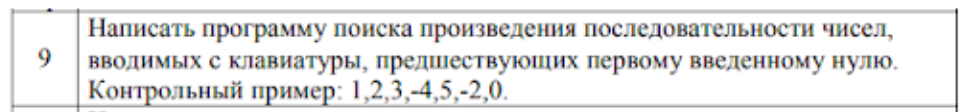
**Лабораторная работа №12**

*Основы программирования на Java:* оператор цикла с постусловием

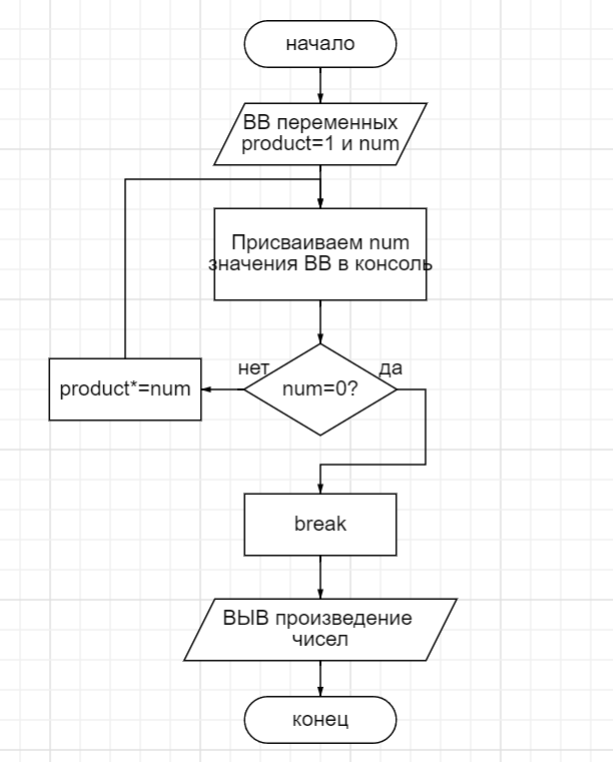
*Цель работы:* Освоить основные способы создания Java-программ либо с помощью обычного редактора, либо с помощью среды разработки. Приобрести навыки программирования циклических алгоритмов в Java.

Задача 1:



| Наименование переменной | Тип данных | Назначение |
| --- | --- | --- |
| scanner | Scanner |  |
| product | int |  |
| num | int |  |

Блок-схема:



import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.*in*);

int product = 1;

int num;

System.*out*.println("Введите последовательность чисел (введите 0 для завершения):");

while (true) {

num = scanner.nextInt();

if (num == 0) {

break;

}

product \*= num;

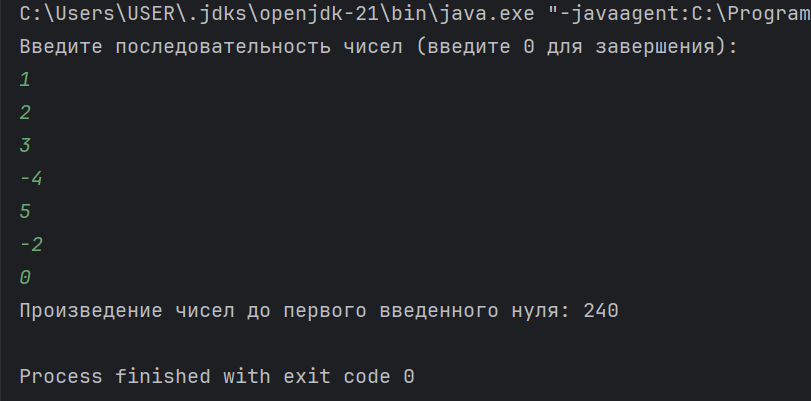
}

System.*out*.println("Произведение чисел до первого введенного нуля: " + product);

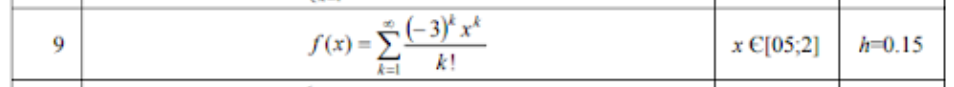
}

}

Результат:

