

Język programowania JAVA

© 2011-12 Radosław Klimek



Vincent Van GOGH: *Mężczyzna pijący filiżankę kawy*

Zadanie 6. Napisz program, który

- tworzy tablicę 30 liczb
- wstawia do tej tablicy liczby od 0 do 29
- sumuje te elementy tablicy, które są nieparzyste

Zadanie 7. Napisz program, który

- tworzy tablicę 1000 liczb
- wstawia do tej tablicy kolejne wielokrotności liczby 3 (3, 6, 9, 12, 15, ...)
- oblicza ilość liczb w tablicy podzielnych przez 5

Zadanie. Średnia arytmetyczna z podanych liczb.

Średnią arytmetyczną oblicza się dzieląc sumę wszystkich podanych liczb przez ich ilość. W tym przypadku pokażemy, jak obliczyć taką średnią dla liczb wczytanych z klawiatury.

W programie skorzystano z pętli while do wczytywania kolejnych liczb z klawiatury. Warunkiem końcowym jest podanie '-1', więc gdy chcemy zakończyć wprowadzanie liczb wpisujemy -1. Gdy podajemy liczby inne niż nasz warunek końcowy, program sumuje je (zmienna suma oraz zlicza ilość podanych liczb (zmienna i). Po wprowadzeniu warunku końcowego, obliczana jest średnia (zmienna średnia) - czyli dzielona jest suma przez ilość liczb a następnie wypisywana na ekran.

Średnia arytmetyczna – rozwiązanie

```
1 import java.util.*;
2 public class Main {
3     public static void main (String args [])
4     {
5         System.out.print ("Obliczanie średniej arytmetycznej.");
6         System.out.println ("Wpisuj kolejne liczby
7         ~~~~~~(-1 kończy pętle)");
8         float n=1;
9         int i=0;
10        float suma=0;
11        float srednia;
12        while (n!=-1)
13        {
14            System.out.print ("Podaj liczbę:");
15            Scanner wejscie=new Scanner(System.in);
16            n = wejscie.nextFloat();
17            if (n!=-1){ suma+=n; i++;}
18            else {System.out.println ("Podałś '-1' - koniec");
19                }
20        }
21        srednia=suma/i;
22        System.out.println ("Średnia z podanych " + i +
23                            " liczb to: " + srednia);
24    }
25 }
```

Zadanie. Sprawdzanie czy liczba jest parzysta.

Sprawdzanie parzystości liczb jest stosunkowo proste. Sprowadza się do użycia dzielenia modulo które zwraca resztę z dzielenia dwóch liczb.

W programie użyto funkcję `czyParzysta()` która zwraca `true/false` w zależności od prawdy. Następnie na podstawie tego co zwróci ta funkcja, wypisywany jest komunikat, że liczba jest parzysta lub nieparzysta.

Parzysta – rozwiązanie

```
1 import java.util.*;
2 public class Main {
3     public static boolean czyParzysta(int n)
4     {
5         if (n%2 == 0) return true;
6         else return false;
7     }
8     public static void main(String args[])
9     {
10        System.out.print("Podaj liczbę do sprawdzenia: ");
11        Scanner wejscie=new Scanner(System.in);
12        int n;
13        n = wejscie.nextInt();
14        if (czyParzysta(n)) System.out.println("Liczba " + n + " jest parzysta");
15        else System.out.println("Liczba " + n + " jest nieparzysta");
16    }
17 }
```

Zadanie. Równanie kwadratowe.

Obliczanie pierwiastków równania kwadratowego wiąże się z wyliczeniem delty i w zależności od niej, obliczeniem wyniku czyli pierwiastków.

W tym przykładzie, współczynniki a, b, c są typu całkowitego stąd użycie `nextInt()`. Delta obliczana jest z zwykłego wzoru na deltę. Przy wypisywaniu wyniku, obliczany jest pierwiastek kwadratowy z delty za pomocą: `Math.sqrt()` - jest to funkcja służąca do obliczania pierwiastków z liczb.

