

Ćwiczenia z programowania obiektowego
10.03.2010 r.

1. Które z poniższych napisów są poprawnymi identyfikatorami zmiennych w Javie?

- (a) 2bad
- (b) zero
- (c) _tmp
- (d) theLastValueButOne
- (e) \$sign
- (f) year2000
- (g) yearly_income

Które z nich są zgodne z konwencjami nazewniczymi Javy?

2. Jaki będzie wynik działania następującego fragmentu programu?

```
...
int a = 7;

if (a > 5) {
    int b = 6*a;
} else {
    int b = 6-a;
}

System.out.println(b);
...
```

3. Jaki będzie wynik działania następującego fragmentu programu?

```
int x;
...
if (x = 7) {
    System.out.println("Seven!");
}
```

4. Jaki będzie wynik działania tego programu? Jaka będzie wartość zmiennej b w punkcie 1?

```
boolean b = false;

if (b = true) {
    System.out.println("1");
} else {
    System.out.println("2");
}
1:
```

5. Które z następujących deklaracji zmiennych są poprawne?

- (a) `float x = 3.2;`
- (b) `boolean a = (7 = 3);`
- (c) `boolean a = (x = 3);`
- (d) `char z = "a";`
- (e) `String b = 'b';`
- (f) `float y = 3.14f`

6. Który z poniższych fragmentów kodu deklaruje tablicę liczb całkowitych o nazwie `img`?

- (a) `int img;`
- (b) `int[] img;`
- (c) `new int img[];`
- (d) `int img = int[];`

7. Czy w czasie pisania programu programista musi wiedzieć, jakiej wielkości będzie tablica?

8. Jaki jest najmniejszy indeks w tablicy rozmiaru 1000? A w tablicy rozmiaru 10008?

9. Jaki jest największy indeks w tablicy rozmiaru 1000? A w tablicy rozmiaru 10008?

10. Dana jest następująca tablica:

```
byte[] data = { 12, 34, 9, 0, -62, 88 };
```

Ile wynosi wartość `data.length`?

11. Jaki będzie wynik działania następującego fragmentu kodu?

```
int[] ar = {2, 4, 6, 8 };  
System.out.println( ar[0] + " " + ar[1] );
```

12. Jaki będzie wynik działania następującego fragmentu kodu?

```
int[] y = new int[10];  
  
y[0] = 34;  
y[1] = 88;  
  
System.out.println( y[0] + " " + y[1] + " " + y[5] );
```

13. Jaki będzie wynik działania następującego programu?

```
int[] zip = new int[5];  
  
zip[0] = 7;  
zip[1] = 4;  
zip[2] = 3;  
zip[3] = 1;  
zip[4] = 9;  
  
System.out.println( zip[ 4 ]/zip[ 1 ] );
```

14. Uzupełnij luki w podanym fragmencie kodu tak, aby elementy tablicy zostały wypisane w kolejności od ostatniego do pierwszego elementu.

```
int[] egArray = { 2, 4, 6, 8, 10, 1, 3, 5, 7, 9 };

for ( int index= _____ ; _____ ; _____ )
    System.out.print( egArray[ index ] + " " );
```

15. Dana jest tablica.

```
double[][] values =
    { {1.2, 9.0, 3.2},
      {9.2, 0.5, 1.5, -1.2},
      {7.3, 7.9, 4.8} } ;
```

Jaką wartość ma values[2][1]?

16. Dana jest tablica.

```
double[][] values =
    { {1.2, 9.0, 3.2},
      {9.2, 0.5, 1.5, -1.2},
      {7.3, 7.9, 4.8} } ;
```

Jaką wartość ma values[3][0]?

17. Dana jest tablica.

```
double[][] things =
    { {1.2, 9.0},
      {9.2, 0.5, 0.0},
      {7.3, 7.9, 1.2, 3.9} } ;
```

Ile wynosi things.length? Ile wynosi things[2].length ?

18. Napisz (na dwa różne sposoby) kod, który utworzy następującą tablicę.

```
12      -9      8
7        14
-32     -1      0
```

19. Dana jest deklaracja.

```
long[][] stuff ;
```

Która z poniższych instrukcji tworzy tablicę o 5 wierszach długości 7 kolumn każda.

