prístupu rozsiahle používanie statických metód?

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

8. (1 b) Iterátory v Java API uľahčujú

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

9. (1 b) 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) Liskovej princípu substitúcie
- (b) princípu polymorfizmu
- (c) princípu uzavretosti blokov
- (d) princípu otvorenosti a uzavretosti kódu
- (e) princípu generalizácie a špecializácie

10. (1 b) Daný je kód v Jave:

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

Axis predstavuje

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

11. (2 b) 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 deklarovať že vyhadzuje výnimku typu MyException
- (b) musí deklarovať alebo ošetrovať výnimku typu MyException
- (c) musí vyhadzovať výnimku typu MyException
- (d) je korektná
- (e) musí ošetrovať výnimku typu MyException

12. (2 b) Daný je nasledujúci kód v Jave:

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

15 b

1 b

2 e

3 a

4 d

5 e

6 c

7 c

8 b

9 d

10 a

11 b

12 a