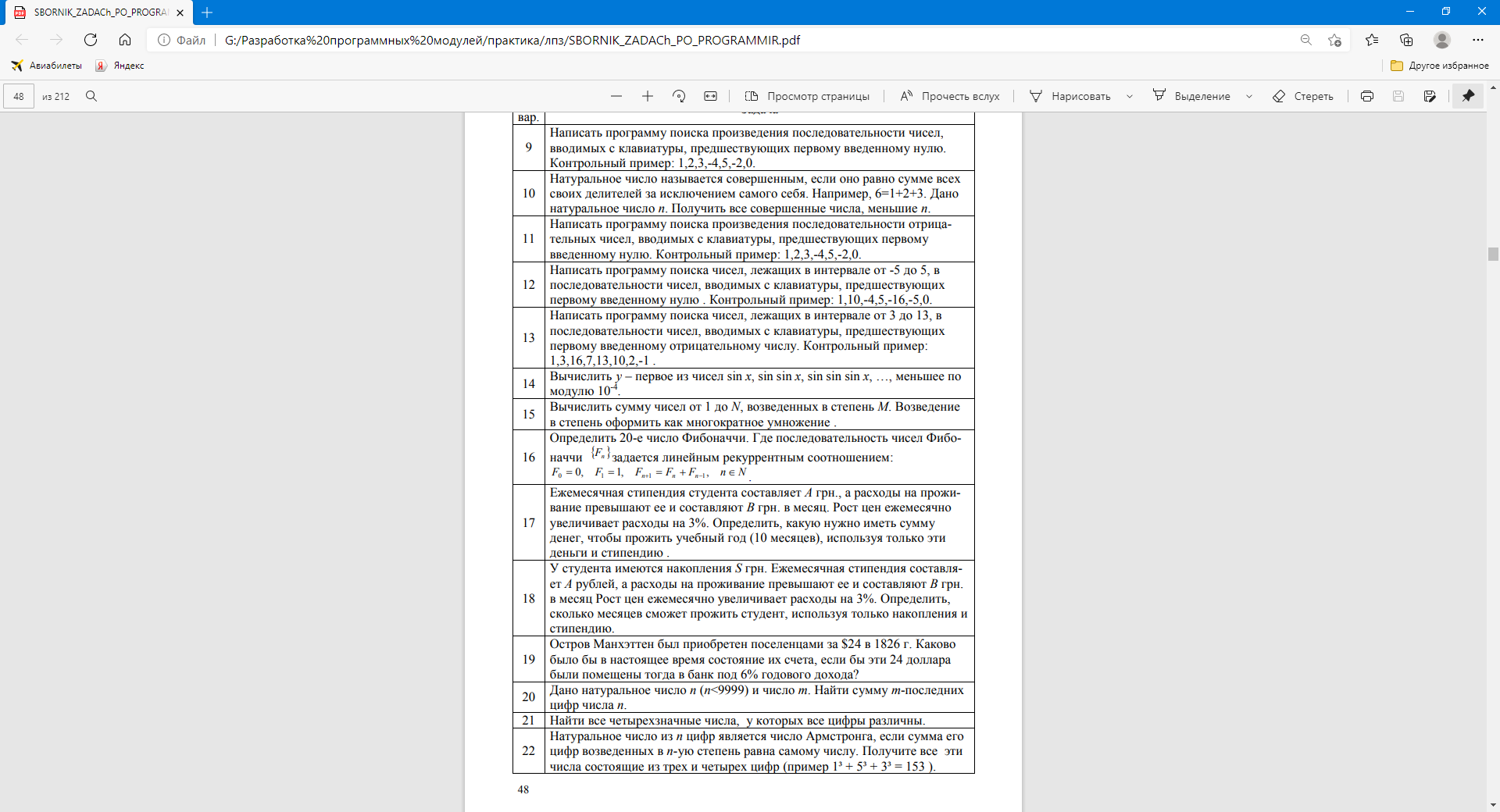
**Лабораторная работа №12**

**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 |
| scholarship | Int | Входная |
| expenses |
| months |
| totalMoney | Выходная |
| inflationRate | double | Входная |

