

(vyplňte tlačенým písmom)

Priezvisko:

Meno:

1 b	
2 b	
3 b	

1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	

Skúška trvá 70 minút.

Odpovede na otázky 1–11 vpíšte do tabuľky. Pri týchto otázkach sa hodnotia len odpovede v tabuľke (bez postupu). Odpoveď musí byť jednoznačná a čitateľná, inak má hodnotu 0 bodov.

V otázkach s ponúknutými odpoveďami je len jedna možnosť správna – do tabuľky píšete len písmeno, ktorým je označená odpoveď, ktorú vyberáte.

Odpoveď na otázku 12 píšete výlučne na list, na ktorom sa nachádza jej znenie.

Poškodený list nebude uznaný.

1. (2 b) Čo sa vypíše po spustení nasledujúceho programu v Jave?

```
class X extends Exception {}
```

```
class A {
    public void m(int i) throws X {
        if (i > 0)
            System.out.print("M");
        else
            throw new X();
    }
    public void op(int i) throws X {
        try {
            m(i);
        } catch (X e) {
            throw e;
        }
    }
    public static void main(String[] args) {
        try {
            new A().op(-1);
            new A().op(1);
        } catch (X e) {
            System.out.print("X");
        } finally {
            System.out.print("F");
        }
        System.out.print("A");
    }
}
```

2. (1 b) Dedeniu medzi triedami v Jave je v jazyku C++ najpodobnejšie dedenie typu

- (a) friend
- (b) protected
- (c) public
- (d) private
- (e) virtual

3. (1 b) V jazyku C++ možno vytvoriť nový objekt

- (a) výlučne definovaním premennej typu danej triedy
- (b) pomocou operátora **new** ako v Jave alebo definovaním premennej typu danej triedy
- (c) iba z tried, ktoré neobsahujú virtuálne funkcie
- (d) výlučne pomocou operátora **new** ako v Jave
- (e) nie z triedy, ale iba zo šablóny

4. (3 b) Trieda, ktorá reprezentuje špeciálnu hru, je odvodená od triedy, ktorá reprezentuje všeobecnú hru. Metóda na pridanie hráča je v špeciálnej hre prekonaná tak, že povoľuje pridanie hráča s akýmkoľvek počtom bodov, kým vo všeobecnej hre počet bodov musí byť väčší ako nula (možné je aj záporné hodnotenie). Týmto sa predpoklady a dôsledky týchto metód zoslabujú, zosilňujú alebo sa nemenia? Je týmto dodržaný Liskovej princíp substitúcie (LSP)?

Odpovedzte vo forme: *predpoklady / dôsledky / LSP*. Položky *predpoklady* a *dôsledky* nahraďte jednou z možností *zoslabujú sa*, *zosilňujú sa* alebo *nemenia sa*. Položku *LSP* nahraďte jednou z možností *dodržaný* alebo *nedodržaný*.

5. (3 b) Súčasťou grafického používateľského rozhrania počítačovej hry je aj tlačidlo `t` (objekt typu `JButton`), v súvislosti s ktorým sa v hre vyskytuje nasledujúci kód v Jave:

```
class GameWindow extends JFrame {
    Player player;
    ...
    class DecreaseEnergy implements ActionListener() {
        public void actionPerformed(ActionEvent e) {
            if (player.hasShield())
                player.setEnergy(player.getEnergy() - 1);
            else
                player.setEnergy(player.getEnergy() - 2);
        }
    }
    ...
    t.addActionListener(new DecreaseEnergy());
    ...
}
```

Z hľadiska flexibility objektovo-orientovaného návrhu by bolo najdôležitejšie

- (a) pomenovať tlačidlo `t` výstižnejšie
- (b) odvodiť zodpovedajúce triedy z triedy `Player` a použiť polymorfizmus namiesto príkazu **if**
- (c) aplikovať modifikátor prístupu **private** na atribút `player`
- (d) zmeniť triedu `DecreaseEnergy` na anonymnú alebo použiť lambda výraz na priradenie metódy `actionPerformed()` k tlačidlu `t`
- (e) vyčleniť kód v metóde `actionPerformed()` do implementácie logiky hry

6. (1 b) V jazyku C# (menný priestor – angl. namespace)

- (a) v jednom adresári možno definovať iba jeden menný priestor
- (b) možno definovať viac menných priestorov v jednom súbore a prvky z rôznych súborov môžu byť súčasťou jedného menného priestoru
- (c) možno definovať viac menných priestorov v jednom súbore, ale prvky z rôznych súborov nemôžu byť súčasťou jedného menného priestoru
- (d) prvky z rôznych súborov môžu byť súčasťou jedného menného priestoru, ale nemožno definovať viac menných priestorov v jednom súbore
- (e) prvky z rôznych súborov môžu byť súčasťou jedného menného priestoru, ak sa nachádzajú v spoločnom adresári

7. (2 b) Súčasťou riešenia problému je aj určitá funkčná časť, ktorá v rôznych prípadoch jeho uplatnenia môže byť iná. Ktorý návrhový vzor by ste použili?

