

Priezvisko:

1 b	
2 b	
3 b	

Meno:

	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 b)** Modifikátory prístupu v Jave slúžia na
 - (a) ochranu údajov, pričom **private** je najvyššia ochrana
 - (b) obmedzenie viditeľnosti prvkov tried, pričom **private** je najvyššie obmedzenie
 - (c) ochranu údajov, pričom **protected** je najvyššia ochrana
 - (d) ochranu prvkov triedy pred úpravou
 - (e) obmedzenie viditeľnosti prvkov tried, pričom **protected** je najvyššie obmedzenie
- (1 b)** Prúd údajov (stream) v Java API sa otvára
 - (a) jeho konštruktorom
 - (b) príkazom `System.open()`
 - (c) jeho prvým použitím
 - (d) jeho metódou `open()`
 - (e) príkazom `IOStream.open()`
- (1 b)** Z hľadiska objektovo-orientovaného prístupu rozsi-ahle používanie statických metód
 - (a) nie je žiaduce, lebo sa nededia
 - (b) nie je žiaduce, lebo nepodporujú zapuzdrenie
 - (c) je žiaduce, lebo sa vykonajú rýchlejšie
 - (d) nie je žiaduce, lebo sa nedajú prekonať
 - (e) je žiaduce, lebo umožňujú polymorfizmus
- (1 b)** Dedenie triedy od rozhrania v Jave predstavuje de-denie
 - (a) štruktúry
 - (b) implementácie
 - (c) správania
 - (d) rozšírenia
 - (e) agregácie
- (2 b)** Ak je potrebné pracovať s V/V systémom Javy, príkaz `import java.io.*;`
 - (a) je zbytočný

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

6. (2 b) V jazyku C++ ste implementovali sčítavanie a odčítavanie farieb. Na zjednodušenie zápisu týchto operácií vhodné je použiť

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

7. (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 vytvára inštanciu rozhrania
- (c) trieda volá všetky metódy rozhrania
- (d) trieda ovplyvňuje rozhranie
- (e) trieda volá niektorú z metód rozhrania

8. (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`

9. (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

10. (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

11. (2 b) Výraz `call(abc*(..))` v jazyku AspectJ znamená

- (a) zachytenie volania všetkých metód, ktorých 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) zachytenie prvého volania metódy, ktorej názov začína na `abc`
- (e) vyvolanie prvej metódy, ktorej názov začína na `abc`

12. (2 b) Synchronizácia statickej metódy

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

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

14. (3 b) Pre architektonický vzor Model–View–Controller nie je významný ani jeden z návrhových vzorov

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

15. (3 b) Po spustení kódu z obr. 1 vypíše sa

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

16. (3 b) Čím treba nahradiť označené časti kódu z obr. 2, 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****: ****2****: `T d` ****3****: `El(d)`
- (c) ****1****: `<T>` ****2****: `<T> d` ****3****: `El(<T> d)`
- (d) ****1****: `<T>` ****2****: `T d` ****3****: `El<T>(d)`
- (e) ****1****: `<data>` ****2****: `data d` ****3****: `El<data>(d)`

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

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

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

```
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. 1: Kód pre otázku 15.

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.

```
public class El<T> {
    private T data;
    private El<T> next;
    public El(T data) { this.data = data; }
    public void add(T data) { next = new El<T>(data); }
}
```

Obr. 2: Kód pre otázku 16.

1 b

2 a

3 d

4 c

5 e

6 c

7 e

8 e

9 a

10 b

11 a

12 c

13 d

14 c

15 b

16 d

17 a

45