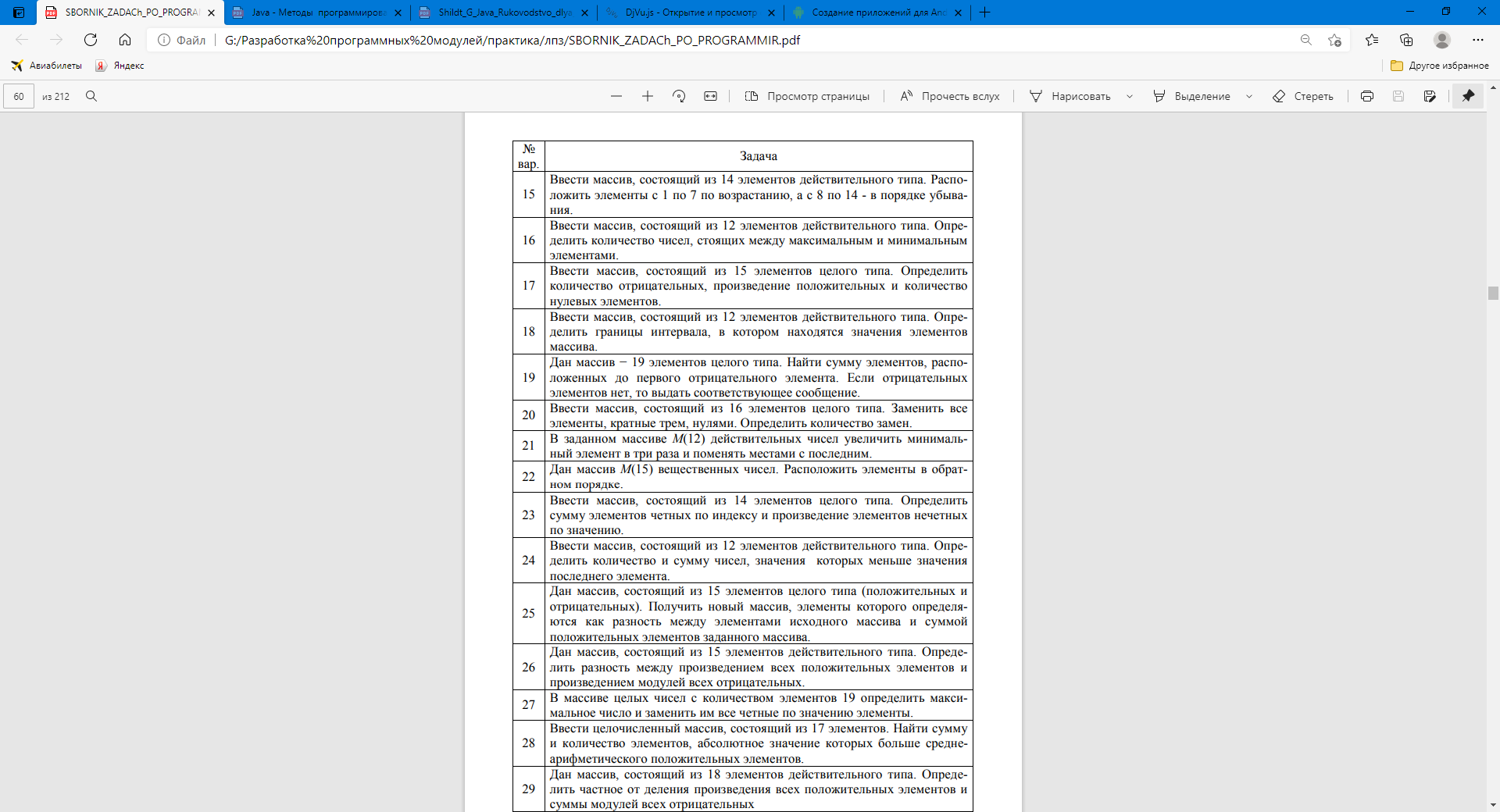
**Лабораторная работа №13**

**Основы программирования на Java: одномерные массивы**

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

**Задание 1**



|  |  |  |
| --- | --- | --- |
| Наименование переменной | Тип данных | Назначение |
| array | int | входные, промежуточные |
| F | float | выходные |
| pN | float | промежуточные |
| nN | float | промежуточные |

