Министерство науки и высшего образования РФ
ФГАОУ ВПО
Национальный исследовательский технологический университет «МИСиС»

Институт Информационных технологий и компьютерных наук (ИТКН)

Кафедра Инфокоммуникационных технологий (ИКТ)

Отчет по лабораторной работе №1
по дисциплине «Объектно-Ориентированное Программирование»
на тему «Классы»

Выполнил:
студент группы БИВТ-24-5

Черных Богдан

Проверил:
Стучилин В. В.

Москва, 2025

Задания:

1 уровень

1. Результаты соревнований по прыжкам в длину определяются по сумме двух попыток. В протоколе для каждого участника указываются: фамилия, общество, результаты первой и второй попыток. Вывести протокол в виде таблицы с заголовком в порядке занятых мест.
2. Составить программу для обработки результатов кросса на 500 м для женщин. Для каждой участницы ввести фамилию, группу, фамилию преподавателя, результат. Получить результирующую таблицу, упорядоченную по результатам, в которой содержится также информация о выполнении норматива. Определить суммарное количество участниц, выполнивших норматив.
3. Радиокомпания провела опрос слушателей по вопросу: «Кого вы считаете человеком года?». Определить пять наиболее часто встречающихся ответов и их долей (в процентах от общего количества ответов).

2 Уровень

1. После окончания соревнования по шахматам турнирная таблица содержит фамилии участников и результаты сыгранных партий (выигрыш – 1 очко, ничья – 1/2 очка, проигрыш – 0 очков). Составить итоговую таблицу в порядке убывания полученных участниками очков.
2. Для формирования сборной по хоккею предварительно отобрано 30 игроков. На основании протоколов игр составлена таблица, в которой содержится штрафное время каждого игрока по каждой игре (2, 5 или 10 мин). Написать программу, которая составляет список кандидатов в сборную в порядке возрастания суммарного штрафного времени. Игрок, оштрафованный на 10 мин, из списка кандидатов исключается.
3. Результаты соревнований фигуристов по одному из видов многоборья представлены оценками семи судей в баллах (от 0,0 до 6,0). По результатам оценок судьи определяется место каждого участника у этого судьи. Места участников определяются далее по сумме мест, которые каждый участник занял у всех судей. Составить программу, определяющую по исходной таблице оценок фамилии и сумму мест участников в порядке занятых ими мест.

3 Уровень

1. В соревнованиях участвуют три команды по 6 человек. Результаты соревнований представлены в виде мест участников каждой команды (1 - 18). Определить команду – победителя, вычислив количество баллов, набранное каждой командой. Участнику, занявшему 1-е место, начисляется 5 баллов, 2-е – 4, 3-е – 3, 4-е – 2, 5-е – 1, остальным – 0 баллов. При равенстве баллов победительницей считается команда, за которую выступает участник, занявший 1-е место.
2. Лыжные гонки проводятся отдельно для двух групп участников. Результаты соревнований заданы в виде фамилий участников и их результатов в каждой группе. Расположить результаты соревнований в каждой группе в порядке занятых мест. Объединить результаты обеих групп с сохранением упорядоченности и вывести в виде таблицы с заголовком.
3. Обработать результаты первенства по футболу. Результаты каждой игры заданы в виде названий команд и счета (количество забитых и пропущенных мячей). Сформировать таблицу очков (выигрыш – 3, ничья – 1, проигрыш – 0) и упорядочить результаты в соответствии с занятым местом. Если сумма очков у двух команд одинакова, то сравниваются разности забитых и пропущенных мячей. Вывести результирующую таблицу, содержащую место, название команды, количество очков.

Цель работы: Изучение и практическое применение основ объектно-ориентированного программирования на языке C# посредством создания и использования классов, их полей, методов, конструкторов и механизмов наследования, а также освоение разработки программ для обработки и представления результатов различных спортивных соревнований.

