**Министерство науки и высшего образования Рф**

**ФГБОУ ВО «БРЯНСКИЙ ГОСУДАРСТВЕННЫЙ  
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**Кафедра «Информатика и программное обеспечение»**

**Лабораторные работы**

**по дисциплине «Проектирование информационных систем»**

**Всего листов 27**

Студент гр. О-19-ИВТ1-ПО-Б

Зач. кн.№ 19.0281

\_\_\_\_\_\_\_\_\_\_\_Бабаев М.Д.

«19» сентября 2022 г.

Преподаватель

\_\_\_\_\_\_\_\_\_Азарченков А.А.

«19» сентября 2022 г.

**Брянск** **2022 г.**

Содержание

[1. Лабораторная работа №1 3](#_Toc114523478)

[1.1. Условие задания 3](#_Toc114523479)

[1.2. Реализация программы 3](#_Toc114523480)

[1.3. Словарь программы 3](#_Toc114523481)

[1.4. Оценка характеристики программы 4](#_Toc114523482)

[2. Лабораторная работа №2 6](#_Toc114523483)

[2.1. Условие задания 6](#_Toc114523484)

[2.2. Реализация программы 6](#_Toc114523485)

[2.3. Словарь программы 6](#_Toc114523486)

[2.4. Словарь операндов программы 7](#_Toc114523487)

[2.5. Входные и выходные переменные программы 7](#_Toc114523488)

[2.6. Оценка характеристик программы 8](#_Toc114523489)

[3. Лабораторная работа №3 10](#_Toc114523490)

[3.1. Условие задания 10](#_Toc114523491)

[3.2. Реализация программы 10](#_Toc114523492)

[3.3. Оценка характеристик программы 11](#_Toc114523493)

# Лабораторная работа №1

## Условие задания

Определить число, образованное 𝑘 старшими цифрами введенного с клавиатуры натурального числа. Исходное число и значение к вводятся с клавиатуры. Пример: для числа 456771 и 𝑘 = 2 искомое число равно 45.

## Реализация программы

Реализация программы представлена на листинге 1.

import java.util.Scanner;

public class Main {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        int x = scanner.nextInt();

        int k = scanner.nextInt();

        int numCount = 0;

        int xCopy = x;

        while (xCopy > 0) {

            numCount++;

            xCopy /= 10;

        }

        System.out.println(x / tenTimes(numCount - k));

    }

    public static int tenTimes(int times) {

        int ten = 1;

        while (times > 0) {

            ten \*= 10;

            times--;

        }

        return ten;

    }

}

Листинг . Реализация программы

## Словарь программы

В таблице 1 приведены операторы и операции, используемые в программе (столбец 2). Номера строк исходной программы, где встречается каждый оператор или операция, указаны в третьем столбце. В четвертом столбце указано число повторений каждого оператора или операции в исходном тексте программы.

Таблица 1

Словарь операторов и операций

|  |  |  |  |
| --- | --- | --- | --- |
| № | Операторы, операции | Номера строк | Количество повторений |
| 1 | new | 5 | 1 |
| 2 | .nextInt | 6, 7 | 2 |
| 3 | .println | 14 | 1 |
| 4 | return | 23 | 1 |
| 5 | while() | 10, 19 | 2 |
| 6 | = | 5, 6, 7, 8, 9, 18 | 6 |
| 7 | ++ | 11 | 1 |
| 8 | -- | 21 | 1 |
| 9 | - | 14 | 1 |
| 10 | /= | 12 | 1 |
| 11 | \*= | 20 | 1 |
| 12 | > | 10, 19 | 2 |
| 13 | / | 14 | 1 |
| 14 | ; | 5, 6, 7, 8, 9, 11, 12, 14, 18, 20, 21, 23 | 12 |
| 15 | . | 5, 6, 7, 14, 14 | 5 |
| 16 | () | 4(4), 5(5) 6(6), 7(7), 10(10), 14(14), 14(14), 17(17), 19(19) | 9 |
| 17 | {} | 3(25), 4(15), 10(13), 17(24), 19(22) | 5 |
|  |  | Итого | 52 |