Równanie kwadratowe – rozwiązanie

```
1 import java.util.*;
2 public class Main {
3     public static void main (String args [])
4     {
5         int a;
6         int b;
7         int c;
8         double delta;
9         System.out.println ("Rownanie kwadratowe ma postac :  $ax^2+bx+c$ ");
10        Scanner wejscie=new Scanner(System.in);
11        System.out.println ("Podaj a :");
12        a = wejscie.nextInt();
13        System.out.println ("Podaj b :");
14        b = wejscie.nextInt();
15        System.out.println ("Podaj c :");
16        c = wejscie.nextInt();
17        if (a==0){
18            System.out.println ("To nie jest funkcja kwadratowa");}
19        else { delta=b*b-4*a*c;
20            if (delta < 0){
21                System.out.println ("delta < 0");}
22            if (delta == 0){
23                System.out.println ("Rownanie ma 1 pierwiastek ,  $x1 : -b/(2*a)$ ");
24            }
25            if (delta > 0){
26                System.out.println ("x1 :  $-b-Math.sqrt(delta)/2*a$ ");
27                System.out.println ("x2 :  $-b+Math.sqrt(delta)/2*a$ ");
28            }
29        }
30    }
31 }
```


Zadane. Oblicz silnię.

Obliczanie silni jest częstym problemem na początku nauki programowania. Java nie posiada wbudowanej funkcji liczącej silni, dlatego musimy ją sami napisać.

Aby obliczyć silnie z liczby wykorzystamy funkcję rekurencyjną (funkcja rekurencyjna wywołuje sama siebie).

W prezentowanym programie, liczba dla której liczymy silnię wczytywana jest z klawiatury a następnie wypisujemy wynik naszej funkcji.

Silnia – rozwiązanie

```
1 import java.util.*;
2 public class Main {
3     public static int silnia(int n){
4         if (n == 0) return 1;
5         else return n * silnia(n - 1);
6     }
7     public static void main(String args[]){
8         int n;
9         System.out.println("Podaj liczbę dla której obliczyć silnię :");
10        Scanner wejscie=new Scanner(System.in);
11        n = wejscie.nextInt();
12        System.out.println(n+"! = "+ silnia(n));
13    }
14 }
```

Zadanie. Losowanie liczb całkowitych z przedziału.

Czasami zachodzi potrzeba wypełnienia tablicy losowymi liczbami lub po prostu do przeprowadzenia pewnych obliczeń. Java posiada klasę `Random` która jest pomocna przy losowaniu.

Aby wylosować liczbę trzeba stworzyć obiekt typu `Random` a następnie dzięki metodzie `nextInt()` wylosować daną liczbę całkowitą. Jeśli chcemy wylosować liczbę z przedziału $0-N$, musimy przekazać górną wartość przedziału jako argument do wymienionej funkcji.

W programie tworzymy funkcję `losuj`, która zwraca wylosowaną liczbę, choć możemy wykonać losowanie bezpośrednio w funkcji `main`.

Obiekt `Random` posiada także inne metody losujące m.in `nextLong()`, `nextFloat()`, `nextDouble()` i inne.

Losowanie liczb całkowitych – rozwiązanie

```
1 import java.util.*;
2 public class Main {
3     public static int losuj(int n){
4         Random rand = new Random();
5         return rand.nextInt(n);
6     }
7     public static void main(String args[]){
8         System.out.print("Podaj górny przedział z jakiego wylosować liczbę [0-n] : ");
9         Scanner wejscie=new Scanner(System.in);
10        int n;
11        n = wejscie.nextInt();
12        System.out.println("Wylosowana liczba to : " + losuj(n));
13    }
14 }
```

Zadanie. Oblicz NWD.

NWD czyli największy wspólny dzielnik obliczyć można za pomocą algorytmu Euklidesa

W programie obliczającym największy wspólny dzielnik w Javie użyjemy rekurencyjnej funkcji NWD która opisuje algorytm Euklidesa.

NWD czyli Największy wspólny dzielnik –rozwiązanie

```
1 import java.util.*;
2 public class Main {
3     public static int NWD(int a, int b) {
4         if (b == 0){ return a; }
5         return NWD(b, (a % b));
6     }
7     public static void main(String args[]){
8         int a;
9         int b;
10        System.out.println("Program szukający NWD dwóch liczb.");
11        System.out.println("Podaj pierwszą liczbę :");
12        Scanner wejscie=new Scanner(System.in);
13        a = wejscie.nextInt();
14        System.out.println("Podaj drugą liczbę :");
15        b = wejscie.nextInt();
16        System.out.println("NWD (" +a+" , "+b+" ) = "+NWD(a , b ));
17    }
18 }
```

Zadanie. NWW.

