УО «Белорусский государственный университет информатики и радиоэлектроники»

Кафедра ПОИТ

Отчет по лабораторной работе №1.4

по предмету

Основы алгоритмизации и программирования

Вариант 14

Выполнил:

Крутько А.А.

Проверила:

Данилова Г.В.

Группа 251004

Минск 2022

Задание:



Код программы на **Delphi**:

Program LabFourthBlockFirst;

Uses

System.SysUtils, Math;

Const

MIN\_NUM = 3;

Var

N, I: Integer;

Abscisses, Ordinates: Array of Real;

Perimetr: Real;

IsIncorrect: Boolean;

Begin

Repeat

Write('Введите число сторон у многоугольника — N (N > 2): ');

IsIncorrect := True;

Try

Readln(N);

Except

IsIncorrect := False;

Writeln('Проверьте правильность ввода');

End;

If (IsIncorrect) and (N < MIN\_NUM) Then

Begin

IsIncorrect := False;

Writeln('Проверьте правильность ввода данных в нужном диапазоне');

End;

Until IsIncorrect;

SetLength(Abscisses, N);

SetLength(Ordinates, N);

Perimetr := 0;

//Массив начнется с нуля (0) после SetLength:

For I := 0 to (N - 1) do

Begin

Repeat

IsIncorrect := True;

Write('Abscissa[',I + 1,']= ');

Try

Readln(Abscisses[I]);

Except

IsIncorrect := False;

Writeln('Проверьте правильность ввода');

End;

Until IsIncorrect;

End;

For I := 0 to (N - 1) do

Begin

Repeat

IsIncorrect := True;

Write('Ordinate[',I + 1,']= ');

Try

Readln(Ordinates[I]);

Except

IsIncorrect := False;

Writeln('Проверьте правильность ввода');

End;

Until IsIncorrect;

End;

//сначала ищем периметр (N-1) сторон, после цикла найдем сторону, соединяющую   
 начало и конец.

For I := 0 to (N - 2) do

Begin

Perimetr := Perimetr + Sqrt((Abscisses[I] - Abscisses[I + 1]) \* (Abscisses[I]   
 - Abscisses[I + 1]) + (Ordinates[I] - Ordinates[I + 1])   
 \* (Ordinates[I] - Ordinates[I + 1]));

End;

Perimetr := Perimetr + Sqrt((Abscisses[0] - Abscisses[N - 1]) \* (Abscisses[0]   
 - Abscisses[N - 1]) + (Ordinates[0] - Ordinates[N - 1]) \*   
 (Ordinates[0] - Ordinates[N - 1]));

Write('Периметр равен: ', FloatToStr(Perimetr));

Readln;

End.

Код программы на **C++**:

#include <iostream>

using namespace std;

int main()

{

setlocale(LC\_ALL, "Russian");

int n;

float perimetr = 0;

float \*abscisses, \*ordinates;

const int MIN\_NUM = 3;

bool isIncorrect;

do {

isIncorrect = false;

cout << "Введите число сторон многоугольника N, N > 2: " << endl;

cin >> n;

if (cin.fail())

{

cin.clear();

while (cin.get() != '\n');

isIncorrect = true;

cout << "Проверьте правильность ввода" << endl;

}

if (!isIncorrect && cin.get() != '\n')

{

cin.clear();

while (cin.get() != '\n');

cout << "Проверьте правильность ввода" << endl;

isIncorrect = true;

}

if (!isIncorrect && n < MIN\_NUM)

{

isIncorrect = true;

cout << "Проверьте правильность ввода данных в нужном диапазоне" << endl;

}

} while (isIncorrect);

abscisses = new float[n];

ordinates = new float[n];

for (int i = 0; i < n; ++i)

{

do {

isIncorrect = false;

cout << "Abscissa[" << (i + 1) << "]:";

cin >> abscisses[i];

if (cin.fail())

{

cin.clear();

while (cin.get() != '\n');

isIncorrect = true;

cout << "Проверьте правильность ввода" << endl;

}

if (!isIncorrect && cin.get() != '\n')

{

cin.clear();

while (cin.get() != '\n');

cout << "Проверьте правильность ввода" << endl;

isIncorrect = true;

}

} while (isIncorrect);

}

for (int i = 0; i < n; ++i)

{

do {

isIncorrect = false;

cout << "Ordinate[" << (i + 1) << "]:";

cin >> ordinates[i];

if (cin.fail())

{

cin.clear();

while (cin.get() != '\n');

isIncorrect = true;

cout << "Проверьте правильность ввода" << endl;

}

if (!isIncorrect && cin.get() != '\n')