## Оценка характеристики программы

Используя формулы (1)

|  |  |  |
| --- | --- | --- |
|  |  | (1) |

и (2)

|  |  |  |
| --- | --- | --- |
|  |  | (2) |

Рассчитаем характеристики программы:

1. Абсолютная сложность , что равно количеству операторов while()
2. Относительная сложность
3. Количество операторов цикла
4. Количество операторов условия
5. Количество модулей или подсистем
6. Отношение числа связей между модулями к числу между модулями (1)
7. Отношение числа ненормальных выходов из множества операторов к общему числу операторов (2)

Из полученных результатов анализа текста программы следует, что исходный код имеет невысокую сложность, так как на 52 операторов текста приходится всего лишь 2 оператора условий. Общее число программных модулей решения также невелико (2 модуля), что подтверждает низкий уровень сложности программы.

# Лабораторная работа №2

## Условие задания

Написать и протестировать функцию, которая «переворачивает» строку, передаваемую ей в качестве параметра, в зеркальное состояние.

## Реализация программы

Реализация программы представлена на листинге 2.

import java.util.Scanner;  
  
public class Main {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.in);  
 String str = scanner.nextLine();  
 char[] array = str.toCharArray();  
 String result = "";  
 for (int i = array.length - 1; i >= 0; i--) {  
 result += array[i];  
 }  
 System.out.println(str);  
 }  
}

Листинг . Реализация программы

## Словарь программы

В таблице 2 приведены операторы и операции, используемые в программе (столбец 2). Номера строк исходной программы, где встречается каждый оператор или операция, указаны в третьем столбце. В четвертом столбце указано число повторений каждого оператора или операции в исходном тексте программы.

Таблица 2

Словарь операторов и операций

|  |  |  |  |
| --- | --- | --- | --- |
| № | Операторы, операции | Номера строк | Количество повторений |
| 1 | new | 5 | 1 |
| 2 | .nextLine | 6 | 1 |
| 3 | .println | 12 | 1 |
| 4 | .toCharArray | 7 | 1 |
| 5 | for() | 9 | 1 |
| 6 | = | 5, 6, 7, 8, 9 | 5 |
| 7 | >= | 9 | 1 |
| 8 | - | 9 | 1 |
| 9 | -- | 9 | 1 |
| 10 | плюс равно | 10 | 1 |
| 11 | [] | 7(7), 10(10) | 2 |
| 12 | () | 5(5), 6(6), 7(7), 9(9), 12(12) | 5 |
| 13 | {} | 3(14), 4(13), 9(11) | 3 |
| 14 | ; | 5, 6, 7, 8, 9, 9, 10, 12 | 8 |
| 15 | . | 6, 7, 9, 12, 12 | 5 |
|  |  | Итого | 37 |

## Словарь операндов программы

В таблице 3 приведены операнды рассматриваемой программы.

Таблица 3

Словарь операндов

|  |  |  |  |
| --- | --- | --- | --- |
| № | Операнды | Номера строк | Количество повторений |
| 1 | Main | 3 | 1 |
| 2 | main | 4 | 1 |
| 3 | args | 4 | 1 |
| 4 | scanner | 5, 6 | 2 |
| 5 | str | 6 | 2 |
| 6 | array | 7, 9, 10 | 3 |
| 7 | result | 8, 12 | 2 |
| 8 | System.out | 12 | 1 |
|  |  | Итого | 13 |

## Входные и выходные переменные программы

Для рассматриваемой программы список входных и выходных параметров представлены таблице 4.

Таблица 4

Входные и выходные переменные

|  |  |
| --- | --- |
| Входные переменные | Выходные переменные |
| str | result |

## Оценка характеристик программы

Используя сформированные таблицы с необходимыми параметрами и для расчета и применяя соотношения Холстеда, вычислим характеристики рассматриваемой программы в таблице 5.