**Блок-схема:**

array, F, pN, nN

i<array.length

array[i]= ввод пользователя

array[i]>=0

array[i]<0

pN\*=array[i]

FasfasfddasfddasfsdfsdfsdЕКУЦЕЦУКЕЦУЕКЕ nN+=array[i]

Начало

F=pN/nN

nN==0

Вывод текста: на ноль делить нельзя

Да

Да

Нет

Нет

Да

Нет

F

Конец

**Код:**

int[] array = new int[18];

float pN=1;

float nN=0;

float F=0;

Scanner scanner = new Scanner(System.in);

for (int i = 0; i<array.length; i++) {

array[i]=scanner.nextInt();

if(array[i]>=0) {

pN \*=array[i];

}else if(array[i]<0) {

nN+=Math.abs(array[i]);

}

}

if (nN==0) {

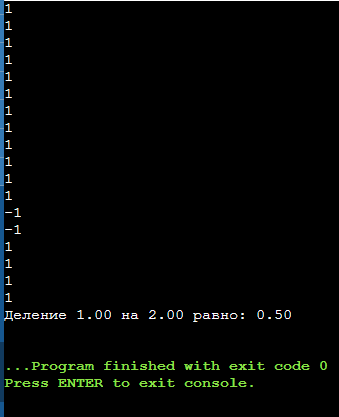
System.out.println("На ноль делить нельзя");

