Laboratorium **Metod Obliczeniowych** Wydział Elektrotechniki Automatyki i Informatyki Politechnika Świętokrzyska	
Studia: **Stacjonarne I stopnia**	Kierunek: **Informatyka**
Data wykonania: **20.11.2017**	Grupa: **3ID13B**
Imię i nazwisko: **Bartłomiej Osak**	
Numer ćwiczenia: **6**	Temat ćwiczenia: **Różniczkowanie numeryczne**

1. Różniczkowanie numeryczne – metoda Stirlinga w środowisku MATLAB.

Polecenie: Wyznacz przybliżoną wartość pierwszej pochodnej dla $x = 2.81$.

x_i	2.3	2.47	2.64	2.81	2.99	3.16	3.33
y_i	56.133	48.1988	40.9923	34.484	28.3211	23.1575	18.602

Kod źródłowy:

```
function[F1, F2] = metoda_stirlinga(X,Y)
    n = size(X,2);
    T=zeros([size(X,2) size(X,2)+1]);
    T(:,1) = X;
    T(:,2) = Y;
    h = T(2,1)-T(1,1);
    disp('Krok: ');
    disp(h);
    fx = 0;
    fx2 = 0;
    for j=3:size(X,2)+1
        for i=size(X,2)-(j-2):-1:1
            T(i,j)=T(i+1,j-1)-T(i,j-1);
        end
    end
    disp('Tablica roznic centralnych: ');
    disp(T);
    a=[1,-1/6,1/30];
    a2=[1,1/12,1/90];
    for i=2:2:n
        fx = fx + a(i/2)*(0.5*(T((((n-i)+1)/2),i+1)+T((((n-i)+1)/2+1),i+1)));
        fx2 = fx2 + a2(i/2)*(T((((n-i)+1)/2),i+2));
    end
    F1 = fx * 1/h;
    F2 = fx2 * 1/(h^2);
end
```

Wywołanie:

- **Przypisanie do wektorów X oraz Y wartości:**

  `X=[2.3 2.47 2.64 2.81 2.99 3.16 3.33]`

  `Y=[56.133 48.1988 40.9923 34.484 28.3211 23.1575 18.602]`

- **Wywołanie funkcji przypisując wyniki do dwóch zmiennych:**

  `[F1 F2] = metoda_stirlinga(X,Y)`

- **Wynik zadziałania:**

```
Krok:  0.1700

Tablica różnic centralnych:

    2.3000   56.1330   -7.9342    0.7277   -0.0295   -0.3233    1.3300   -3.3818
    2.4700   48.1988   -7.2065    0.6982   -0.3528    1.0067   -2.0518         0
    2.6400   40.9923   -6.5083    0.3454    0.6539   -1.0451         0         0
    2.8100   34.4840   -6.1629    0.9993   -0.3912         0         0         0
    2.9900   28.3211   -5.1636    0.6081         0         0         0         0
    3.1600   23.1575   -4.5555         0         0         0         0         0
    3.3300   18.6020         0         0         0         0         0         0

F1 = -37.4866

F2 = 13.5542
```

Zrzut ekranu z działania programu:

```
Command Window
>> X=[2.3 2.47 2.64 2.81 2.99 3.16 3.33]

X =

    2.3000    2.4700    2.6400    2.8100    2.9900    3.1600    3.3300

>> Y=[56.133 48.1988 40.9923 34.484 28.3211 23.1575 18.602]

Y =

   56.1330   48.1988   40.9923   34.4840   28.3211   23.1575   18.6020

>> [F1 F2] = metoda_stirlinga(X,Y)
Krok:
    0.1700

Tablica różnic centralnych:
    2.3000   56.1330   -7.9342    0.7277   -0.0295   -0.3233    1.3300   -3.3818
    2.4700   48.1988   -7.2065    0.6982   -0.3528    1.0067   -2.0518         0
    2.6400   40.9923   -6.5083    0.3454    0.6539   -1.0451         0         0
    2.8100   34.4840   -6.1629    0.9993   -0.3912         0         0         0
    2.9900   28.3211   -5.1636    0.6081         0         0         0         0
    3.1600   23.1575   -4.5555         0         0         0         0         0
    3.3300   18.6020         0         0         0         0         0         0


F1 =

  -37.4866


F2 =

   13.5542
```

Opis programu:

Program metoda_stirlinga oblicza pierwszą oraz drugą pochodną dla punktu środkowego wektora X. Przyjmuje dwa argumenty wywołania:

- X – wektor wartości x-owych
- Y – wektor wartości y-owych

Na początku do zmienne n przypisywany jest rozmiar wektora X. Następnie zerowana jest macierz T, do której następnie przypisujemy odpowiednio do pierwszej kolumny wartości wektora X, a do drugiej wartości z wektora Y. Następnie obliczamy krok (skok), czyli do zmiennej h przypisujemy różnicę drugiego elementu wektora X oraz pierwszego elementu wektora X. Dalej inicjowane są zmienne fx oraz fx2 wartościami zerowymi. Zmienne te będą nam potrzebne w celu przechowywania tymczasowego wyniku w trakcie wykonywania operacji iterowania, czyli obliczania pochodnej. Następnie obliczana jest tablica różnic centralnych za pomocą iteracji. Wykorzystujemy tutaj pętlę for. Iteracja rozpoczyna się od wartości 3 do n+1. W pętli zewnętrznej definiujemy pętlę wewnętrzną for, która iteruje od wartości n-(iterator_zewnętrzny-2) do wartości 1 dekrementując o 1. W taki sposób otrzymujemy tablicę różnic centralnych, która jest niezbędna do dalszych obliczeń. Następnie do wektora a oraz a2 wpisujemy stałe wynikające ze wzoru na pierwszą oraz drugą pochodną obliczanej za pomocą metody Stirlinga. Dalej tworzymy pętlę iterującą od wartości 2 do rozmiaru tablicy (n) co 2. Wewnątrz pętli obliczamy pierwszą oraz drugą pochodną na podstawie wzorów:

$$f^{(1)}(x) = \frac{1}{h}\left[\left(\frac{f\left(x-\frac{1}{2}h\right)+\delta f\left(x+\frac{1}{2}h\right)}{2}\right)-\frac{1}{6}\left(\frac{\delta^3 f\left(x-\frac{1}{2}h\right)+\delta^3 f\left(x+\frac{1}{2}h\right)}{2}\right)+\frac{1}{30}\left(\frac{\delta^3 f\left(x-\frac{1}{2}h\right)+\delta^3 f\left(x+\frac{1}{2}h\right)}{2}\right)+\ldots\right]$$

$$f^{(2)}(x) = \frac{1}{h^2}\left[\delta^2 f(x_0)+\frac{1}{12}\delta^4 f(x_0)+\frac{1}{90}\delta^6 f(x_0)+\cdots\right]$$

Po obliczeniu wartości obu pochodnych zapisywane są do zmiennych F1 oraz F2, które są wartościami zwracanymi przez program.

2. Różniczkowanie numeryczne – metoda Stirlinga w języku Java.

Polecenie: Wyznacz przybliżoną wartość pierwszej pochodnej dla $x = 2.81$.

x_i	2.3	2.47	2.64	2.81	2.99	3.16	3.33
y_i	56.133	48.1988	40.9923	34.484	28.3211	23.1575	18.602