Таблица 5

Значения метрик Холстеда для программы

|  |  |  |
| --- | --- | --- |
| Наименование характеристик | Обозначение и формула для вычисления | Значение |
| Число простых (уникальных) операторов и операций |  | 15 |
| Число простых (уникальных) операндов |  | 8 |
| Общее число всех операторов и операций |  | 37 |
| Общее число всех операндов |  | 13 |
| Число входных и выходных переменных (параметров) |  | 2 |
| Словарь программы |  | 23 |
| Длина реализации программы |  | 50 |
| Объем программы (в битах) |  | 227 |
| Потенциальный объем программы |  | 8 |
| Уровень реализации программы |  | 0,0352 |
| Уровень реализации языка |  | 0,2816 |
| Работа программирования |  | 6448 |

Вывод

Анализируя полученные показатели, можно сказать, что уровень исследуемой программы весьма низкий, так как потенциальный объем программы в значительной степени меньше ее реального объема, данном решении возможности языка программирования Java использованы на низком уровне.

# Лабораторная работа №3

## Условие задания

В целочисленной матрице А размером 𝑁 × 𝑀 заменить элементы главной диагонали на номера столбцов (числа 𝑁 и М задаются с клавиатуры в диапазоне от 3 до 10). Первоначальное заполнение матрицы осуществить при помощи датчика случайных чисел в диапазоне от -10 до 10. Исходную и видоизмененную матрицы вывести на экран.

## Реализация программы

Реализация программы представлена на листинге 3

public class Main {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.in);  
 int rows, cols;  
 do{  
 System.out.println("Введите число строк и столбков (>=3 И <= 10)");  
 System.out.print("Строки = ");  
 rows = scanner.nextInt();  
 System.out.print("Столбцы = ");  
 cols = scanner.nextInt();  
 }while (rows < 3 || rows > 10 || cols < 3 || cols > 10);  
 int [][] matrix = initMatrix(rows, cols);  
 printMatrix(matrix);  
 replaceMainDiagonal(matrix);  
 System.out.println("==========");  
 printMatrix(matrix);  
 }  
  
 private static void replaceMainDiagonal(int[][] matrix) {  
 int cols = matrix[0].length;  
 for (int i = 0; i < cols; i++){  
 matrix[i][i] = i;  
 }  
 }  
  
 private static void printMatrix(int[][] matrix) {  
 for (int i = 0; i < matrix.length; i++){  
 for(int j = 0; j < matrix[i].length; j++){  
 System.out.print(String.format("%3d", matrix[i][j]) + " ");  
 }  
 System.out.println();  
 }  
 }  
  
 private static int[][] initMatrix(int rows, int cols) {  
 int [][] matrixToFill = new int[rows][cols];  
 Random random = new Random();  
 for(int i = 0; i < rows; i++){  
 for(int j = 0; j < cols; j++){  
 matrixToFill[i][j] = random.nextInt(21) - 10;  
 }  
 }  
 return matrixToFill;  
 }  
}

Листинг 3. Реализация программы

## Оценка характеристик программы

Объявленные переменные программы представлены в таблице 6

Таблица 6

Объявленные переменные программы

|  |  |  |
| --- | --- | --- |
| № | Наименование переменных | Номера строк |
| P (для расчётов и для обеспечения вывода) | | |
| 1 | rows | 7 |
| 2 | cols | 7, 23 |
| M (модифицируемы или создаваемые) | | |
| 1 | i | 24, 30, 41 |
| 2 | j | 31, 42 |
| 3 | matrix | 15 |
| 4 | matrixToFill | 39 |
| C (упрощающие переменные) | | |
| 1 | random | 40 |
| T (не используемые в программе) | | |
|  | отсутсвуют |  |

Функциональные группы следующие:

Тогда основная метрика:

На основе полученных значений метрики Чепина уровень сложности данного решения можно считать сравнительно низким, так как в исходном тексте программы используется незначительное количество переменных, что не затрудняет понимание программы.