Уровень 1 Задания 1, 2, 3; Уровень 2 Задания 7, 8, 9; Уровень 3 Задания 3, 4, 5 были сделаны для этой работы
Код программы:

```
//svg does precious
using System;                           // Аналог <iostream> для работы с консолью и основными функциями
using System.Collections.Generic;       // Аналог <vector>, <list>, <map>, <set>, <unordered_map>, <unordered_set>, <stack>, <queue>
using System.Text;                      // Аналог <string>, <cstring> (для работы со строками и StringBuilder)
using System.Linq;                      // Аналог <algorithm> (для работы с LINQ, сортировок, поиска и т.д.)
using System.IO;                        // Аналог <cstdio>, <fstream> (работа с файлами)
using System.Globalization;             // Аналог <iomanip> (для форматирования)
using System.Collections;               // Работа с различными коллекциями (например, ArrayList)
using System.Threading;                 // Потоки и многопоточность
using System.Runtime.Serialization;     // Аналог <stdexcept> (работа с исключениями)
using System.Reflection;                // Аналог <typeinfo> (информация о типах, рефлексия)
using System.Diagnostics;               // Аналог <utility>, <std::pair> (вспомогательные функции и классы)
using System.ComponentModel;            // Дополнительные утилиты и атрибуты
using System.Numerics;                  // Работа с большими числами и математическими операциями
using System.Globalization;
using System.Diagnostics;
using System.Net;
using System.Numerics;
// Для работы с потоками данных:
using System.Threading.Tasks;           // Асинхронные задачи

// Для работы с датами и временем:
using System.Timers;                    // Для работы с таймерами и временем
using System.Collections.Generic;
using System.Text;
using System.Linq;
using System.IO; //important
using System.Globalization;
using System.Collections;
using System.Threading;
using System.Runtime.Serialization;
using System.Reflection;
using System.Diagnostics;
using System.ComponentModel;
using System.Numerics;
using System.Globalization;
using System.Diagnostics;
using System.Net;
using System.Numerics;
using System.Threading.Tasks;
using System.Security.Cryptography;
using System.Data;
using System.Data.SqlClient;
using System.Xml;
using System.Xml.Linq;
using System.Runtime.InteropServices;
using System.Security;
```

```
using System.Web;
using System.Media;
using System.Drawing;
using System.Configuration;
using System.Timers;
using System.Runtime.Remoting;
using System.Runtime.CompilerServices;
using System.Runtime.Versioning;
using System.CodeDom;
using System.CodeDom.Compiler;
using System.Collections.Concurrent;
using System.Runtime;
using System.Windows;
using System.Windows.Input;
using System.Security.Principal;
using System.Security.Permissions;
using System.Resources;
using C = System.Console; //console
using dl = System.Decimal;//decimal
using str = System.String;//string
using l = System.Int64;   //long
using u = System.UInt64;  //Ulong
using db = System.Double; //Double

/*
Izzspot - 19 years
Boogie B - 20 years
SJ - 21 years
Bandokay - life sentence

Youngest in the charge
OFB, we don't window shop
Bro caught him an opp and tried turn him off (Bow, bow)
In this X3, man's swervin' off (Skrr, skrr)
Free Boogie Bando, he got birded off (Free my bro, free my bro)
Whenever we get a burner loss
We just cop a next one and go burst it off (Ay)
Lil bro's tellin' me he got his earnings wrong
We just took him OT, now his trapline's gone (Ring, ring)
Hashtag
Bro backed this ting and just started squeezin' (Clarted)
When it broad day, it was freezin'
Hashtag fuckery, hashtag screamin'
One on the hand ting woi, left man leanin', leanin' (Fucker)
Show us cause it's good to feel it
Shortie's cooze and she must be dreamin'
*/
/* ¡Adelante Barcelona, adelante Cataluña! Visca el Barça! Visca Catalunya!
   ¡Al diablo con todos los demás, porque lo más importante en la vida es el
fútbol!
*/
//-------------------------------------------------------------------------------
-----
//-------------------------------------------------------------------------------
-----
//-------------------------------------------------------------------------------
-----

//Лабораторная Работа №7

// 1 Уровень
// Задание 1: Результаты соревнований по прыжкам в длину
```

```
class LongJumpParticipantCustom
{
    public string LastName { get; set; }
    public string Club { get; set; }
    public double JumpAttempt1 { get; set; }
    public double JumpAttempt2 { get; set; }
    public double TotalJumpScore => JumpAttempt1 + JumpAttempt2;

    public LongJumpParticipantCustom(string lastName, string club, double
attempt1, double attempt2)
    {
        LastName = lastName;
        Club = club;
        JumpAttempt1 = attempt1;
        JumpAttempt2 = attempt2;
    }
}

class LongJumpCompetitionCustom
{
    public List<LongJumpParticipantCustom> Participants { get; set; } = new();

    public void AddParticipant(LongJumpParticipantCustom participant)
    {
        Participants.Add(participant);
    }

    public void DisplayResults()
    {
        var sortedParticipants = Participants.OrderByDescending(p =>