Начало

Вв scholarship expenses months inflationRate

totalMoney = scholarship \* months

прибавляем стипендию и вычитаем расходы, увеличиваем расходы на 3%, вычитаем месяц

months > 0

Выв totalMoney

Конец

public class Main {

public static void main(String[] args) {

int scholarship = 1000; // ежемесячная стипендия A

int expenses = 1200; // расходы на проживание B

double inflationRate = 0.03;

int months = 10; // количество месяцев в учебном году

int totalMoney = scholarship \* months; // общая сумма денег на учебный год

do {

totalMoney += scholarship - expenses; // прибавляем стипендию и вычитаем расходы

expenses \*= (1 + inflationRate); // увеличиваем расходы на 3%

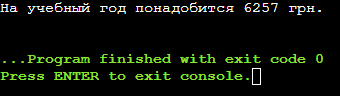
months--;

} while (months > 0);

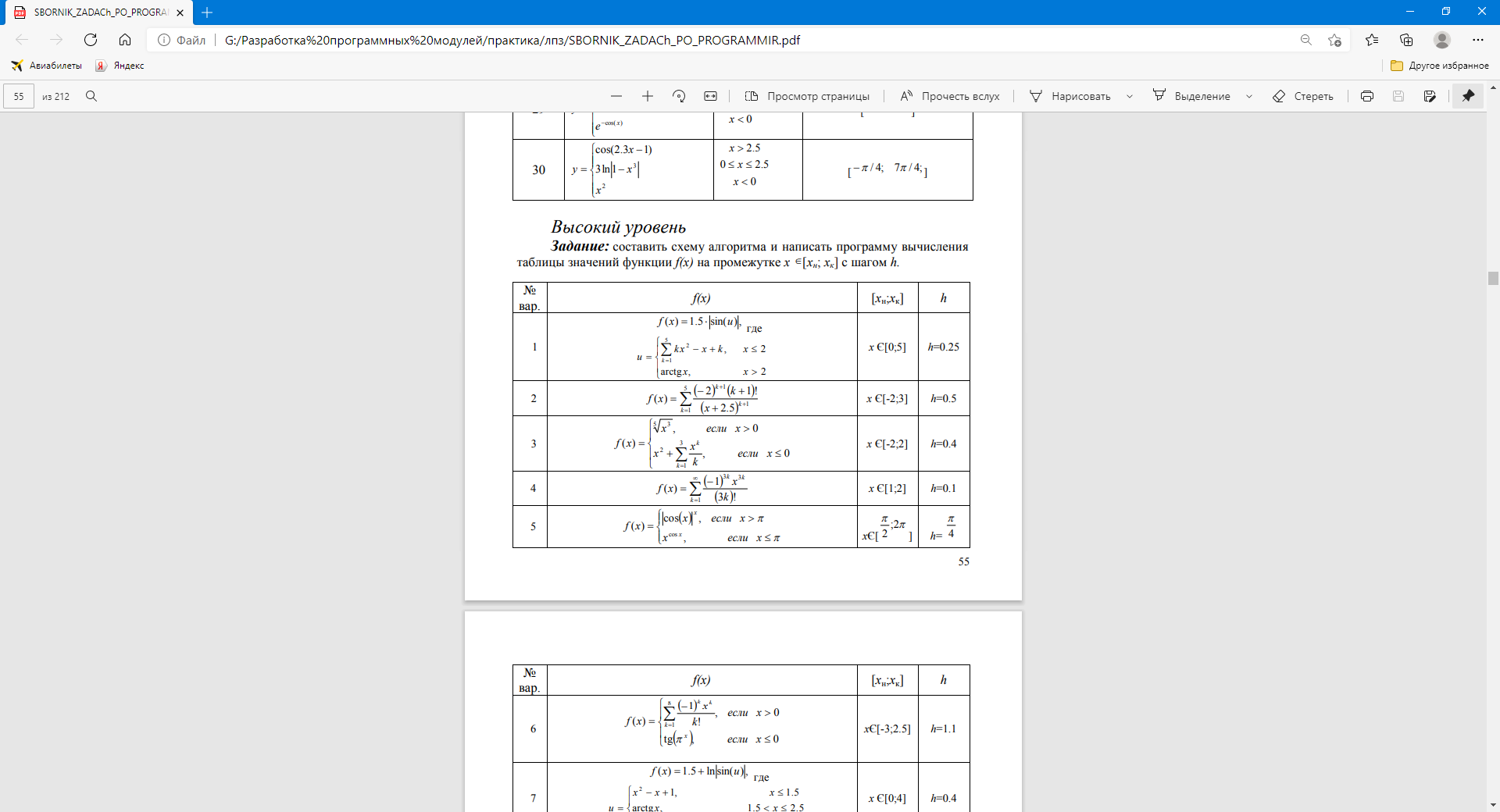
System.out.println("На учебный год понадобится " + totalMoney + " грн.");

