

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
Кафедра «Вычислительная техника»**

**ОТЧЁТ  
по дисциплине «Л и ОА в ИЗ»  
по лабораторной работе №1 «Простые структуры данных»**

Выполнил студент гр. 22ВВВЗ:

Гречихин П. П..

Приняли:

к.т.н., доцент Юрова О.В.

к.э.н., доцент Акифьев И.В.

Пенза

2023

## **Цель:**

В данной лабораторной работе мы изучаем простые структуры данных, такие как массивы, строки (массивы символов), а также простейшую составную структуру данных (struct).

## **Общие сведения:**

**Массив** – это конечная совокупность данных одного типа.

Массивы могут состоять из целых чисел, чисел с плавающей запятой, символов и других типов данных. Существуют даже массивы массивов, – многомерные массивы.

Любой массив в программе должен быть описан: вначале указывается тип массива. затем его имя, после имени массива добавляют квадратные скобки ([ ]) внутри которых обычно стоит число, показывающее количество элементов массива. Например:

```
int x[10];
```

Запись определяет x как массив из 10 целых чисел.

Для **многомерных массивов** количество пар квадратных скобок равно размерности массива, а число внутри скобок показывает размер массива по данному измерению. Например, описание двумерного массива выглядит так:

```
int a[2][5];.
```

Такое описание можно трактовать как матрицу из 2 строк и 5 столбцов.

Для обращения к некоторому элементу массива указывают его имя и индекс, заключенный в квадратные скобки (для многомерного массива – несколько индексов, заключенные в отдельные квадратные скобки):

```
a[1][3], x[i], a[0][k+2] и т.д.
```

В случае, если индексом является переменная, то она должна быть целого типа.

Важно запомнить, что индексация массивов в языке Си начинается с нуля, т.е. обращение к первому элементу массива выглядит так:

```
a[0]; a[0][0];
```

Индекс может быть не только целой константой или целой переменной, но и любым выражением целого типа. Переменная с индексами в программе используется наравне с простой переменной (например, в операторе присваивания, в функциях ввода - вывода).

### **Динамический массив:**

Для использования функций динамического распределения памяти необходимо подключение библиотеки `<malloc.h>` или `<stdlib.h>` (в зависимости от компилятора и используемого стандарта языка).

Функции динамического выделения памяти:

`void *malloc(размер массива в байтах);`

`void *calloc(число элементов, размер элемента в байтах);`

Выделяют блок памяти, размером (размер массива в байтах) или (число элементов \* размер элемента в байтах) байт, и возвращает указатель на начало блока.

Элементами массива могут быть как данные простых типов, так и составных типов (например, структуры), поэтому для точного определения размера элемента массива в общем случае используется функция:

`int sizeof(тип элемента);`

Эта функция возвращает количество байт, занимаемое элементом указанного типа.

Память, динамически выделенная с использованием функций `calloc()`, `malloc()`, должна быть освобождена после окончания использования функцией `free(указатель)`.

Для примера, код:

```
int *a; // указатель на массив
```

```
int n = 10;
```

```
a = (int*)malloc(n * sizeof(int)); // Выделение памяти
```

```
a[1] = a[1] + a[3];
```

```
free(a); // Освобождение памяти
```

Создает массив размером 10 элементов и после некоторых преобразований освобождает занимаемую им память.

Для **генерации случайных чисел** в языке Си используется функция `rand()`, которая содержится в библиотеке `<stdlib.h>`.

Функция возвращает случайное целое число в диапазоне от нуля до `RAND_MAX`. Глобальная константа `RAND_MAX` является максимальным значением, которое может возвращаться функцией `rand`, это значение определено в заголовочном файле `stdlib.h`.

Чтобы ограничить сверху случайные числа, можно воспользоваться операцией получения остатка от деления. Остаток от деления на число всегда меньше этого числа. Поэтому для ограничения сверху случайных чисел возьмем остаток от деления на граничное значение:

```
int k;  
k = rand()%100;
```

В приведенном примере значение `k` не превысит 99. Т.е. для `rand()%n` возвращается любое число в диапазоне `[0, n)`. Если необходимо получить числа в диапазоне `[0, n]`, то необходимо изменить выражение на `rand()%(n+1)`.