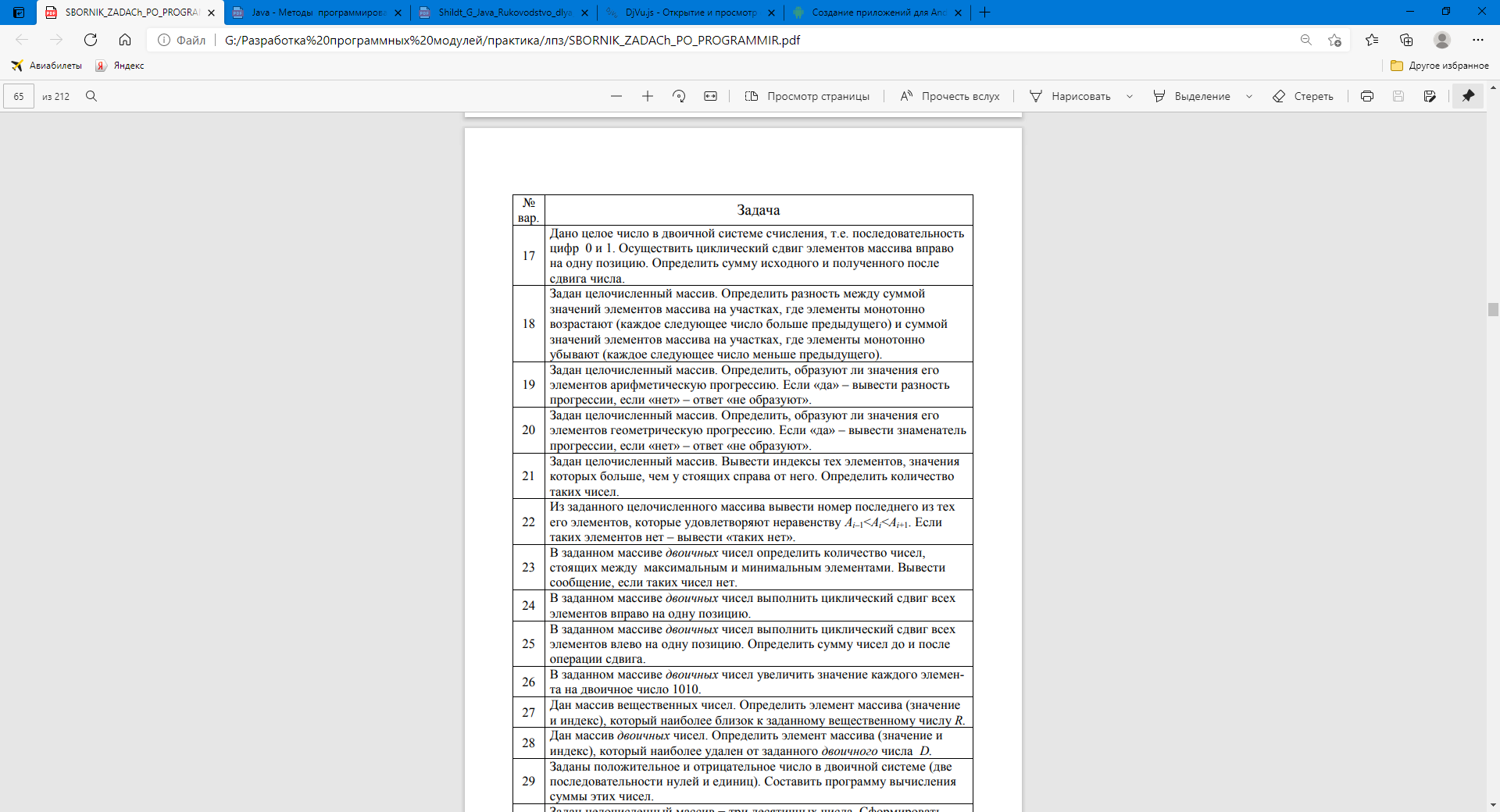
} else {

F=pN/nN;

System.out.printf("Деление %.2f на %.2f равно: %.2f%n", pN, nN, pN / nN);

}

**Результат:**

**Задание 2:**

|  |  |  |
| --- | --- | --- |
| Наименование переменной | Тип данных | Назначение |
| userInput | string | входные |
| num | double | промежуточные |
| sum | промежуточные выходные |
| array | char[] | промежуточные |
| scanner | Scanner | входные |

**Блок-схема:**

**Метод toDecimalNumber**

array, userInput,num

i=array.length;i>=0;i--

Преобразование ввода в массив символов

array[i]==’1’

num+=2^i

Пыавнрценупцуепйуу return num

Да

Нет

scanner, sum

Начало

sum

Конец

sum = toDecimalNumber(ввод)-toDecimalNumber(ввод)

**Код:**

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

double sum = toDecimalNumber(scanner.nextLine()) - toDecimalNumber(scanner.nextLine());

System.out.println(sum);

}

public static double toDecimalNumber(String userInput) {

char[] array = userInput.toCharArray();

double num = 0;

for (int i = array.length - 1; i >= 0; i--) {

if (array[i] == '1') {

num += Math.pow(2, i);

}

}

return num;

}

**Результат:**



**Контрольные вопросы:**

1. Массив в Java - это упорядоченная коллекция элементов одного типа данных, которая хранит фиксированное количество элементов. Элементы массива доступны по их индексам.
2. В Java массив объявляется с указанием типа данных элементов и их количества. Например:

int[] myArray; // объявление массива целых чисел

1. Для доступа к элементу массива в Java используется его индекс, который начинается с нуля. Например, чтобы получить доступ к третьему элементу массива, используется индекс 2. Примеры доступа к элементам массива:

int[] myArray = {10, 20, 30, 40, 50};

int firstElement = myArray[0];

int thirdElement = myArray[2];

1. Существует несколько способов инициализации массива в Java:
   * Использование оператора new:

int[] myArray = new int[5]; // массив из 5 элементов, все элементы инициализируются значением по умолчанию

* + Инициализация значений при объявлении:

int[] myArray = {10, 20, 30, 40, 50};

* + Динамическая инициализация:

int[] myArray = new int[] {10, 20, 30, 40, 50};