Министерство образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе №1

по курсу “Логика и основа алгоритмизации в ИЗ”

на тему “ Простые структуры данных”

Выполнил студент группы 21ВВ3:

Ванюшин И.А.

Приняли:

Митрохин М.А., Юрова О.В.

Пенза 2022

**Название**

Простые структуры данных.

**Цель работы:** выполнить лабораторные указания 1-5 используя простые структуры данных.

**Лабораторное задание:**

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.).

**Задание 1:**

Максимальный и минимальный элементы массива index вычисляются на этапе заполнения массива случайными значениями в заданном диапазоне, и хранятся в создаваемом объекте.

Алгоритм для нахождения минимального и максимального элементов массива:

foreach (int item in index)

{

if (maxv < item) maxv = item;

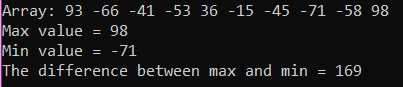
if (minv > item) minv = item;

}

max = maxv;

min = minv;

Результат работы алгоритма:



**Задание 2:**

Заполнение массива случайными числами происходит с помощью объекта типа Random, аналогично в С++ используется функция rand(), возвращающая случайное число в пределах 32767.

Данный метод заполняет случайными числами двумерный массив, однако он наследуется классом IntArray, где одно из измерений массива равно единице:

public void FillRand(int start, int end)

{

Random temp = new Random();

for (int y = 0; y < rows; y++)

for (int x = 0; x < columns; x++)

Index[y, x] = temp.Next() % (end - start + 1) + start;

UpdateProperty();

}

Результат работы алгоритма:



**Задание 3:**

Динамическое выделение памяти в языке C# происходит более автоматизировано чем в языке С++. Это обусловлено и наличием в первом автоматизированного сборщика мусора, а потому программисту не нужно следить за очищением неиспользуемой зарезервированной под значения памяти.

Данная строчка конструктора класса IntArray динамически выделяет память на одномерный массив Index с количеством элементов, равным значению аргумента size:

Index = new int[1, size];

Результат работы программы:



**Задание 4:**

Алгоритм представлен методом класса Matrix, который возвращает массив целых чисел – сумм чисел в столбцах массива:

virtual public int[] ColsAmount()

{

int[] temp = new int[columns];

int amount = 0;

for (int x = 0; x < columns; x++)

{

for (int y = 0; y < rows; y++)

amount += index[y, x];

temp[x] = amount;

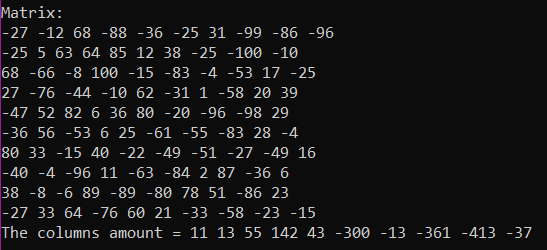
amount = 0;

}

return temp;

}

Результат работы программы:



**Задание 5:**

Структура Student содержит в себе конструктор для удобства.

Создается массив, содержащий структуры Student:

Student[] students = new Student[studentsAmount];

И заполняется в цикле for.

Для поиска инициализируется отдельно от массива структура Student, в которой лежат искомые значения, и массив Conditions, который содержит нули (если значение искомое) и единицы (в противном случае). Если какие-то из значений не нужны, то при помощи компактной записи тернарным оператором мы их не учитываем.

Алгоритм поиска:

Console.WriteLine("\n[Results]\n");

foreach (Student item in students)

{

if ((conditionParams[0] == 0 ? item.firstName == search.firstName : true) &&

(conditionParams[1] == 0 ? item.secondName == search.secondName : true) &&

(conditionParams[2] == 0 ? item.faculty == search.faculty : true) &&

(conditionParams[3] == 0 ? item.recordBook == search.recordBook : true))

{

counter++;

Console.WriteLine($"First Name: {item.firstName}; " +

$"Second Name: {item.secondName}; " +

$"Faculty: {item.faculty}; " +

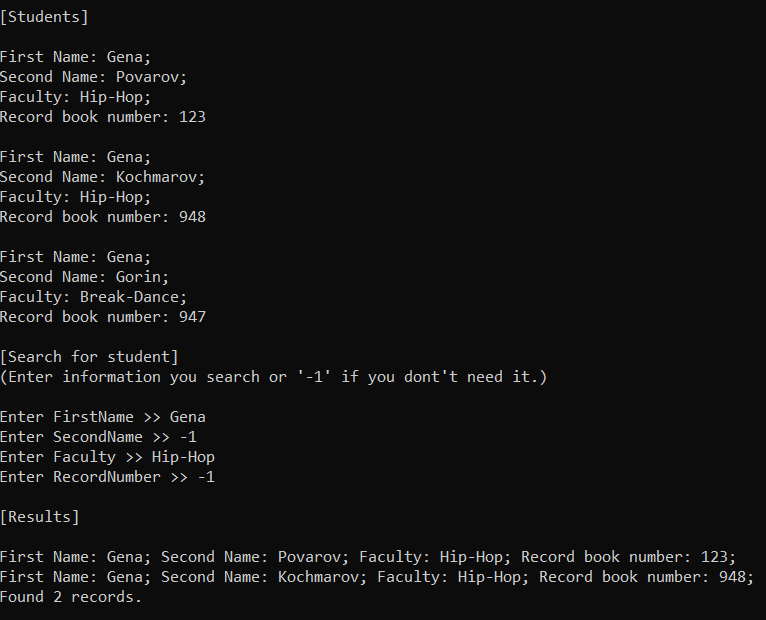
$"Record book number: {item.recordBook};");

}

}

Console.Write($"Found {counter} records.");

Результат работы алгоритма:



**Вывод:** Были выполнены лабораторные указания 1-5 с использованием простых структур данных. Освежены в памяти навыки, приобретенные с предыдущего курса.