Для получения чисел из некоторого отрезка `[min, max]` необходимо найти остаток от деления на длину этого отрезка и выполнить сдвиг возвращаемого остатка на нижнюю границу отрезка:

```
int k;  
k = min + rand()%(max-min+1);
```

При повторных запусках программы функция `rand()` будет возвращать одинаковые последовательности чисел. Для получения уникальных последовательностей необходимо инициализировать генератор случайных чисел различными значениями. Для этого используется функция `srand()`, аргументом которой является инициализирующее целое число.

В качестве уникального для каждого запуска программы числа чаще всего используется текущее время, возвращаемое функцией `time()`, прототип которой описан в библиотеке `time.h`. Передав в `time()` в качестве параметра

NULL, мы получим целое число, которое можно передать в srand(), установив, таким образом уникальное инициализирующее значение генератора:

```
srand(time(NULL));
```

```
int k;
```

```
k = rand();
```

Теперь генерируемая последовательность чисел k будет уникальной.

**Структура** - это совокупность переменных одного или нескольких типов, сгруппированных в один элемент.

Объявление структуры осуществляется с помощью ключевого слова struct, за которым идет ее тип и далее список элементов, заключенных в фигурные скобки:

```
struct тип { тип элемента_1 имя элемента_1;
```

```
.....
```

```
тип элемента_n имя элемента_n;
```

```
};
```

Именем элемента может быть любой идентификатор.

Определение структуры:

```
struct student {
```

```
    int Nomzach;
```

```
    char famil[20], name[20], facult[20];
```

```
};
```

После фигурной скобки, заканчивающей список элементов, могут инициализироваться структуры заданного типа, например:

```
struct student{...} Ivanov, Petrov;
```

Доступ к отдельным полям структуры осуществляется с помощью оператора . (оператор точка). Например, доступ к полю name в структуре Ivanov осуществляется так:

```
Ivanov.name;
```

**Практическая часть:**

**Задание 1:** написать программу, вычисляющую разницу между максимальным и минимальным элементами массива.

**Задание 2:** написать программу, реализующую инициализацию массива случайными числами.

**Задание 3:** написать программу, реализующую создание массива произвольного размера, вводимого с клавиатуры.

**Задание 4:** написать программу, вычисляющую сумму значений в каждом столбце (или строке) двумерного массива.

**Задание 5:** написать программу, осуществляющую поиск среди структур student структуру с заданными параметрами (фамилией, именем и т.д.).

### Реализация:

Данная программа полностью написана на языке программирования Java. Её условно можно поделить на три части. Первая выполняет сразу три первых задания. Она создаёт массив, заполняет его случайными числами и ищет разницу между максимальным и минимальным элементами массива. Вторая часть выполняет четвёртое задание. Она создаёт матрицу и ищет сумму каждого столбца или каждой строки по нашему выбору.

```
Введите размер массива: 5
Введите кол-во строк матрицы: 4
Введите кол-во столбцов матрицы: 4
Вы хотите ввести массив в ручную (д/н): y

7 1 1 4 5

4 6 2 3
8 4 9 2
4 7 2 9
3 1 3 9

макс - мин = 6
Введите номер столбца/строки: 2
Сумма 2 столбца: 18
Сумма 2 строки: 23
```

Третья часть выполняет пятое задание. Она создает список и осуществляет в нём поиск.

```
Введите фамилию: Горин
Введите имя: Геннадий
Введите факультет: ИВТ
Введите № зач. книж.: 45865
Введите фамилию: Гречихин
Введите имя: Павел
Введите факультет: ФВТ
Введите № зач. книж.: 86415
Введите фамилию: Горшков
Введите имя: Федор
Введите факультет: ИВТ
Введите № зач. книж.: 9481
Студент Горин Геннадий факультета ИВТ | №45865
Студент Гречихин Павел факультета ФВТ | №86415
Студент Горшков Федор факультета ИВТ | №9481

Введите параметр поиска Павел
Нашелся (¬_¬ ) Студент Гречихин Павел факультета ФВТ | №86415

Process finished with exit code 0
```

### Заключение:

В данной лабораторной работе мы изучили простые структуры данных, такие как массивы, строки (массивы символов), а также простейшую составную структуру данных (struct).