{

cin.clear();

while (cin.get() != '\n');

cout << "Проверьте правильность ввода" << endl;

isIncorrect = true;

}

} while (isIncorrect);

}

for (int i = 0; i < n - 1; ++i)

{

perimetr += sqrt((abscisses[i] - abscisses[i + 1]) \* (abscisses[i] -   
 abscisses[i + 1]) + (ordinates[i] - ordinates[i + 1]) \*   
 (ordinates[i] - ordinates[i + 1]));

}

perimetr += sqrt((abscisses[0] - abscisses[n - 1]) \* (abscisses[0] -   
 abscisses[n - 1]) + (ordinates[0] - ordinates[n - 1]) \*   
 (ordinates[0] - ordinates[n - 1]));

delete[] abscisses, ordinates;

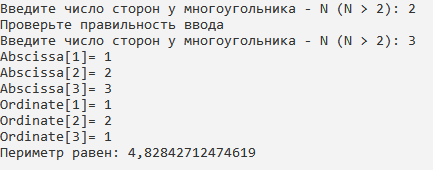
cout << "Периметр равен: " << perimetr;

}

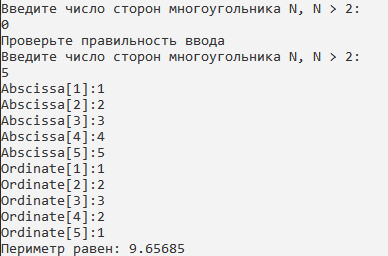
Код программы на **Java**:

import java.util.Scanner;  
public class LabFourthBlockFirst {  
 public static void main(String[] args) {  
 Scanner scan = new Scanner(System.in);  
 int n = 0;  
 float perimetr = 0;  
 float[] abscisses, ordinates;  
 boolean isIncorrect;  
 final int MIN\_NUM = 3;  
 do {  
 System.out.print("Введите число сторон многоугольника N, N > 2: ");  
 isIncorrect = false;  
 try {  
 n = Integer.parseInt(scan.nextLine());  
 } catch (Exception e) {  
 System.err.println("Проверьте правильность ввода данных");  
 isIncorrect = true;  
 }  
 if (!isIncorrect && n < MIN\_NUM) {  
 System.err.println("Проверьте правильность ввода данных в нужном   
 диапазоне");  
 isIncorrect = true;  
 }  
 } while (isIncorrect);  
 abscisses = new float[n];  
 ordinates = new float[n];  
 for (int i = 0; i < n; ++i) {  
 do {  
 isIncorrect = false;  
 System.out.println("Abscissa[" + (i + 1) + "]:");  
 try {  
 abscisses[i] = Float.parseFloat(scan.nextLine());  
 } catch (Exception e) {  
 System.err.println("Проверьте правильность ввода данных в нужном   
 диапазоне");  
 isIncorrect = true;  
 }  
 } while(isIncorrect);  
 }  
 for (int i = 0; i < n; ++i) {  
 do {  
 isIncorrect = false;  
 System.out.println("Ordinate[" + (i + 1) + "]:");  
 try {  
 ordinates[i] = Float.parseFloat(scan.nextLine());  
 } catch (Exception e) {  
 System.err.println("Проверьте правильность ввода данных");  
 isIncorrect = true;  
 }  
 } while (isIncorrect);  
 }  
 scan.close();  
 for (int i = 0; i < n - 1; ++i) {  
 perimetr += Math.sqrt((abscisses[i] - abscisses[i + 1]) \*   
 (abscisses[i] - abscisses[i + 1]) + (ordinates[i] -   
 ordinates[i + 1]) \* (ordinates[i] - ordinates[i + 1]));  
 }  
 perimetr += Math.sqrt((abscisses[0] - abscisses[n - 1]) \*   
 (abscisses[0] - abscisses[n - 1]) +   
 (ordinates[0] - ordinates[n - 1]) \*   
 (ordinates[0] - ordinates[n - 1]));  
 System.out.println("Периметр равен: " + perimetr);  
 }  
}

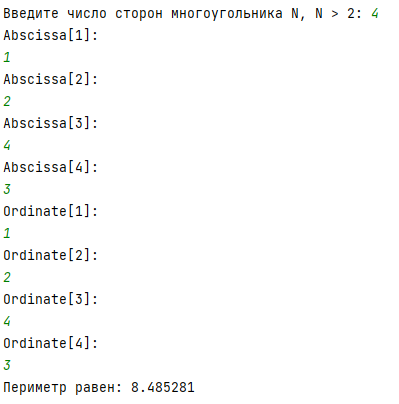
Результат на **Delphi**:



Результат на **C++**:

****

Результат на **Java**:



**Блок-cхема**:

