

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт Космических и информационных технологий

институт

Кафедра «Информатика»

кафедра

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №3

Массивы, строки, коллекции, кортежи

тема

Вариант 11 (1)

Преподаватель

Студент КИ18-166 031831229

номер группы, зачетной книжки

подпись, дата

подпись, дата

А. А. Чикизов

инициалы, фамилия

В. А. Прекель

инициалы, фамилия

Красноярск 2020

1 Задание

Общее задание для всех вариантов

- Создать параметризованный список структур
- Параметры в открытый конструктор структуры передавать через кортеж (tuple)

например (STUDENT S = new STUDENT (new Tuple< string, string,int []> ("FIO", "GROUP", new int[] {2,2,2,2,2}));

- Произвести сортировку используя интерфейс IComparable по разным полям
- Перегрузить метод ToString() структуры, использовать его для вывода на экран элементов коллекции
- Быть готовым ответить на вопрос почему при использовании коллекции невозможно изменить

один из элементов структуры, а можно заменить элемент только целиком.

Вариант 1

Описать структуру с именем STUDENT, содержащую следующие поля:

- ☐ фамилия и инициалы;
- ☐ номер группы;
- ☐ успеваемость (массив из пяти элементов).

Написать программу, выполняющую следующие действия:

- ☐ ввод с клавиатуры данных в массив, состоящий из десяти структур типа STUDENT (записи должны быть упорядочены по возрастанию номера группы);
- ☐ вывод на экран фамилий и номеров групп для всех студентов, включенных в массив, если средний балл студента больше 4,0 (если таких студентов нет, вывести соответствующее сообщение).

2 Исходный код основного алгоритма

Листинг 1 – CSharpLabs.Lab03/CSharpLabs.Lab03.Core/Student.cs

```
using System;
using System.Linq;

namespace CSharpLabs.Lab03.Core
{
    public struct Student : IComparable<Student>
    {
        public string Name { get; set; }
        public string Group { get; set; }

        public int[] Marks { get; set; }

        public Student(Tuple<string, string, int[]> data)
        {
            (Name, Group, Marks) = data;
        }

        public int CompareTo(Student other) =>
            String.Compare(Name, other.Name, StringComparison.OrdinalIgnoreCase)
            != 0
            ? String.Compare(Name, other.Name,
                StringComparison.OrdinalIgnoreCase)
            : String.Compare(Group, other.Group,
                StringComparison.OrdinalIgnoreCase);

        public double AverageScore() => Marks.Average();

        public override string ToString() =>
            $"Student: Name: {Name}; Group: {Group}; Marks: {String.Join(", ",
                Marks)}; AverageScore: {AverageScore()}";
    }
}
```

Листинг 2 – CSharpLabs.Lab03/CSharpLabs.Lab03.Console/Program.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

using CSharpLabs.Lab03.Core;

static Student InputStudent()
{
    try
    {
        var name = Console.ReadLine();
        var group = Console.ReadLine();
        var marks = Console.ReadLine()?
            .Split(new[] { ',' }, StringSplitOptions.TrimEntries)
            .Select(Int32.Parse)
            .ToArray();
        if (name == null || group == null || marks == null)
        {
            throw new NullReferenceException();
        }
    }
}
```

```

        return new Student(new Tuple<string, string, int[]>(name, group,
marks));
    }
    catch (Exception e)
    {
        throw new ApplicationException("Ошибка при считывании", e);
    }
}

Console.InputEncoding = Encoding.Unicode;
Console.OutputEncoding = Encoding.Unicode;
var list = new List<Student>();
for (var i = 0; i < 10; i++)
{
    list.Add(InputStudent());
}

list.Sort();

Console.WriteLine(String.Join("\n", list));
Console.WriteLine();

var gt4 = list.Where(s => s.AverageScore() > 4).ToList();
Console.WriteLine(gt4.Any() ? String.Join("\n", gt4) : "Нет таких элементов");

```