

Pokud zadání nerozumíte nebo se vám zdá nejednoznačné, zeptejte se. Pište čitelně, nečitelná řešení nebudeme uznávat.

1. Odkrojujte následující program a s použitím notace z přednášky popište stav paměti v místech označených komentáři (stavy A, B a C). Předpokládejte, že typ `double` má neomezenou přesnost a že garbage collector odstraní z haldy alokované instance okamžitě poté, co přestanou být z programu dostupné.

```
class Vector {
    final double x, y;
    private static Vector testVector = null;

    Vector(double x, double y) {
        this.x = x;
        this.y = y;
        // stav B
    }

    double getX() {
        return x;
    }

    double getY() {
        return y;
    }

    static Vector testVector() {
        // stav A
        if (testVector == null) testVector = new Vector(1.0, 0.0);
        return testVector;
    }

    Vector funcionMagica() {
        double[] a = {1.0, 0.5, -1.0, -1.0};
        Vector v = this;
        for (double d : a) v = new Vector(Math.sin(Math.PI * d),
            Math.cos(Math.PI * d));
        return v;
    }

    public static void main(String[] args) {
        Vector v = Vector.testVector();
        double k = v.getY() / v.getX();
        Vector v2 = v.funcionMagica();
        // stav C
    }
}
```

OMO, 16. 4. 2013

Jméno:

2. Napište, co vypíše následující kód.

```
abstract class A {
    abstract void fa(A a);

    void fb(B b) {
        System.out.println("1");
    }

    void fc(C c) {
        System.out.println("2");
    }
}

class B extends A {
    void fa(A a) {
        a.fb(this);
    }

    void fc(C c) {
        System.out.println("3");
    }
}

class C extends A {
    void fa(A a) {
        a.fc(this);
    }

    void fb(B b) {
        System.out.println("4");
    }
}

class Main {
    public static void main(String args[]) {
        new B().fa(new B());
        new B().fa(new C());
        new C().fa(new C());
        new C().fa(new B());
    }
}
```

3. Najděte a vysvětlete chybu při překladu.

```
class A {
    static int i = 0;

    static void f() {
        System.out.println("A.f()");
    }

    void g() {
        System.out.println(h());
    }

    private String h() {
        return "A.h()";
    }
}

class B extends A {
    static void f() {
        System.out.println("B.f()");
        g();
    }

    private String h() {
        return "B.h(): " + (++i);
    }
}

class Task3 {
    public static void main(String[] args) {
        A a = new B();
        a.f();
        a.g();
    }
}
```

4. Vyznačte řádek, na kterém dojde k chybě za běhu programu, a napište na které instanci k dané chybě došlo. Svoji odpověď vysvětlete.

```
class Main {
    public static void main(String args[]) {
        A a = new A();
        B b1 = new B(a);
        B b2 = new B(b1);

        System.out.println(b2.f());
    }
}

interface I {
    int f();

    void g();
}

class A implements I {
    I a;
    int c;

    public int f() {
        return a.f() + c;
    }

    public void g() {
        this.a = new B(a);
        c = 70;
    }
}

class B extends A {
    public B(I t) {
        a = t;
        c = 30;
    }

    public int f() {
        c += 30;
        return c > 50 ? super.f() + c : c + 50;
    }
}
```

5. Třída `Table` reprezentuje tabulku, ve které může každé číslo být asociováno se žádným, jedním nebo více řetězci. Metoda `add(Integer key, String value)` přidá do tabulky asociaci klíče `key` s hodnotou `value`, metoda `remove(Integer key, String value)` odstraní z tabulky asociaci klíče `key` s hodnotou `value` a metoda `getValues(Integer key)` vrátí množinu všech hodnot asociovaných s klíčem `key`. Vaším úkolem je dopsat kód metody `add`, kód ostatních metod ani atributy modifikovat nesmíte.

```
class Table {
    private Map<Integer, Set<String>> map = new HashMap<>();

    public void remove(Integer key, String value) {
        if (map.containsKey(key)) {
            Set<String> set = map.get(key);
            set.remove(value);
            if (set.isEmpty()) map.remove(key);
        }
    }

    public Set<String> getValues(Integer key) {
        Set<String> set = map.get(key);
        if (set == null) set = new HashSet<>();
        return set;
    }

    public void add(Integer key, String value) {

    }
}
```