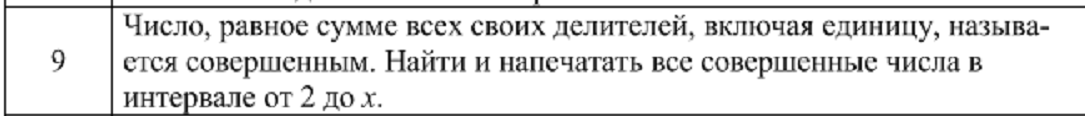
**Лабораторная работа №11**

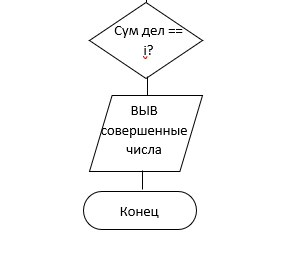
*Основы программирования на Java*: оператор цикла с предусловием

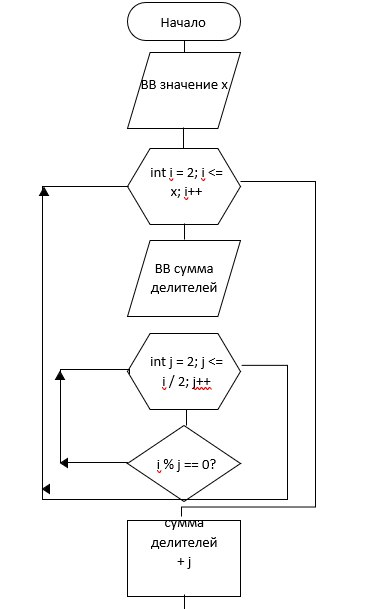
*Цель работы:* Освоить основные способы создания Java-программ либо с помощью обычного редактора, либо с помощью среды разработки. Приобрести навыки программирования циклических алгоритмов в Java.

**Задание 1:**



| Наименование переменной | Тип данных | Назначение |
| --- | --- | --- |
| input | Scanner | Содержит экземпляр класса Scanner, который предназначен для работы с потоком IO (Ввода\Вывода, в данном случае потока ввода - System.in \ STDIN) |
| upperBound | int | Переменная хранит значение верхней границы, до которой будет работать алгоритм поиска совершенных чисел. Получается из входного потока (пользовательского ввода) |
| i | int | Переменная итерации цикла (внешнего цикла), принимает значения от 2 (включительно), до upperBound (включительно) с шагом в единицу. По смыслу - перебирает все числа от 2 до верхней границы для поиска совершенного числа. 1, 0 и отрицательные числа не рассматриваются в данном алгоритме |
| sumOfDivisors | int | Переменная, которая хранит сумму делителей текущего числа i. Инициализируется единицей, так как начинаем итерацию внутреннего цикла с двух |
| j | int | Переменная итерации цикла (внутреннего цикла), принимает значения от 2 (включительно), до i деленного на два (включительно) с шагом в единицу.  По смыслу - перебирает все потенциальные делители текущего числа i для поиска делителя и последующего суммирования полученных делителей |

**Блок-схема:**

****

**Код:**

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

System.out.print("Введите х: ");

int upperBound = input.nextInt();

for (int i = 2; i <= upperBound; i++) {

int sumOfDivisors = 1;

for (int j = 2; j <= i / 2; j++) {

if (i % j == 0) {

sumOfDivisors += j;

}

}

if (sumOfDivisors == i) {

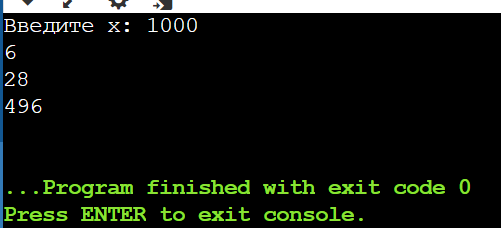
System.out.println(i);

}

}

}

}

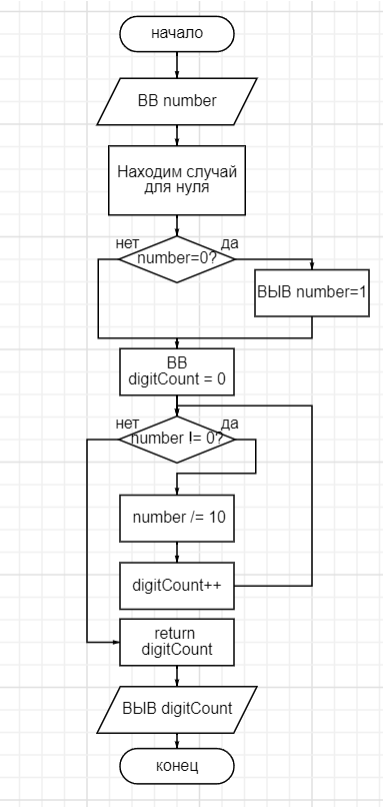


**Задание 2:**



| Наименование переменной | Тип данных | Назначение |
| --- | --- | --- |
| digitCount | int | Хранение количества разрядов числа |
| number | int | Хранение считанного числа с пользовательского ввода |

**Блок-схема:**



**Код:**

Решение с помощью циклов

import org.junit.jupiter.api.Test;

import java.util.Scanner;

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

public class Main {

public static int getDigitCount(int number) {

// Выделим заранее случай для нуля, иначе использовать цикл do while или for(;;) c постусловием внутри

if (number == 0) {

return 1;

}

// Инициализируем переменную количества цифр в числе нулем

int digitCount = 0;

// Цикл, пока число не равно нулю

while (number != 0) {

// Убираем у числа последний разряд путем целочисленного деления на 10 и присваиваем полученное число в number

number /= 10;

// Увеличиваем счетчик количества разрядов числа

digitCount++;

}

return digitCount;

}

public static void main(String[] args) {

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

System.*out*.print("Введите х: ");

int number = input.nextInt();

int digitCount = Main.*getDigitCount*(number);

System.*out*.println(digitCount);

}

}

class DigitCounterTest {

// Простой тест для проверки работоспособности написанной функции

// https://habr.com/ru/articles/591587/

@Test

void testGetDigitCountSuccess() {

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

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

*assertEquals*(2, Main.*getDigitCount*(10));

*assertEquals*(2, Main.*getDigitCount*(11));

*assertEquals*(3, Main.*getDigitCount*(100));

*assertEquals*(4, Main.*getDigitCount*(1000));

*assertEquals*(4, Main.*getDigitCount*(-1000));

}

}

Решение без цикла с переводом int в строку и последующим подсчетом количества символов:

import java.util.Scanner;

public class Main {

public static void main(String[] args) throws Exception {

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

System.*out*.print("Введите х: ");

int number = input.nextInt();

if (number < 0) {

throw new Exception(String.*format*("По условиям задачи число должно быть неотрицательным, получено %d", number));

}

int digitCount = Integer.*toString*(number).length();

System.*out*.println(digitCount);

}

}

Решение без цикла с переводом int в строку и последующим подсчетом количества символов с учетом ввода отрицательного числа:

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

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

System.*out*.print("Введите х: ");

int number = input.nextInt();

int digitCount = Integer.*toString*(number).length();

// Если число отрицательное, то впереди есть лидирующий минус, который нужно вычесть из длины строки

if (number < 0) {

digitCount--;

}

System.*out*.println(digitCount);

}

}

Контрольные вопросы для защиты

⦁ Что такое цикл?

⦁ Какие операторы цикла в Java?

⦁ Оператор while? Примеры.

⦁ Операторы break, continue.