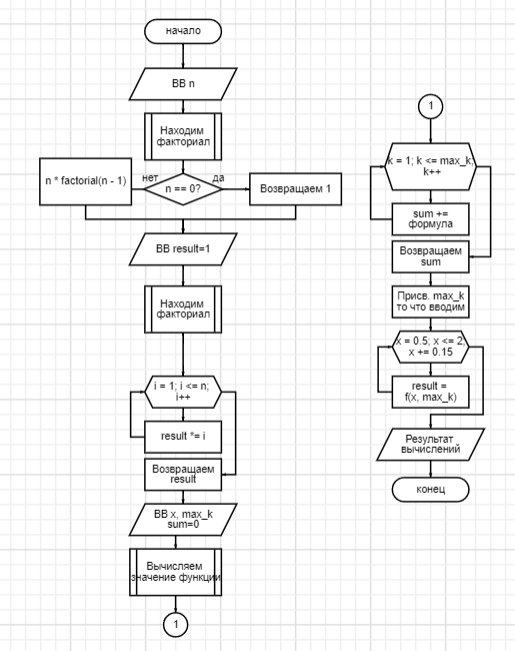


Задача 2:



| Наименование переменной | Тип данных | Назначение |
| --- | --- | --- |
| result | int |  |
| sum | double |  |
| n | int |  |
| i | int |  |
| k | int |  |
| input | Scanner |  |
| max\_k | int |  |
| x | double |  |
| result | double |  |

Блок-схема:



Код

import org.junit.jupiter.api.Test;

import java.util.Scanner;

import static org.junit.jupiter.api.Assertions.*assertEquals*;

public class Main {

public static int factorial(int n) {

if (n == 0) return 1;

return n \* *factorial*(n - 1);

}

public static int factorial\_loop(int n) {

int result = 1;

for (int i = 1; i <= n; i++) {

result \*= i;

}

return result;

}

public static double f(double x, int max\_k) {

double sum = 0;

for (int k = 1; k <= max\_k; k++) {

sum += (Math.*pow*(-3, k) \* Math.*pow*(x, k)) / *factorial*(k);

}

return sum;

}

public static void main(String[] args) {

Scanner input = new Scanner(System.*in*);

System.*out*.print("Введите k: ");

int max\_k = input.nextInt();

for (double x = 0.5; x <= 2; x += 0.15) {

double result = *f*(x, max\_k);

System.*out*.printf("Результат вычисления функции для x=%f, k=%d равен: %f\n", x, max\_k, result);

}

}

}

class MainTest {

@Test

void testFactorialSuccess() {

*assertEquals*(1, Main.*factorial*(0));

*assertEquals*(1, Main.*factorial*(1));

*assertEquals*(2, Main.*factorial*(2));

*assertEquals*(6, Main.*factorial*(3));

*assertEquals*(24, Main.*factorial*(4));

*assertEquals*(120, Main.*factorial*(5));

*assertEquals*(362880, Main.*factorial*(9));

}

@Test

void testFactorialLoopSuccess() {

*assertEquals*(1, Main.*factorial\_loop*(0));

*assertEquals*(1, Main.*factorial\_loop*(1));

*assertEquals*(2, Main.*factorial\_loop*(2));

*assertEquals*(6, Main.*factorial\_loop*(3));

*assertEquals*(24, Main.*factorial\_loop*(4));

*assertEquals*(120, Main.*factorial\_loop*(5));

*assertEquals*(362880, Main.*factorial\_loop*(9));

}

@Test

void testFSuccess() {

*assertEquals*(-3.0, Main.*f*(1, 1));

*assertEquals*(1.5, Main.*f*(1, 2));

*assertEquals*(-6.0, Main.*f*(2, 1));

*assertEquals*(12.0, Main.*f*(2, 2));

*assertEquals*(-1.5, Main.*f*(0.5, 1));

*assertEquals*(-0.375, Main.*f*(0.5, 2));

}

}

Результат:

