

Ćwiczenia z programowania obiektowego
17.03.2010 r.

1. Jak zmienia się diagram obiektów podczas wykonania następującego kodu?

```
01: public class Pair {
02:     private int x;
03:     private int y;
04:
05:     public Pair(int x, int y) {
06:         this.x = x;
07:         this.y = y;
08:     }
09:
10:     public static void main(String args[]) {
11:         Pair a;
12:         Pair b;
13:
14:         a = new Pair(1,2);
15:         b = new Pair(3,4);
16:
17:         Pair[] pairs;
18:
19:         pairs = new Pair[10];
20:         pairs[0] = new Pair(1,4);
21:         pairs[1] = a;
22:
23:         a = b;
24:
25:         pairs[1].x = 7;
26:         pairs[1] = new Pair(2,3);
27:         pairs[1] = null;
28:     }
29: }
```

2. Jaki jest wynik wykonania następującego kodu?

```
public class Mixer {
    public Mixer() { }
    public Mixer(Mixer m) { m1 = m; }
    Mixer m1;

    void go() { System.out.print("hi "); }

    public static void main(String[] args) {
        Mixer m2 = new Mixer();
        Mixer m3 = new Mixer(m2); m3.go();
        Mixer m4 = m3.m1;
        Mixer m5 = m2.m1;
m5.go();
    }
}
```

3. Dany jest następujący kod.

```
String strA = new String("Roasted ");
strA        = new String("Toasted ");
strA        = new String("Fried ");
strA        = new String("Baked ");
strA        = new String("Beans ");
```

Ile (w sumie) obiektów zostanie utworzonych? Ile obiektów jest dostępnych (osiągalnych) po wykonaniu ostatniego wyrażenia?

4. Dany jest następujący kod.

```
public class CardBoard {
    int story = 5;
    CardBoard go() {
        return new CardBoard();
    }
    public static void main(String[] args) {
        CardBoard c1 = new CardBoard();
        CardBoard c2 = new CardBoard();
        CardBoard c3 = c1.go();
        c1 = null;
        // koniec
    }
}
```

Ile obiektów nadaje się do odśmiecenia w chwili, gdy osiągamy //koniec?

5. Jaki jest wynik wykonania następującego kodu?

```
class Fizz {
    int x = 5;
    public static void main(String[] args) {
        Fizz f1 = new Fizz();
        Fizz f2 = new Fizz();
        Fizz f3 = FizzSwitch(f1,f2);
        System.out.println((f1 == f3) + " " + (f1.x == f3.x) +
            " " + f1.x);
    }
    static Fizz FizzSwitch(Fizz x, Fizz y) {
        Fizz z = x;
        z.x = 6;
        return z;
    }
}
```

6. Jaki jest wynik działania następującego programu?

```
01: class Cup {
02:     String type;
03:     Cup(String type) {
04:         this.type = type;
05:     }
06: }
07:
08: public class Test {
09:     static void fillWithCoffee(Cup cup) {
10:         cup.type = "coffee";
11:     }
12:
13:     static void makeSure(int certainty) {
14:         certainty = 100;
15:     }
16:
17:     public static void main(String[] args) {
18:         Cup cup = new Cup("tea");
19:         int certainty = 50;
20:
21:         fillWithCoffee(cup);
22:         makeSure(certainty);
23:
24:         System.out.println("I'm " + certainty + "% certain" +
25:             " the cup is filled with " + cup.type + ".");
26:     }
27: }
```

7. Dany jest następujący kod.

```
01: void test() {
02:     Car car = new Car("Red");
03:     helper(car);
04: }
05:
06: void helper(Car c) {
07:     c = new Car("Black");
08:     anotherHelper(c);
09: }
10:
11: void anotherHelper(Car c) {
12:     c = new Car("White");
13: }
```

Na końcu metody `test()`, zmienna `car` przechowuje referencję do jednego z obiektów klasy `Car`. Do którego?

8. Co zostanie wypisane w linii 10?

```
01: public class FollowMe{
02:     int a;
03:     int b;
04:
05:     public void funct(){
06:         a=1;
07:         b=1;
08:         int[] c={2};
09:         changeMe(b,c);
10:         System.out.println(a+" "+b+" "+c[0]+" ");
11:     }
12:
13:     public void changeMe(int b,int[] c){
14:         a=2;
15:         b=2;
16:         c[0]=1;
17:     }
18:
19:     public static void main(String [] args){
20:         FollowMe obj=new FollowMe();
21:         obj.funct();
22:     }
23: }
```

9. Dana jest klasa Point:

```
public class Point {
    private int x;
    private int y;

    public Point() {
        this.x = 0;
        this.y = 0;
    }

    public Point(int x, int y) {
        this.x = x;
        this.y = y;
    }

    public boolean equals(Point other) {
        if (other == null) {
            return false;
        }
        return ((x == other.x) && (y == other.y));
    }

    public void move(int x, int y) {
        this.x = x;
        this.y = y;
    }
}
```

Jaki jest wynik działania następujących fragmentów kodu?

- (a) `Point pointA = new Point();`
`Point pointB = new Point(0, 0);`
`System.out.println(pointA == pointB);`
- (b) `Point pointA = new Point(21, 17);`
`Point pointB = pointA;`
`System.out.println(pointA == pointB);`
- (c) `Point pointA = new Point(21, 17);`
`Point pointB = new Point(21, 17);`
`System.out.println(pointA == pointB);`
- (d) `Point pointA = new Point(21, 17);`
`Point pointB = new Point(21, 17);`
`System.out.println(pointA.equals(pointB));`
- (e) `Point pointA = new Point(21, 17);`
`Point pointB = new Point(21, 17);`
`pointA.move(8, 12);`
`System.out.println(pointA.equals(pointB));`