**Лабораторная работа № 40-41**

**«Описание класса на языке ООП»**

**Цель работы:** получение навыков описания класса на языке ООП в С#.

**Задания для лабораторной работы:**

1. Исходные данные вводятся из файла.

2. Результаты выводятся на экран и в файлы.

3.1 Строка таблицы данных содержит следующую информацию о курсе: ФИО, группа, количество баллов, набранных на вступительных экзаменах. Первые буквы ФИО студентов должны начинаться с первых букв ФИО студента, сдающего практическую работу.

Требуется найти:

3.1.1) перечень фамилий с указанием числа студентов с такой фамилией;

3.1.2) перечень имён с указанием числа студентов с таким именем;

3.1.3) перечень отчеств с указанием числа студентов с таким отчеством;

3.1.4) перечень групп с указанием числа студентов, обучающихся в них;

3.1.5) перечень баллов с указанием числа студентов, набравших это количество баллов;

3.1.6) перечень имен, встречающихся максимальное количество раз;

3.1.7) перечень групп, в которых максимальное количество обучающихся студентов;

3.1.8) перечень студентов, набравших максимальное количество баллов;

3.1.9) перечень групп, в которых максимальное количество студентов, набравших максимальное количество баллов;

3.1.10) перечень групп, в которых средний балл студентов по группе максимален.

class Student

{

public string FullName { get; set; }

public string Group { get; set; }

public int Score { get; set; }

public string LastName => FullName.Split(' ')[0];

public string FirstName => FullName.Split(' ')[1];

public string Patronymic => FullName.Split(' ')[2];

}

class Program

{

static void Main()

{

var students = LoadStudentsFromFile("students.txt");

PrintLastNamesCount(students);

PrintFirstNamesCount(students);

PrintPatronymicsCount(students);

PrintGroupsCount(students);

PrintScoresCount(students);

PrintMostFrequentFirstNames(students);

PrintGroupsWithMostStudents(students);

PrintTopScorers(students);

PrintGroupsWithTopScorers(students);

PrintGroupsWithMaxAverageScore(students);

}

static List<Student> LoadStudentsFromFile(string filePath)

{

var students = new List<Student>();

var lines = File.ReadAllLines(filePath);

foreach (var line in lines)

{

var parts = line.Split(',');

students.Add(new Student

{

FullName = parts[0].Trim(),

Group = parts[1].Trim(),

Score = int.Parse(parts[2].Trim())

});

}

return students;

}

static void PrintLastNamesCount(List<Student> students)

{

var lastNameCount = students.GroupBy(s => s.LastName)

.Select(g => new { LastName = g.Key, Count = g.Count() })

.ToList();

Console.WriteLine("Перечень фамилий и количество студентов:");

foreach (var item in lastNameCount)

{

Console.WriteLine($"{item.LastName}: {item.Count}");

}

}

static void PrintFirstNamesCount(List<Student> students)

{

var firstNameCount = students.GroupBy(s => s.FirstName)

.Select(g => new { FirstName = g.Key, Count = g.Count() })

.ToList();

Console.WriteLine("Перечень имён и количество студентов:");

foreach (var item in firstNameCount)

{

Console.WriteLine($"{item.FirstName}: {item.Count}");

}

}

static void PrintPatronymicsCount(List<Student> students)

{

var patronymicCount = students.GroupBy(s => s.Patronymic)

.Select(g => new { Patronymic = g.Key, Count = g.Count() })

.ToList();

Console.WriteLine("Перечень отчеств и количество студентов:");

foreach (var item in patronymicCount)

{

Console.WriteLine($"{item.Patronymic}: {item.Count}");

}

}

static void PrintGroupsCount(List<Student> students)

{

var groupCount = students.GroupBy(s => s.Group)

.Select(g => new { Group = g.Key, Count = g.Count() })

.ToList();

Console.WriteLine("Перечень групп и количество студентов:");

foreach (var item in groupCount)

{

Console.WriteLine($"{item.Group}: {item.Count}");

}

}

static void PrintScoresCount(List<Student> students)

{

var scoreCount = students.GroupBy(s => s.Score)

.Select(g => new { Score = g.Key, Count = g.Count() })

.ToList();

Console.WriteLine("Перечень баллов и количество студентов:");

foreach (var item in scoreCount)

{

Console.WriteLine($"{item.Score}: {item.Count}");

}

}

static void PrintMostFrequentFirstNames(List<Student> students)

{

var maxCount = students.GroupBy(s => s.FirstName)

.Select(g => new { FirstName = g.Key, Count = g.Count() })

.Max(g => g.Count);

var mostFrequentFirstNames = students.GroupBy(s => s.FirstName)

.Where(g => g.Count() == maxCount)

.Select(g => g.Key)

.ToList();

Console.WriteLine("Имена, встречающиеся максимальное количество раз:");

foreach (var name in mostFrequentFirstNames)

{

Console.WriteLine(name);

