ćw.1.

1. Zaznacz poprawne odpowiedzi w poniższych pytaniach

(1) [1 pkt.] Uruchamialny program *Java* musi składać się z:
- a. publicznej klasy `Main`
- b. klasy zawierającej statyczną metodę `main`
- c. zbioru metod
- d. deklaracji pól

(2) [1 pkt.] Które z poniższych zmiennych zostały zainicjowane domyślnymi wartościami:
- a. `int wrt = 0;`
- b. `boolean logi = false;`
- c. `Object obj = new Object();`
- d. `String str = null;`

(3) [1 pkt.] Które z poniższych deklaracji tablic są poprawne:
- a. `int tab[] = {};`
- b. `Object objTab[] = new String[5];`
- c. `boolean tabB[][] = false, true, true;`
- d. `String strTab[] = null;`

(4) [1 pkt.] Deklarując klasę `A` należy pamiętać że:
- a. w każdej klasie zostanie umieszczony domyślny konstruktor pusty
- b. klasa będzie zawierała metodę `equals` porównującą stany wszystkich pól tej klasy
- c. metoda `toString` będzie wyświetlać informacje o położeniu obiektu tej klasy
- d. fabrykacja obiektu tej klasy może spowodować `OutOfMemoryException`

(6) [1 pkt.] Które z właściwości zmiennej `this` są prawdziwe:
- a. metody statyczne nie mogą odwoływać się do zmiennej `this`
- b. każdy obiekt posiada zmienną `this`
- c. `this` powstaje podczas tworzenia obiektu
- d. użycie `this` w konstruktorze jest niemożliwe, gdyż obiekt jeszcze nie istnieje

(7) [1 pkt.] Aby podnieść wyjątek w języku *Java* należy:
- a. użyć klauzuli `try - catch`
- b. wykorzystać słowo kluczowe `throw`
- c. umieścić w nagłówku metody `throws`
- d. utworzyć klasę dziedziczącą po `Exception`

(1) [1 pkt.] W wyniku kompilacji programu zapisanego w języku Java powstaje:
- a. program interpretowany przez maszynę wirtualną
- b. `java`
- c. `html`
- d. plik z rozszerzeniem `.class`

(2) [1 pkt.] Aplikacja *Java* rozpoczyna swoje wykonanie od wywołania funkcji:
- a. `public static main(String[] args)`
- b. `static void main(String[] args)`
- c. `public static void main(String[] args)`
- d. `public static void main()`

(3) [1 pkt.] Które z poniższych zmiennych zostały zainicjowane wartościami literału:
- a. `int a = 'c';`
- b. `boolean b = 5 > 3 ? true : false;`
- c. `double pi = 3.14;`
- d. `long wrt = 5 * 5;`

(4) [1 pkt.] Zakładając że: `int i; float f; char c; boolean b;` które z poniższych przypisań nie wymagają zastosowania konwersji:
- a. `i = f;`
- b. `i = c;`
- c. `b = i;`
- d. `b = f;`

(5) [1 pkt.] Instrukcja warunkowa *if* wykona:
- a. instrukcję następującą po orzeczeniu, jeśli wartość logiczna tego orzeczenia jest *true*
- b. instrukcje w sekcji *else*, jeżeli wartość logiczna tego orzeczenia jest *true*
- c. instrukcje w obu sekcjach, jeżeli orzeczenie będzie miało wartość logiczną *unknown*
- d. żadnej z instrukcji, jeżeli wynikiem operacji będzie wartość liczbowa

(6) [1 pkt.] Które z poniższych pętli wykonają swoje ciało przynajmniej jeden raz:
- a. *while*
- b. *for*
- c. *do - while*
- d. żadna z powyższych

(7) [1 pkt.] Które z poniższych twierdzeń są prawdziwe w języku Java:

a. zmienna tablicowa zawsze jest zainicjowana

b. w tablicach wielowymiarowych ilość elementów w wierszach zawsze jest taka sama

○ odwołanie się do nieistniejącego elementu tablicy, skutkuje błędem

d. elementy tablic są indeksowane od wartości 1

(8) [1 pkt.] Które z poniższych wyrażeń regularnych opiszą datę w formacie 12-Sty-2011

a. `\d\d-\p{L}-\d\d\d\d`

b. `\d{1-2}-[A-Z][a-z]{1-2}-\d{4}`

○ `\d{1-2}-[A-Za-z]{1-3}-\d{4}`

d. `\d\d-\D{3}-\d{4}`

(9) [1 pkt]

```
public void fun(int[] tab){
    for(int i = 1; i < tab.length; i++){
        int val = tab[i];
        int j = i;
        while( j > 0 && tab[j - 1] > val){
            tab[j] = tab[j-1];
            j--;
        }
        tab[j] = val;
    }
}
```

Powyższy algorytm reprezentuje:

a. selectionsort (sortowanie przez wybieranie)

b. listę

c. stos

○ insertionsort (sortowanie przez wstawianie)

ćw.2.