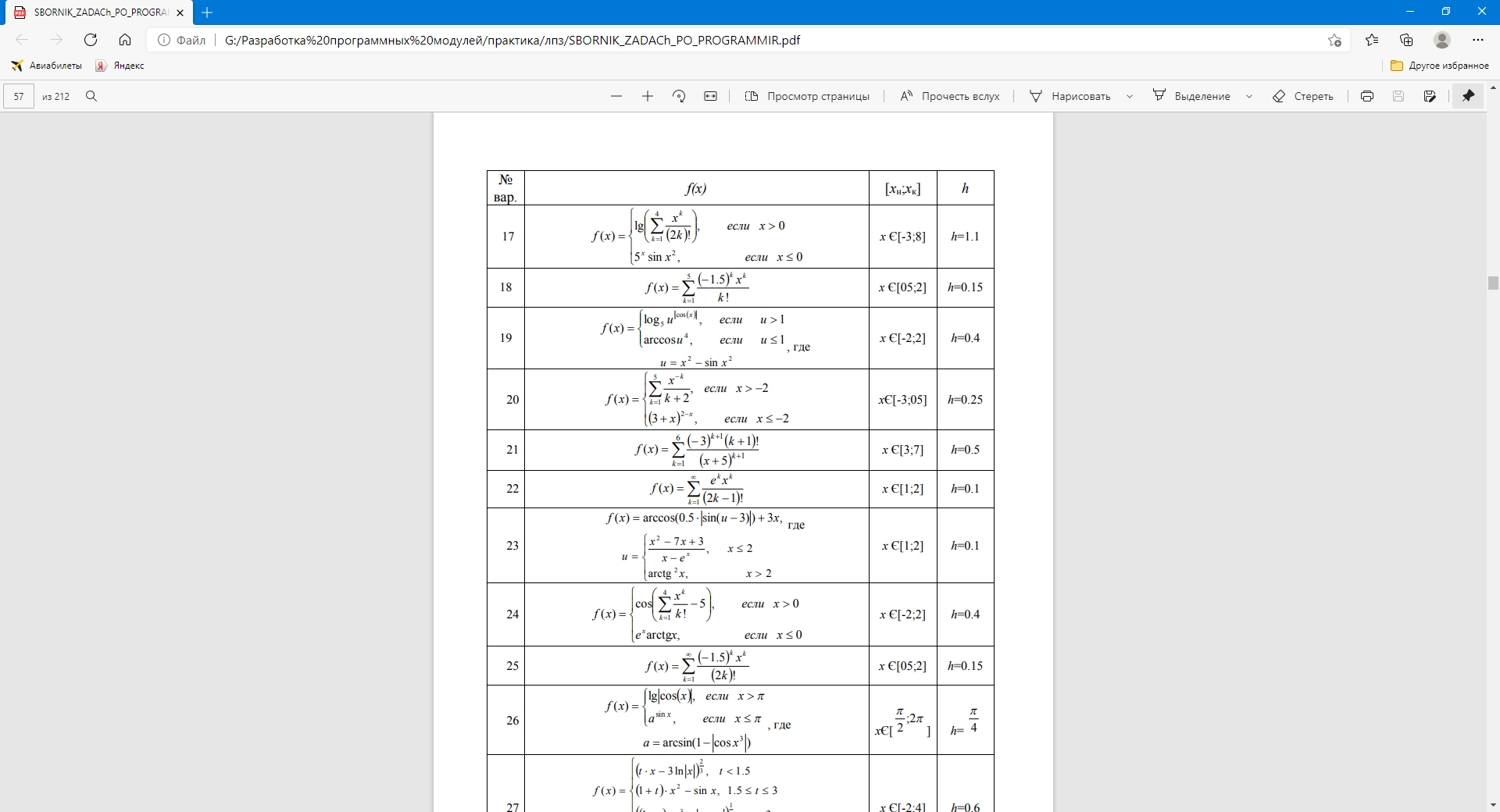
}

}



**Задача №2**





|  |  |  |
| --- | --- | --- |
| Наименование переменной | Тип данных | Назначение |
| x, h | double | Входная |
| result | Выходная |
| sum | Вспомогательная |
| numerator |
| denominator |
| k | int |

Начало

Вв переменных с табл

do с условием

x > 0

Math.pow(5, x) \* Math.sin(Math.pow(x, 2))

x <= 8.0

int k = 1; k <= 4; k++

sum += k

Math.log(sum) + Math.pow(x, 3)

int k = 1; k <= 4; k++

denominator \*= (2 \* k)

numerator / denominator

Конец

public class Main {

public static void main(String[] args) {

double x = -3.0; // начальное значение x

double h = 1.1; // шаг

double result;

// Цикл вычисления функции для заданного диапазона x

do {

if (x > 0) {

result = calculateFirstFunction(x);

} else {

result = calculateSecondFunction(x);

}

// Выводим результат, округленный до целого числа

System.out.println("F(" + x + ") = " + Math.round(result));

// Увеличиваем x на шаг h

x += h;

} while (x <= 8.0); // Проверяем условие окончания цикла

}

// Вычисление первой функции

private static double calculateFirstFunction(double x) {

double sum = 0;

for (int k = 1; k <= 4; k++) {