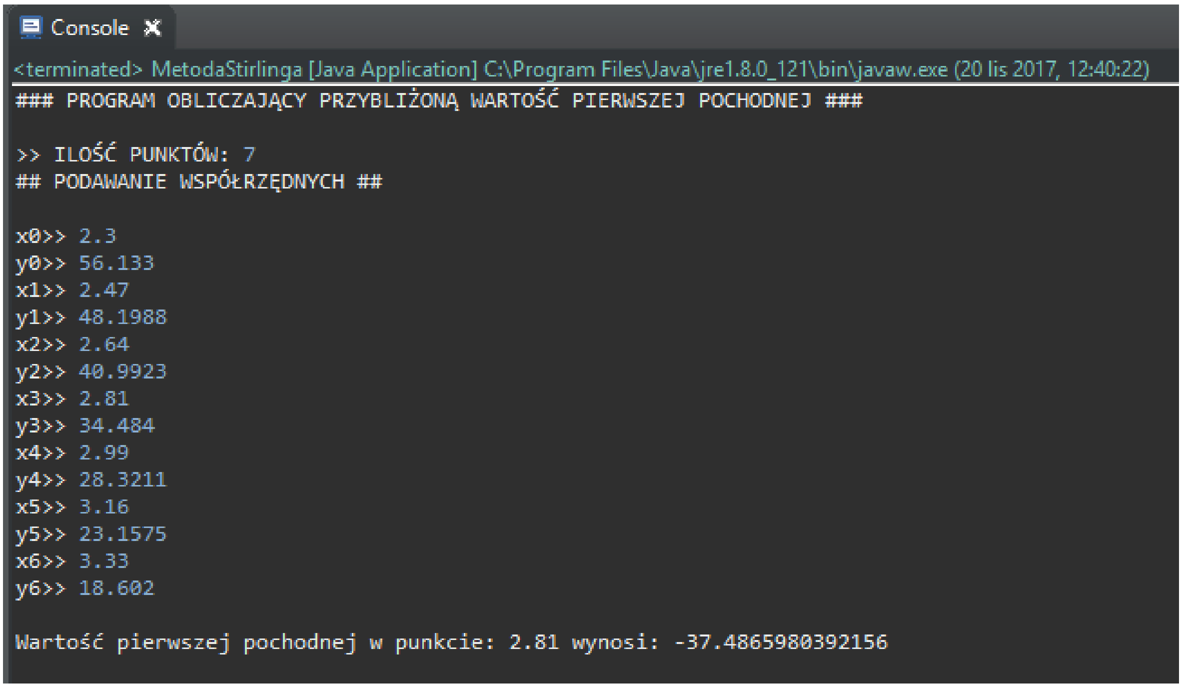
Kod źródłowy:

```
import java.util.Scanner;
public class MetodaStirlinga {
    private static double punkt = 0D;
    private static double[][] pkt;
    private static double[] wsp = { 1.0, -1.0 / 6.0, 1.0 / 30.0 };
    private static int rozmiar = 0;
    private static double h = 0D;
    private static double fx = 0D;
    private static double wynik = 0D;
    private static Scanner in = new Scanner(System.in);
    public static void wprowadzDane() {
        boolean koniec = false;
        String tmpX;
        String tmpY;
        String tmpRozmiar;
        while (!koniec) {
            try {
              System.out.println("### PROGRAM OBLICZAJĄCY PRZYBLIŻONĄ WARTOŚĆ PIERWSZEJ POCHODNEJ
              ###\n");
                System.out.print(">> ILOŚĆ PUNKTÓW: ");
                tmpRozmiar = in.next();
                rozmiar = Integer.parseInt(tmpRozmiar);
                pkt = new double[rozmiar][rozmiar + 1];
                System.out.println("## PODAWANIE WSPÓŁRZĘDNYCH ##\n");
                for (int i = 0; i < rozmiar; i++) {
                    System.out.print("x" + i + ">> ");
                    tmpX = in.next();
                    pkt[i][0] = Double.parseDouble(tmpX);
                    System.out.print("y" + i + ">> ");
                    tmpY = in.next();
                    pkt[i][1] = Double.parseDouble(tmpY);
                    if (i == rozmiar - 1) {
                        koniec = true;
                    }
                }
            } catch (Exception e) {
                System.out.println("\n[BŁAD!] BŁĘDNIE WPISANO DANE!\n");
                for (int i = 0; i < 500; i++) {
                    System.out.println("\n");
                }
            }
        }
    }
    public static void roznicaCentralna() {
        double x = rozmiar / 2.0 + 0.5;
        punkt = pkt[(int) x - 1][0];
        h = pkt[1][0] - pkt[0][0];
        for (int j = 2; j <= rozmiar; j++) {
            for (int i = rozmiar - j; i >= 0; i--) {
                pkt[i][j] = pkt[i + 1][j - 1] - pkt[i][j - 1];
            }
        }
    }
```

```java
    public static void wyznaczPochodna() {
        int j = -1;
        for (int i = 1; i < rozmiar - 1; i += 2) {
            fx += wsp[++j] * (0.5 * (pkt[(rozmiar - (i + 1)) / 2][i + 1] + pkt[((rozmiar - (i + 
1)) / 2) + 1][i + 1]));
        }
        wynik = (1 / h) * fx;
        System.out.println("\nWartość pierwszej pochodnej w punkcie: " + punkt + " wynosi: " + 
wynik);
    }

    public static void main(String[] args) {
        wprowadzDane();
        roznicaCentralna();
        wyznaczPochodna();
    }
}
```

Zrzut ekranu z działania programu:



```
Console
<terminated> MetodaStirlinga [Java Application] C:\Program Files\Java\jre1.8.0_121\bin\javaw.exe (20 lis 2017, 12:40:22)
### PROGRAM OBLICZAJĄCY PRZYBLIŻONĄ WARTOŚĆ PIERWSZEJ POCHODNEJ ###

>> ILOŚĆ PUNKTÓW: 7
## PODAWANIE WSPÓŁRZĘDNYCH ##

x0>> 2.3
y0>> 56.133
x1>> 2.47
y1>> 48.1988
x2>> 2.64
y2>> 40.9923
x3>> 2.81
y3>> 34.484
x4>> 2.99
y4>> 28.3211
x5>> 3.16
y5>> 23.1575
x6>> 3.33
y6>> 18.602

Wartość pierwszej pochodnej w punkcie: 2.81 wynosi: -37.4865980392156
```