### Листинг:

```
import java.util.*;
class Students{
    String famil,
        name,
        facult;
    int NomZach;
    public Students(String famil, String name, String facult,
int nomzach){
        this.famil = famil;
        this.name= name;
        this.facult = facult;
        this.NomZach = nomzach;
    }

    public void print(){
        System.out.println("Студент " + famil + " " + name + "
факультета " + facult + " | №" + NomZach);
    }
}

class Lab_1{
    ArrayList<Integer> myArray = new ArrayList();
    int[][] myMatrix;
    int num, columns, rows;
    Lab_1(int num, int rows, int columns) {
        this.num = num;
        this.rows = rows;
        this.columns = columns;
        myMatrix = new int[rows][columns];
        Scanner scanner = new Scanner(System.in);
        System.out.print("Вы хотите ввести массив в ручную
(д/н): ");
        char y_n = scanner.next().charAt(0);
        if (y_n != 'д' || y_n != 'y') {
            System.out.println();
            for (int i = 0; i < num; i++) {
                myArray.add((int) (-10 + Math.random() * 20));
                System.out.print(myArray.get(i) + " ");
            }
        } else {
            System.out.println("\nВводите числа: ");
            for (int i = 0; i < num; i++) {
                myArray.add(scanner.nextInt());
            }
        }
        System.out.println("\n");
    }
}
```



```

        for(int i = 0; i < rows; i++)
        {
            for(int j = 0; j < columns; j++)
            {
                myMatrix[i][j] = (int) (1 + Math.random() * 9);
                System.out.print(myMatrix[i][j] + " ");
            }
            System.out.println();
        }
    }

    public void task_1()
    {
        int min = myArray.get(0), max = myArray.get(0);
        for (int i = 1; i < num; i++) {

            if (myArray.get(i) < min) {
                min = myArray.get(i);
            } else if (myArray.get(i) > max) {
                max = myArray.get(i);
            }

        }

        System.out.println();
        //if(min >= 0) {
            System.out.println("макс - мин = " + (max - min));
        //} else {System.out.println("макс - мин = " + (max +
min));}
    }

    public void task_4(char pos) {
        int sum = 0;
        if (pos == 'с') {
            for (int i = 0; i < rows; i++) {
                for (int j = 0; j < columns; j++) {
                    sum += myMatrix[i][j];
                }
                System.out.println("Сумма " + (i + 1) + "
строки: " + sum);
                sum = 0;
            }
        } else if (pos == 'к') {
            for (int i = 0; i < columns; i++) {
                for (int j = 0; j < rows; j++) {
                    sum += myMatrix[j][i];
                }
                System.out.println("Сумма " + (i + 1) + "
столбцов: " + sum);
            }
        }
    }
}

```

```

        sum = 0;
    }
    } else {System.out.println("Ошибка! не верный ввод");}
}

public class Main {
    public static void main(String[] args) {
        int num, rows, columns;
        String famil,
            name,
            facult;
        int NomZach;
        Students[] student = new Students[3];
        Scanner scanner = new Scanner(System.in);
        System.out.print("Введите размер массива: ");
        num = scanner.nextInt();
        System.out.print("Введите кол-во строк матрицы: ");
        rows = scanner.nextInt();
        System.out.print("Введите кол-во столбцов матрицы: ");
        columns = scanner.nextInt();
        Lab_1 t = new Lab_1(num, rows, columns);
        t.task_1();
        System.out.print("Введите что хотите сложить (к/с): ");
        char pos = scanner.next().charAt(0);
        t.task_4(pos);
        scanner.nextLine();
        for(int i = 0; i < 3; i++){
            System.out.print("Введите фамилию: ");
            famil = scanner.nextLine();
            System.out.print("Введите имя: ");
            name = scanner.nextLine();
            System.out.print("Введите факультет: ");
            facult = scanner.nextLine();
            System.out.print("Введите № зач. книж.: ");
            NomZach = scanner.nextInt();
            scanner.nextLine();
            student[i] = new Students(famil, name, facult,
NomZach);
        }

        for(int i = 0; i < 3; i++){
            student[i].print();
        }

        System.out.print("\n\nВведите параметр поиска ");
        String filter = scanner.nextLine();
        boolean key = true;
    }
}

```

```

        for(int i = 0; i < 3; i++){
            try {
                if (
                    (filter.equals(student[i].famil)) ||
                    (filter.equals(student[i].name))
                ||
                (filter.equals(student[i].facult)) ||
                (Integer.parseInt(filter)) ==
                student[i].NomZach
                ) {
                    System.out.print("Нашелся (¬_¬ ) ");
                    student[i].print();
                    key = false;
                }
            } catch (NumberFormatException nfe){
                ;
            }
        }
        if(key) {
            System.out.println("Данного индивида не существует
(ಥ_ಥ)");
        }
    }
}

```