- (a) Observer
- (b) MVC
- (c) Composite
- (d) Visitor
- (e) Strategy

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

```
class C extends Thread {
    private static int x = 'a';
    private static int y = 'b';
    public void ab() {
        x = 'a';
        y = 'b';
    }
    public void ba() {
        x = 'b';
        y = 'a';
    }
    public void t() {
        if (x == y)
            e();
    }
    public void e() {
        System.out.println("=");
    }
}

class M {
    public static void main(String[] args) {
        C o = new C();

        new Thread() {
            public void run(){
                for (int i = 1; i < 100000; i++)
                    o.ab();
            }
        }.start();

        new Thread() {
            public void run(){
                for (int i = 1; i < 100000; i++)
                    o.ba();
            }
        }.start();

        new Thread() {
            public void run(){
                for (int i = 1; i < 100000; i++)
                    o.t();
            }
        }.start();
    }
}
```

Ktoré metódy musia byť synchronizované, aby sa nikdy nevy-
písal znak =?

9. (1 b) Na to, aby v jazyku AspectJ bolo možné vykonať kód pred metódou

- (a) pred jej názvom treba použiť kľúčové slovo before
- (b) nie je potrebné do nej zasahovať
- (c) treba ju označiť anotáciou @before
- (d) treba ju premiestniť do aspektu
- (e) jej názvu musí predchádzať predponá before

10. (1 b) Výhodou použitia aspektov v jazyku AspectJ je, že

- (a) lepšie sústreďujú kód podľa záležitostí
- (b) automaticky opravujú chyby v programe
- (c) zvyšujú bezpečnosť
- (d) zrychlujú vykonávanie programu
- (e) umožňujú generovať lepšiu dokumentáciu než Javadoc

11. (3 b) Čo sa vypíše po spustení nasledujúceho programu v Jave?

```
interface Y {
    void a();
}

abstract class J implements Y {
    public void a() {
        System.out.print("J");
    }
}

class K extends J {
    public void a(int i) {
        System.out.print("K");
    }
}

class L extends K {
    public void a() {
        super.a();
        System.out.print("L");
    }
}

class A {
    static void s() {
        Y o1 = new L();
        L o2 = new L();
        K o3 = new K();
        J o4 = new K();
        J o5 = new L();

        o1.a();
        o2.a();
        ((J)o3).a();
        o4.a();
        ((L)o5).a();
    }
}

class B extends A {
    static void s() {
        Y o1 = new L();
        K o2 = new K();
        Y o3 = new K();
        J o4 = new K();
        L o5 = new L();

        o1.a();
        ((J)o2).a();
        o3.a();
        ((K)o4).a();
        o5.a();
    }
}

class M {
    public static void main(String[] args) {
        A o = new B();
        o.s();
    }
}
```

(vypláte tlačným písmom)

Priezvisko:

Meno:

12. (10 b) V informačnom systéme fakulty sa okrem iného vedie evidencia o predmetoch. Jestvujú tri druhy predmetov: prednášané predmety s cvičeniami, prednášané predmety bez cvičení a projektové predmety. Každý druh predmetu je implementovaný vlastnou triedou. Pre každý predmet sa eviduje názov, anotácia a počet kreditov, ale niektoré údaje sú špecifické podľa druhu predmetu. Prednášané predmety majú prednášateľa a garanta, kým projektové predmety nemajú prednášateľa, ale iba garanta. Pri prednášaných predmetoch sa eviduje počet hodín prednášok. Pri prednášaných predmetoch s cvičeniami sa k tomu eviduje aj počet hodín cvičení a zoznam asistentov.

Systém poskytuje možnosť výpisu súhrnných informácií o predmetoch v úplnej a skrátenej forme. Pri úplnej forme sa vypíšu všetky informácie, pričom sa nevypisujú položky irelevantné vzhľadom na druh daného predmetu. Pri skrátenej forme sa vypíše iba názov predmetu a meno prednášateľa a/a-lebo garanta. V ďalších verziách možno vznikne potreba za novými formami výpisov.

Navrhните a implementujte v Jave zodpovedajúce objektovo-orientované riešenie. Základný návrh predložte vo forme diagramu tried s najvýznamnejšími vzťahmi, operáciami a atribútmi.

Aplikujte adekvátne mechanizmy objektovo-orientovaného programovania. Ak je to vhodné, aplikujte niektorý z návrhových vzorov. Identifikujte explicitne, čím sú modelované a implementované roly aplikovaného vzoru a vysvetlite, čo sa týmto vzorom dosahuje.

Sústredte sa na aplikačnú logiku – GUI nie je predmetom otázky. Poskytnite príklad použitia, v ktorom vytvoríte príslušné objekty a spustíte ich interakciu.

30

1 XFA

2 c

3 b

4 zoslabujú sa / nemenia sa / dodržaný

5 e

6 b

7 e

8 C.ab(), C.ba() a C.t()

9 b

10 a

11 JLJLJJL

V poslednej otázke mal byť aplikovaný vzor Visitor. Druhy predmetov by boli v role Elementu, a druhy výpisov v role Visitora.

Otázka bude hodnotená podľa nasledujúceho kľúča:

- zabezpečenie základnej funkčnosti – 4 b
- návrh vzhľadom na princíp otvorenosti a uzavretosti kódu a adekvátne použitie zapuzdrenia – 6 b

Pri hodnotení oboch častí bude zohľadnené poskytnuté vysvetlenie (vrátane príkladu použitia) a zodpovedajúci diagram tried vo výške približne 10–20% hodnotenia príslušnej časti.