NWW czyli najmniejsza wspólna wielokrotność obliczona może być za pomocą algorytmu Euklidesa który służy do obliczenia NWD.

W programie obliczającym NWW, skorzystamy z funkcji liczącej NWD czyli największy wspólny dzielnik w Javie. Liczby dla których szukany jest NWW wczytywane są z klawiatury, obliczany ich iloczyn a następnie podzielony przez ich NWD.

NWW czyli najmniejsza wspólna wielokrotność dwóch liczb – rozwiązanie

```
1 import java.util.*
2 public class Main {
3     public static int NWD(int a, int b) {
4         if (b == 0){ return a; }
5         return NWD(b, (a % b));
6     }
7     public static void main(String args []){
8         int a;
9         int b;
10        System.out.println("Program szukający NWD dwóch liczb.");
11        System.out.println("Podaj pierwszą liczbę:");
12        Scanner wejscie = new Scanner(System.in);
13        a = wejscie.nextInt();
14        System.out.println("Podaj drugą liczbę:");
15        b = wejscie.nextInt();
16        System.out.println("NWD(" +a+" , "+b+" ) = "+NWD(a , b));
17    }
18 }
```


Suma szeregu harmonicznego

Algorytm $1/(1+1/2+1/3+1/4+1/5+.....)$

- 1 Podaj dokładność eps spełniającą warunki: $0.0000001 < \text{eps} < 1$,
Dokładność oznacza, że suma szeregu podana przez program różni się od poprzedniej o wartość mniejszą niż eps (każda suma różni się, ponieważ jest sumą równej liczby elementów)
- 2 Zainicjuj sumę szeregu $\text{suma} = 0$;
- 3 Oblicz pierwszy element $\text{el} = 1$
- 4 Zainicjuj licznik elementów szeregu $i = 2$
- 5 Oblicz sumę szeregu $\text{suma} = \text{suma} + \text{el}$
 - 1 Oblicz kolejny element szeregu $\text{el} = 1/i$
 - 2 Zwiększ licznik elementów o 1
 - 3 Sprawdź, czy kolejny element spełnia warunek $\text{el} \geq \text{eps}$. Jeśli tak, przejdź do kroku 5, w przeciwnym wypadku przejdź do kroku 6.
- 6 Podaj wartość sumy szeregu

Suma szeregu harmonicznego – rozwiązanie

```
1 public class szereg //klasa publiczna , nieabstrakcyjna , niefinalna
2 {
3     public static void main (String[] args)
4     {
5         long ii ;
6         double el , suma , eps ;
7         eps=0.00023;
8         el=1;
9         suma=0;
10        ii=2;
11        do
12        { suma=suma+el ;
13          el=1.0/ii ; //obliczenia na wartościach double
14          ii++;
15        }
16        while (el>=eps); //sumuj tak długo , aż kolejny element sumy będzie mniejszy od
17                        //dokładności , co oznacza , że wartość kolejnej sumy wzrośnie
18                        //poniżej założonej dokładności eps
19        System.out.println ("Suma elementow szeregu harmonicznego"
20 +suma
21 + " z dokladnoscia " + eps); //wyświetlanie wartości double
22    }
23 }
```

Szachownica

```
1  szachownica : rysuje szachownicę na ekranie za pomocą zagnieżdżonej pętli for
2  public class szachownica //klasa publiczna, nieabstrakcyjna, niefinalna
3  {
4      public static void main (String [] args)
5      {
6          int x, y, //rozmiar pola szachownicy
7          for (y=0; y<12; y++) // rysowanie kolejnej linii szachownicy
8          {
9              for (x=0; x<12; x++) // rysowanie kolejnej linii rzędu szachownicy
10             if ((y+x)%2==0) //czy suma rząd+kolumna szachownicy parzysta
11                 System.out.print("*"); // jeśli tak – rysowanie zapętlionych pól
12             else
13                 System.out.print(" "); // jeśli nie – rysowanie pustych pól
14             System.out.println( ); // nowa linia
15         }
16     }
17 }
```

Zadanie. Napisać program, który wczytuje od użytkownika liczbę oznaczającą, ile należy wylosować liczb z zakresu od -100 do 100. Następnie dokonuje obliczeń stosunku wylosowanych liczb dodatnich (bez zera) do liczb ujemnych oraz najmniejszej i największej znalezionej liczby. Wynikiem działania programu powinna być informacja na temat wszystkich trzech wartości. W programie nie należy używać tablic.