- (a) stuff = new stuff[5][7] ;
- (b) stuff = new long[5][7] ;
- (c) stuff = long[5][7] ;
- (d) stuff = long[7][5] ;

20. Dana jest następująca tablica.

```
int[] [] items =
    { {0, 1, 3, 4},
      {4, 3, 99, 0, 7 },
      {3, 2} } ;
```

Które z poniższych wyrażeń zamienia 99 na 77?

- (a) `items[1][2] = 77;`
- (b) `items[2][1] = 77;`
- (c) `items[99] = 77;`
- (d) `items[2][3] = 77;`

21. Które z poniższych wyrażeń tworzy i przypisuje na zmienną `array` dwuwymiarową tablicę o 7 wierszach, ale nie tworzy jeszcze wierszy?

- (a) `int[] [] array = new int[7] [];`
- (b) `int[] [] array = new int[7];`
- (c) `int[] [] array = new int[] [7];`
- (d) `int[] array[7] = new int[];`

22. Dana jest następująca tablica.

```
int[] [] items =
    { {0, 1, 3, 4},
      {4, 3, 99, 0, 7 },
      {3, 2} } ;
```

Które z poniższych wyrażeń zastępuje w tablicy `items` wiersz 0 przez zupełnie nowy wiersz?

- (a) `items[0][0] = 8;`
`items[0][1] = 12;`
`items[0][2] = 6;`
- (b) `items[0] = { 8, 12, 6 };`
- (c) `items[0] = new { 8, 12, 6 };`
- (d) `int[] temp = { 8, 12, 6 };`
`items[0] = temp;`

23. Jaki jest wynik działania programu?

```
public class OF {
    public static void main(String [] args) {
        int r = 1;
        int x = 11;
        r += x;
        if ((x > 4) && (x < 10)) {
            r += 2 * x;
        } else {
            r += 3 * x;
        }
        System.out.println("OF(11) is: " + r);
    }
}
```

24. Jaki jest wynik działania programu?

```
class Foozit {
    public static void main(String[] args) {
        int x = 0;
        int y = 0;
        for(int z = 0; z < 5; z++)
            if((++x > 2) || (++y > 2))
                x++;
        System.out.println(x + " " + y);
    }
}
```

25. Dany jest program:

```
class Fork {
    public static void main(String[] args) {
        if(args.length == 1 | args[1].equals("test")) {
            System.out.println("test case");
        } else {
            System.out.println("production " + args[0]);
        }
    }
}
```

Jaki będzie wynik wykonania następującego polecenia?

```
java Fork live2
```

26. Jaki będzie wynik działania następującego programu?

```
class Hexy {
    public static void main(String[] args) {
        int i = 42;
        String s = (i<40) ? "life" : (i>50) ? "universe" : "everything";
        System.out.println(s);
    }
}
```

27. Jaki będzie wynik działania programu?

```
class Swill {
    public static void main(String[] args) {
        String s = "-";
        int x = 2;
        switch(x) {
            case 1: s += "e";
            case 2: s += "c";
            case 3: s += "m";
            default: s += "X";
            case 4: s += "p";
        }
        System.out.println(s);
    }
}
```

28. Jaki będzie wynik działania programu?

```
class Swill2 {
    public static void main(String[] args) {
        String s = "-";
        long x = 2;
        switch(x) {
            case 1: s += "e";
            case 2: s += "c";
            case 3: s += "m";
            default: s += "X";
            case 4: s += "p";
        }
        System.out.println(s);
    }
}
```

29. Dane są dwie tablice liczb całkowitych posortowane rosnąco. Traktujemy je jako reprezentacje dwóch zbiorów. Napisać fragment kodu w Javie, który wypisze na wyjście część wspólną obu zbiorów.
30. Dane są dwie tablice liczb całkowitych. Tablice reprezentują dwa wielomiany. Napisz fragment kodu w Javie, który policzy sumę tych wielomianów. Zadbaj o to, by rozmiar tablicy wynikowej był optymalny.