Opis programu:

Program składa się z jednej klasy publicznej MetodaStirlinga. Klasa posiada 8 pól statycznych:

- double punkt – pole przechowujące wartość, dla której obliczamy pochodną pierwszego stopnia.
- double[ ][ ] pkt – tablica dwuwymiarowa przechowująca w pierwszej kolumnie wartości x-owe, w drugiej kolumnie wartości Y-owe, w następnych wartości wyznaczane poprzez obliczanie tablicy różnic centralnych.
- int rozmiar – pole przechowuje rozmiar tablicy, czyli w naszym przypadku wynosi on 7.
- double h – pole przechowuje wartość skoku.
- double fx – pole przechowujące wartość tymczasową, używane w metodzie wyznaczPochodna().
- double wynik – pole przechowujące końcowy wynik.
- Scanner in – inicjacja skanera potrzebnego do wpisywania danych przez użytkownika.

Ponadto program posiada 4 metody statyczne:

- wprowadzDane() – metoda odpowiadająca za wprowadzanie danych do pierwszej i drugiej kolumny statycznej tablicy dwuwymiarowej. Zabezpieczona przed podaniem błędnych wartości.

- roznicaCentralna() – metoda odpowiedzialna za wyznaczanie tablicy różnic centralnych zgodnie z podstawowym algorytmem.
- wyznaczPochodna() – metoda wyznaczająca wartość przybliżonej pierwszej pochodnej na podstawie wzoru ogólnego:

$$f^{(1)}(x) = \frac{1}{h}\left[\left(\frac{f\left(x-\frac{1}{2}h\right)+\delta f\left(x+\frac{1}{2}h\right)}{2}\right)-\frac{1}{6}\left(\frac{\delta^3 f\left(x-\frac{1}{2}h\right)+\delta^3 f\left(x+\frac{1}{2}h\right)}{2}\right)+\frac{1}{30}\left(\frac{\delta^3 f\left(x-\frac{1}{2}h\right)+\delta^3 f\left(x+\frac{1}{2}h\right)}{2}\right)+\ldots\right]$$

3. Różniczkowanie numeryczne i błędy różniczkowania numerycznego.

Różniczkowaniem numerycznym nazywamy obliczanie pochodnych funkcji tabelarycznej:

x_i	x_0	x_1	…	x_n
$y_i = f(x_i)$	y_0	y_1	…	y_n

Funkcja dyskretna nie jest funkcją różniczkowalną (nie jest też funkcją ciągłą). Zagadnienie różniczkowania funkcji $y = f(x)$ zadanej tabelą polega na zastąpieniu jej funkcją różniczkowalną $y = F(x)$, której pochodne dobrze przybliżają pochodne funkcji $y = f(x)$, a następnie przyjęciu, że:

$$f^{(k)}(x) \approx F^{(k)}(x)\ dla\ k = 1,2,\ldots$$

Obliczona wartość $F^{(k)}(x)$ jest obarczona **błędem metody różniczkowania:**

$$r(k,x) = f^{(k)}(x) - F^{(k)}(x)$$

Najczęściej w charakterze funkcji $y = F(x)$ przyjmuje się wielomian, gdyż wielomiany mają ciągłe pochodne wszystkich rzędów dla każdego rzeczywistego x oraz przybliżają jednostajnie funkcję ciągłą $y = f(x)$ na odcinku domkniętym $[a: b]$.

Błąd w różniczkowaniu numerycznym można zauważyć rozważając funkcję $f(x) = e^x$. Licząc pochodną w punkcie x_0 korzystając z dwupunktowych różnic centralnych:

$$f^{(1)}(x_i) = \frac{f(x_{i+1}) - f(x_{i-1})}{2h} + O(h^2), \qquad gdzie\ x_{i+1} = h\ oraz\ x_{i-1} = -h$$

$$f^{(1)}(0) = \frac{e^h - e^{-h}}{2h} + O(h^2)$$

Podczas obliczeń komputer wprowadza zaokrąglenia. Gdy zmniejszymy h, błąd obcięcia maleje, lecz błąd zaokrąglenia nadal rośnie.