}

}

static void PrintGroupsWithMostStudents(List<Student> students)

{

var maxCount = students.GroupBy(s => s.Group)

.Select(g => new { Group = g.Key, Count = g.Count() })

.Max(g => g.Count);

var groupsWithMostStudents = students.GroupBy(s => s.Group)

.Where(g => g.Count() == maxCount)

.Select(g => g.Key)

.ToList();

Console.WriteLine("Группы с максимальным количеством студентов:");

foreach (var group in groupsWithMostStudents)

{

Console.WriteLine(group);

}

}

static void PrintTopScorers(List<Student> students)

{

var maxScore = students.Max(s => s.Score);

var topScorers = students.Where(s => s.Score == maxScore)

.Select(s => s.FullName)

.ToList();

Console.WriteLine("Студенты с максимальным количеством баллов:");

foreach (var scorer in topScorers)

{

Console.WriteLine(scorer);

}

}

static void PrintGroupsWithTopScorers(List<Student> students)

{

var maxScore = students.Max(s => s.Score);

var groupsWithTopScorers = students.Where(s => s.Score == maxScore)

.GroupBy(s => s.Group)

.Select(g => g.Key)

.ToList();

Console.WriteLine("Группы с максимальным количеством студентов, набравших максимальное количество баллов:");

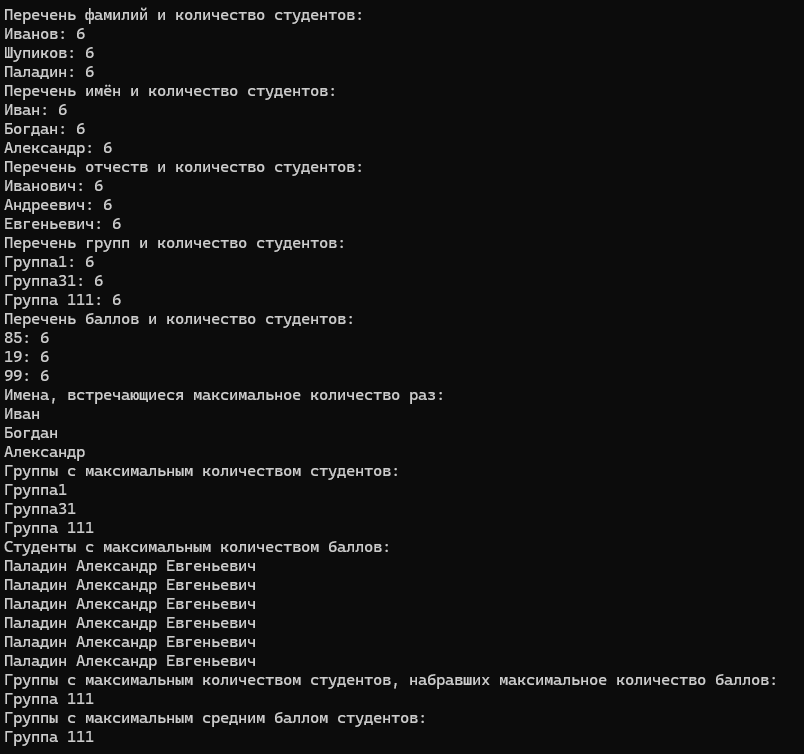
foreach (var group in groupsWithTopScorers)

{

Console.WriteLine(group);

}

}



Контрольные вопросы:

Что такое класс?

Это шаблон для создания объектов, который определяет набор свойств и методов, которые будут доступны для объектов этого класса.

Где определяется класс?

Класс определяется в пространстве имен (namespace) в C#. Он может быть объявлен в отдельном файле или в одном файле с другими классами.

Приведите примеры уровней доступа к компонентам классов.

существуют следующие уровни доступа (модификаторы доступа):

• public: доступен из любого места.

• private: доступен только внутри самого класса.

• protected: доступен внутри класса и в производных классах.

• internal: доступен только внутри текущей сборки.

• protected internal: доступен внутри текущей сборки и в производных классах.

Какие члены класса бывают?

Члены класса могут включать:

• Поля (fields): переменные, которые хранят состояние объекта.

• Свойства (properties): специальные методы для доступа к полям.

• Методы (methods): функции, которые определяют поведение объекта.

• Конструкторы (constructors): специальные методы, которые вызываются при создании объекта.

• Деструкторы (destructors): методы, которые вызываются при уничтожении объекта.

• Индексы (indexers): позволяют обращаться к объекту как к массиву.

Как называются переменные внутри класса?

Переменные внутри класса называются поля (fields).

Приведите пример метода и поля внутри своего класса

namespace ZZZ

{

public class Calculator

{

// Поле

private double \_result;

// Метод для сложения

public void Add(double number)

{

\_result += number;

}

// Метод для получения результата

public double GetResult()

{

return \_result;

}

}

}

**Вывод:** получил навыки описания класса на языке ООП в С#.