Przykład dla liczby podanej przez użytkownika 3, następuje losowanie:
11 -5 93

W wyniku na ekran zostaje wyświetlona informacja:
2 -5 93

```
// bo stosunek 2/1 min: -5, max: 93
```

Statystyka – rozwiązanie

```
1 import java.util.Scanner;
2 import java.util.Random;
3
4 public class Main {
5
6     public static void main(String[] args) {
7         Scanner in = new Scanner(System.in);
8         Random r = new Random();
9
10        System.out.println("Podaj liczbę losowań");
11        int los = in.nextInt();
12        int lDodatnie = 0;
13        int lUjemne = 0;
14        int min = 101; // Więcej od największej możliwej
15        int max = -101; // Mniej od najmniejszej możliwej
16        for(int i=0; i<los; i++){
17            // Losowanie
18            int liczba = r.nextInt(201)-100;
19            // Zliczanie dodatnich i ujemnych
20            if (liczba > 0)
21                lDodatnie++;
22            else if (liczba < 0)
23                lUjemne++;
24            // Szukanie minimum (sposób z instrukcją if)
25            if (liczba < min)
26                min = liczba;
27            // Szukanie maksimum (sposób z trójarg. op. war.)
28            max = liczba > max ? liczba : max;
29        }
30        double stosunek = lDodatnie / (float)lUjemne; // Niezbędne rzutowanie
31        System.out.println(stosunek + "\n" + min + "\n" + max);
32    }
33 }
```



Przykład z wykorzystaniem switch (1/3)

```
1 import java.io.IOException;
2 import java.util.Scanner;
3
4 public class Main {
5
6     public static int suma(int a, int b){
7         return a+b;
8     }
9
10    public static double sinus(int n){
11        return Math.sin(n);
12    }
13
14    public static void informacja(){
15        System.out.println("To jest przykład wyświetlania menu z
16                            + "zastosowaniem instrukcji switch.");
17    }
18
19    public static int menu(){
20        System.out.println();
21        System.out.println("*****");
22        System.out.println("*****MENU*****");
23        System.out.println("*****");
24        System.out.println("1. Suma");
25        System.out.println("2. Sinus");
26        System.out.println("3. Informacja");
27        System.out.println("0. Koniec");
28
29        Scanner in = new Scanner(System.in);
30        int w = in.nextInt();
31
32        return w;
33    }
}
```



Przykład z wykorzystaniem switch (2/3)

```
34 public static void main(String[] args) throws IOException {
35     Scanner in = new Scanner(System.in);
36
37     int wybor = menu();
38
39     while (wybor != 0) {
40         switch (wybor) {
41             case 1:
42                 int a, b;
43                 System.out.println("Podaj dwie liczby do zsumowania");
44                 a = in.nextInt();
45                 b = in.nextInt();
46
47                 int wynik = suma(a, b);
48
49                 System.out.format("Suma liczb wynosi "+wynik);
50
51                 break;
52
53             case 2:
54                 System.out.println("Podaj liczbę do obliczenia sinusa");
55                 a = in.nextInt();
56
57                 double wynik2 = sinus(a);
58
59                 System.out.format("Sinus wynosi "+wynik2);
60
61                 break;
62
63             case 3:
64                 informacja();
65
66         }
```



Przykład z wykorzystaniem switch (3/3)

```
67         System.out.println("\nWciśnij Enter, aby kontynuować...");
68         System.in.read();
69
70         wybor = menu();
71     }
72
73
74     System.out.println("*****");
75     System.out.println("\nKoniec programu\n\n");
76 }
77 }
```