sum += k;

}

double numerator = Math.log(sum) + Math.pow(x, 3);

double denominator = 1;

for (int k = 1; k <= 4; k++) {

denominator \*= (2 \* k);

}

return numerator / denominator;

}

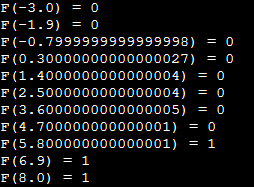
// Вычисление второй функции

private static double calculateSecondFunction(double x) {

return Math.pow(5, x) \* Math.sin(Math.pow(x, 2));

}

}



**Контрольные вопросы для защиты**

1. Что такое цикл?

это конструкция, позволяющая выполнять одну и ту же последовательность операций несколько раз.

1. Какие операторы цикла в Java?

for, while и do-while.

1. Оператор do-while? Особенности.

это цикл, который выполняется как минимум один раз, даже если условие не выполнено. Особенность оператора do-while заключается в том, что проверка условия происходит после выполнения тела цикла.

1. Операторы break, continue.

break в Java используется для выхода из цикла досрочно. Когда оператор break выполняется внутри цикла, выполнение цикла прерывается и управление передается за пределы цикла.

Оператор continue в Java используется для перехода к следующей итерации цикла, минуя оставшуюся часть кода внутри цикла и начиная снова с проверки условия.