МИНИСТРЕРСТВО ОБРАЗОВАНИЯ И НАУКИ

РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОУ НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

ИМ. Р.Е. АЛЕКСЕЕВА

ИНСТИТУТ РАДИОЭЛЕКТРОНИКИ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

КАФЕДРА " Вычислительные системы и технологии "

Дисциплина "Программирование на языках высокого уровня"

**Отчет**

по лабораторной работе № 2

Тема: *«»*

Вариант 12

Выполнил: *Соловьева Анастасия Сергеевна*

студент группы: *19-САИ*

Проверил: Тюрин Иван Васильевич

Нижний Новгород

2021

***Задача №1.***

*1.1Создайте switch, который выводит в каждой секции case сообщение вида:”Секция номер: 1”.*

*1.2 Разместите switch в цикле проходящем все допустимые значения switch.*

*1.3.Удаляем break*

*2.1.Напишите функцию. Которая принимает целочисленный аргумент и возвращает последовательность Фибоначчи, количество элементов в которой равно аргументу. Через цикл*

*2.2 с использованием рекурсии*

*3. Написать программу, которая находит все четырехзначные числа вампиры*

***Алгоритм:***

**1.1.**

switch(Group){

case(1):

System.out.println (“Секция номер 1”);

break;

case(2):

System.out.println (“Секция номер 2”);

break;

case(3):

System.out.println (“Секция номер 3”);

break;

default:

System.out.println (“Что-то не так»);

}

**1.2**

**int group = 1;**

**for(int i = 1; i < 4; i++){**

switch(group){

case(1):

System.out.println (“Секция номер 1”);

break;

case(2):

System.out.println (“Секция номер 2”);

break;

case(3):

System.out.println (“Секция номер 3”);

break;

default:

System.out.println (“Что-то не так»);

}

**group++;**

**}**

**1.3**

int group = 1;

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

switch(group){

case(1):

System.out.println (“Секция номер 1”);

case(2):

System.out.println (“Секция номер 2”);

case(3):

System.out.println (“Секция номер 3”);

default:

System.out.println (“Что-то не так»);

}

group++;

}

**2.1**

public class FibWhile {

public static void main(String[] args)

{

int maxNumber = 10, previousNumber = 0, nextNumber = 1;

System.out.print("fib("+maxNumber+")->");

int i=1;

while(i <= maxNumber)

{

System.out.print(previousNumber+" ");

int sum = previousNumber + nextNumber;

previousNumber = nextNumber;

nextNumber = sum;

i++;

}

}

}

**2.2**

public class Fib{

public static int fibRec(int n){

if(n == 0){

return 0;

}

if(n == 1 || n == 2){

return 1;

}

return fibRec(n-2) + fibRec(n-1);

}

public static void main(String args[]) {

int maxNumber = 10;

System.out.print("fib("+maxNumber+") ->");

for(int i = 0; i < maxNumber; i++){

System.out.print(fibRec (i) +" ");

}

}

}

**3.**

public class VampiresSearch {

static int digitsInNumber(int number, int digit) {

int total = 0;

while (number > 0) {

if (number % 10 == digit)

total++;

number /= 10;

}

return total;

}

public static void main(String[] args) {

int n, n1, n2;

for (int d11 = 1; d11 <= 9; d11++) // перебираем старшую цифру первого "клыка" (идем от 1 к 9)

toNextNumber1:

for (int d12 = 0; d12 <= 9; d12++) // перебираем младшую цифру первого "клыка" (от 0 до 9)

for (int d21 = 9; d21 > 0; d21--) // перебираем старшую цифру второго "клыка" (от 9 до 1)

for (int d22 = 9; d22 >= 0; d22--) { // перебираем младшую цифру второго "клыка" (от 9 до 0)

if (d12 == 0 && d22 == 0) continue; // отсеиваем случаи, когда оба числа оканчиваются 0

n1 = d11 \* 10 + d12; // вычисляем первый потенциальный клык

n2 = d21 \* 10 + d22; // вычисляем второй потенциальный клык

n = n1 \* n2;

if (n < 1000 || n1 > n2) continue toNextNumber1; // если произведение не 4-значное, пропускаем

// Второе условие гарантирует отсутствие повтора из-за перемены мест клыков

// Собственно, главная проверка, является ли число вампиром,

// т.е. для любой цифры количества её вхождений в исходное число и в клыки должны быть равными

if (digitsInNumber(n, d11) == digitsInNumber(n1, d11) + digitsInNumber(n2, d11)

&& digitsInNumber(n, d12) == digitsInNumber(n1, d12) + digitsInNumber(n2, d12)

&& digitsInNumber(n, d21) == digitsInNumber(n1, d21) + digitsInNumber(n2, d21)

&& digitsInNumber(n, d22) == digitsInNumber(n1, d22) + digitsInNumber(n2, d22)) {

System.out.println(n + " = " + n1 + " \* " + n2);

}

}

}

}