II. [10 pkt.] Dany jest poniższy, poprawnie kompilujący się program. Wpisz w kratki wynik działania programu, zachowując kolejność symboli wyświetlanych na konsoli

```
1  public
2    class Main{
3
4    public static void main(String[] args){
5      int mat[][] = {
6        { 1, 2, 3, 4},
7        { 5, 6, 7, 8},
8        { 9,10,11,12}
9      };
10
11     try{
12       fun( mat, 0, 0);
13     }catch(Exception ex){
14       System.out.println(ex);
15     }
16
17     A a = new A(15);
18     new A(10);
19     new A(5);
20
21     B.make().show();
22   }
23
24   public static void fun(int[] tab, int i, int j){
25     for(int k = 0; k < i; k++){
26       System.out.print("["+tab[k]+"] ");
27     }
28     System.out.println();
29   }
30
31   public static int fun(int[][] tab, int i, int j){
32     if(i >= 0 && j >= 0){
33       System.out.println("H"+tab[i][j]);
34       return tab[i][j] + fun(tab, i-1, j-1);
35     }else
36       return 0;
37   }
38 }
39
40 class A{
41   int wrtl;
42   A next;
43
44   private A(){
45     System.out.println("G"+wrtl);
46     B.make().add(this);
47   }
48
49   public A(int wrtl){
50     this();
51     this.wrtl = wrtl;
52     System.out.println("F"+wrtl);
53   }
54
55   public A(A objA){
56     this(objA.wrtl-1);
57     System.out.println("E"+wrtl);
58   }
59
60   public String toString(){
61     return "D"+wrtl;
62   }
63 }
```

```
64
65 class B{
66   private static B struct;
67   public static B make(){
68     System.out.println("C");
69     if(struct == null)
70       struct = new B();
71     return struct;
72   }
73
74   private A head;
75   private B(){
76     head = null;
77     System.out.println("B");
78   }
79
80   public void add(A obj){
81     if(head == null){
82       head = obj;
83     }else{
84       A tmp = head;
85       while(tmp.next != null)
86         tmp = tmp.next;
87       tmp.next = obj;
88     }
89     System.out.println("A");
90   }
91
92   public void show(){
93     A tmp = head;
94     while(tmp != null){
95       System.out.println(tmp);
96       tmp = tmp.next;
97     }
98   }
99 }
```

H1
H6
H11
java.lang.ArrayIndexOutOfBoundException:
Index 3 out of bounds for length 3

```
public
  class Main{

  public static int mid(int wrt){
    System.out.println("A");
    int tmp = 0,
      i = 0;
    while(i < 5){
      tmp += wrt;
      i++;
    }
    return tmp / wrt;
  }

  public static int mid(int[] wrt){
    System.out.println("B");
    int tmp = 0;
    for(int i=0; i < wrt.length; i++){
      tmp += wrt[i];
    }
    return tmp / wrt.length;
  }

  public static int mid(double wrt){
    System.out.println("C");
    int tmp = 0,
      i = 0;
    do{
      System.out.print("C"+i);
      i += 0.5;
    }while(wrt < 5.0);
    return tmp / (int)wrt;
  }

  public static void show(int[] tab){
    System.out.println("D");
    for(int i=0; i<3; i++){
      for(int j=0; j<4; j++){
        System.out.print(tab[(i*4)+j]+", ");
      }
      System.out.println();
    }
  }

  public static void show(char[][][] tab){
    System.out.println("E");
    for(int i=0; i<tab.length; i++){
      for(int j=0; j<tab[i].length; j++){
        System.out.print(tab[i][j]+", ");
      }
      System.out.println();
    }
  }
```

```
  public void operacja(int wrt){
    wrt += 1;
    System.out.println("F"+wrt);
  }

  public int operacja(int[] wrt){
    System.out.println("G"+wrt[0]);
    return wrt[0];
  }

  public static void fill(char[][][] myArr){
    System.out.println("H");
    for(int i=0; i<myArr.length; i++)
      for(int j=0; j<myArr[i].length; j++)
        if(i==j)
          myArr[i][j] = '0';
        else
          myArr[i][j] = '1';
  }

  public static void main(String[] args){
    int[] tab = {
      1, 2, 3, 4,
      5, 6, 7, 8,
      9, 10, 11, 12
    };

    show(tab);
    System.out.print("I");
    System.out.println(mid(tab));

    int szerokosc = ((int)Math.random()*100) + 1,
      wysokosc = 5;

    char[][][] pary = new char[wysokosc][szerokosc];
    fill(pary);
    show(pary);
  }
}
```

Zw. 3.

III. [7 pkt.] Uzupełnij ciało poniższej metody, tak aby zwracała wartość `true` gdy dostarczony argument jest zawiera znaki reprezentujące liczby lub wartość `false` w przeciwnym przypadku.

```
public boolean metoda(char[] slowo){
```

```
public boolean metoda (char [] slowo){
    for (int i=0; i< slowo. length; i++){
        if(slowo[i] >= '0' && slowo[i] <= '9'){
            return true;
        }
    }
    return false;
}
```

Zw. 4.

III. [10 pkt.] Uzupełnij ciało poniższej metody, tak aby zwracana była przekątna tablicy gdy suma elementów tej tablicy jest nieparzysta lub suma elementów środkowej kolumny w przeciwnym przypadku.

```
public int[] metoda(int[][] slowo){
```

```
public int[] metoda (int [][] slowo){
    int sum=0;
    for (int i=0; i< slowo. length; i++){
        for (int j=0; j< slowo[i]. length; i++){
            sum += slowo[i][j];
        }
    }
    int[] tab= new int[slowo. length];
    if (sum%2 != 0){
        int j=0;
        for (int i=0; i< tab length; i++){
            tab[i] = slowo[j][j];
            j++;
        }
    }else{
        int j=0;
        for (int i=0; i< tab.length; i++){
            tab[i]=slowo[tab.length/2][j];
            j++;
        }
    }
    return tab;
}
```