

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

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

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

2. (1 b) Modifikátory prístupu v Jave slúžia na

- (a) obmedzenie viditeľnosti prvkov tried, pričom **private** je najvyššie obmedzenie
- (b) ochranu prvkov triedy pred úpravou
- (c) ochranu údajov, pričom **private** je najvyššia ochrana
- (d) ochranu údajov, pričom **protected** je najvyššia ochrana
- (e) obmedzenie viditeľnosti prvkov tried, pričom **protected** je najvyššie obmedzenie

3. (1 b) Dedenie triedy od rozhrania v Jave predstavuje dedenie

- (a) implementácie
- (b) rozšírenia
- (c) štruktúry
- (d) agregácie
- (e) správania

4. (1 b) Z hľadiska objektovo-orientovaného prístupu rozsiahle používanie statických metód

- (a) nie je žiaduce, lebo sa nedajú prekonať
- (b) nie je žiaduce, lebo sa nededia
- (c) je žiaduce, lebo sa vykonávajú rýchlejšie
- (d) nie je žiaduce, lebo nepodporujú zapuzdrenie
- (e) je žiaduce, lebo umožňujú polymorfizmus

5. (2 b) V jazyku UML hrana znázornená prerušovanou čiarou so šípku v smere od triedy k rozhraniu a označením «use» znamená že

- (a) trieda implementuje metódy predpísané rozhraním
- (b) trieda volá niektorú z metód rozhrania
- (c) trieda vytvára inštanciu rozhrania
- (d) trieda ovplyvňuje rozhranie
- (e) trieda volá všetky metódy rozhrania

6. (2 b) Výraz `call(abc*(..))` v jazyku AspectJ znamená

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

7. (2 b) Ak je potrebné pracovať s V/V systémom Javy, príkaz

`import java.io.*;`

- (a) je nevyhnutný
- (b) nie je nevyhnutný v programe, ale pri kompilácii musí byť zadaný parameter `java.io.*`
- (c) nie je nevyhnutný
- (d) je zbytočný
- (e) nie je nevyhnutný, ale potrebné triedy V/V systému musia byť skopirované do daného programu

8. (2 b) V jazyku C++ ste implementovali sčítavanie a odčítavanie farieb. Na zjednodušenie zápisu týchto operácií vhodné je použiť

- (a) preťaženie operátorov
- (b) virtuálne funkcie
- (c) virtuálne operátory
- (d) preťaženie funkcií
- (e) statické funkcie

9. (2 b) Synchronizácia statickej metódy

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

10. (2 b) Okno v rámci Swing sa dá vytvoriť zavolaním

- (a) konšuktora triedy `SwingWindow`
- (b) metódy `newWindow()` triedy `JFrame`
- (c) statickej metódy `EventQueue.newWindow()`
- (d) statickej metódy `SwingUtils.newWindow()`
- (e) konšuktora triedy `JFrame`

11. (2 b) Jeden z rozdielov medzi abstraktnou triedou a rozhraním v Jave je ten, že

- (a) je možné vytvárať inštancie rozhraní, ale nie aj abstraktných tried
- (b) abstraktná trieda môže dediť od rozhrania, ale nie naopak
- (c) rozhranie môže dediť od abstraktnej triedy, ale nie naopak
- (d) je možné vytvárať inštancie abstraktných tried, ale nie aj rozhraní
- (e) konkrétne triedy môžu dediť od abstraktnej triedy, ale nie od rozhrania

12. (2 b) V diagrame sekvencií jazyka UML horizontálne šípky s plnou čiarou označujú

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

13. (3 b) Čím treba nahradiť označené časti kódu z obr. 1, aby metóda add() pripájala ďalší prvok so zadaným údajom data k aktuálnemu prvku pre rôzne typy údajov?

- (a) ****1****: ****2****: d ****3****: El<d>(d)
- (b) ****1****: <T> ****2****: <T> d ****3****: El(<T> d)
- (c) ****1****: <T> ****2****: T d ****3****: El<T>(d)
- (d) ****1****: <data> ****2****: data d ****3****: El<data>(d)
- (e) ****1****: ****2****: T d ****3****: El(d)

```
public class El<T> {  
    private T data;  
    private El**1** next;  
    public El(**2**) { data = d; }  
    public void add(**2**) { next = new **3**; }  
}
```

Obr. 1: Kód pre otázku 13.

14. (3 b) Po spustení kódu z obr. 2 vypíše sa

- (a) abcadb
- (b) abcd
- (c) bbcd
- (d) bbcbdb
- (e) abcbdb

```
interface I {  
    void op();  
}  
class A implements I {  
    public void op() { System.out.print("a"); }  
}  
class B extends A {  
    public void op() { System.out.print("b"); }  
}  
class C implements I {  
    protected A a;  
    public C(A a) { this.a = a; }  
    public void op() {  
        System.out.print("c");  
        a.op();  
    }  
}  
class D extends C {  
    public D(A a) { super(a); }  
    public void op() {  
        System.out.print("d");  
        a.op();  
    }  
}  
class M {  
    public void opi(I... i) {  
        for (I x : i)  
            x.op();  
    }  
    public static void main(String[] args) {  
        A o1 = new B();  
        B o2 = new B();  
        C o3 = new C(o1);  
        D o4 = new D(o2);  
        new M().opi(o1, o2, o3, o4);  
    }  
}
```

Obr. 2: Kód pre otázku 14.

15. (3 b) Pre architektonický vzor Model–View–Controller nie je významný ani jeden z návrhových vzorov

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

16. (3 b) V programe je každý druh geometrického útvaru reprezentovaný triedou. Rozmery útvarov sú reprezentované atribútmi a ich čítanie a zmena zabezpečené prostredníctvom metód. Je vhodné odvodiť triedu, ktorá reprezentuje štvorec, od triedy, ktorá reprezentuje obdĺžnik?

- (a) nie, lebo to porušuje princíp otvorenosti a uzavretosti
- (b) nie, lebo to porušuje Liskovej princíp substitúcie
- (c) áno, ale len ak sú príslušné atribúty **private** a metódy **public**
- (d) áno
- (e) áno, ale prekonaním metód musí byť zabezpečené, aby sa pri štvorcí výška menila zároveň so šírkou

17. (3 b) Na sledovanie pohybu kurzora myši nad tlačidlom v rámci Swing by ste

- (a) v samostatnej niti sledovali v slučke, či sa kurzor nenachádza nad tlačidlom
- (b) vytvorili niť udalostí a registrovali ju pre tlačidlo
- (c) vytvorili prijímač udalosti a registrovali ho pre tlačidlo
- (d) v niti na odosielanie udalosti sledovali v slučke, či sa kurzor nenachádza nad tlačidlom
- (e) registrovali tlačidlo v niti na odosielanie udalosti

18. (10 b) V informačnom systéme finančnej inštitúcie sa okrem iného vedie evidencia o bežných účtoch a úveroch klienta. Systém poskytuje možnosť výpisu stavu všetkých účtov a úverov pre zvoleného klienta vzhľadom na rôzne kritériá ako sú celková dlžná suma (pre účet sa berie reálny debet, a ak nie je, tak 0) a schválený úver (pre účet sa berie povolený debet). Dá sa očakávať, že v budúcnosti bude potrebné podporiť ďalšie kritériá pre výpis stavu.

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 a

3 e

4 a

5 b

6 e

7 c

8 a

9 b

10 e

11 b

12 a

13 c

14 d

15 e

16 b

17 c

45