p.TotalJumpScore).ToList();
        Console.WriteLine("Результаты соревнований по прыжкам в длину:");
        Console.WriteLine("Место\tФамилия\t\tОбщество\tПопытка 1\tПопытка
2\tСумма");
        for (int i = 0; i < sortedParticipants.Count; i++)
        {
            var p = sortedParticipants[i];
            Console.WriteLine($"{i +
1}\t{p.LastName}\t\t{p.Club}\t\t{p.JumpAttempt1:F2}\t\t{p.JumpAttempt2:F2}\t\t{p.T
otalJumpScore:F2}");
        }
    }
}

// Задание 2: Результаты кросса на 500 м для женщин
class CrossCountryParticipantCustom
{
    public string LastName { get; set; }
    public string Group { get; set; }
    public string Coach { get; set; }
    public double Result { get; set; }
    public bool NormativeMet { get; set; }

    public CrossCountryParticipantCustom(string lastName, string group, string
coach, double result, double normative)
    {
        LastName = lastName;
        Group = group;
        Coach = coach;
        Result = result;
        NormativeMet = result <= normative;
    }
```

```
}

class CrossCountryCompetitionCustom
{
    public List<CrossCountryParticipantCustom> Participants { get; set; } = new();

    public void AddParticipant(CrossCountryParticipantCustom participant)
    {
        Participants.Add(participant);
    }

    public void DisplayResults(double normative)
    {
        var sortedParticipants = Participants.OrderBy(p => p.Result).ToList();
        int normativeCount = sortedParticipants.Count(p => p.NormativeMet);

        Console.WriteLine("Результаты кросса на 500 м для женщин:");

Console.WriteLine("Фамилия\t\tГруппа\t\tПреподаватель\tРезультат\tНорматив");
        foreach (var p in sortedParticipants)
        {

Console.WriteLine($"{p.LastName}\t\t{p.Group}\t{p.Coach}\t\t{p.Result:F1}\t\t{(p.N
ormativeMet ? "Выполнен" : "Не выполнен")}");
        }
        Console.WriteLine($"Всего участниц, выполнивших норматив:
{normativeCount}");
    }
}

// Задание 3: Опрос радиокомпании «Человек года»
class VoteCustom
{
    public string Candidate { get; set; }
    public int VoteCount { get; set; }

    public VoteCustom(string candidate, int count)
    {
        Candidate = candidate;
        VoteCount = count;
    }
}

class SurveyCustom
{
    public List<string> Votes { get; set; } = new();

    public void AddVote(string candidate)
    {
        Votes.Add(candidate);
    }

    public void DisplayTopCandidates(int topN)
    {
        var voteCounts = Votes.GroupBy(v => v)
                              .Select(g => new VoteCustom(g.Key, g.Count()))
                              .OrderByDescending(v => v.VoteCount)
                              .Take(topN)
                              .ToList();

        int totalVotes = Votes.Count;
        Console.WriteLine("Результаты опроса 'Человек года':");
```

```
        Console.WriteLine("Кандидат\tГолосов\t\tДоля (%)");
        foreach (var v in voteCounts)
        {
            double percentage = (double)v.VoteCount / totalVotes * 100;

Console.WriteLine($"{v.Candidate}\t\t{v.VoteCount}\t\t{percentage:F2}");
        }
    }
}

// 2 Уровень

// Задание 7: Турнирная таблица по шахматам
class ChessPlayerCustom
{
    public string LastName { get; set; }
    public double Points { get; set; }

    public ChessPlayerCustom(string lastName)
    {
        LastName = lastName;
        Points = 0.0;
    }

    public void AddResult(double result)
    {
        Points += result;
    }
}

class ChessTournamentCustom
{
    public List<ChessPlayerCustom> Players { get; set; } = new();

    public void AddPlayer(ChessPlayerCustom player)
    {
        Players.Add(player);
    }

    public void AddGameResult(string player1LastName, string player2LastName,
double player1Result, double player2Result)
    {
        var player1 = Players.FirstOrDefault(p => p.LastName == player1LastName);
        var player2 = Players.FirstOrDefault(p => p.LastName == player2LastName);

        if (player1 != null && player2 != null)
        {
            player1.AddResult(player1Result);
            player2.AddResult(player2Result);
        }
    }

    public void DisplayResults()
    {
        var sortedPlayers = Players.OrderByDescending(p => p.Points).ToList();
        Console.WriteLine("Турнирная таблица по шахматам:");
        Console.WriteLine("Место\tФамилия\t\tОчки");
        for (int i = 0; i < sortedPlayers.Count; i++)
        {
            var player = sortedPlayers[i];
            Console.WriteLine($"{i +
1}\t{player.LastName}\t\t{player.Points:F2}");
```

```
        }
    }
}

// Задание 8: Сборная по хоккею
class HockeyPlayerCustom
{
    public string LastName { get; set; }
    public List<int> PenaltyTimes { get; set; } = new();

    public HockeyPlayerCustom(string lastName)
    {
        LastName = lastName;
    }

    public void AddPenaltyTime(int penaltyTime)
    {
        PenaltyTimes.Add(penaltyTime);
    }

    public int TotalPenaltyTime => PenaltyTimes.Sum();
}

class HockeyTeamSelectionCustom
{
    public List<HockeyPlayerCustom> Players { get; set; } = new();

    public void AddPlayer(HockeyPlayerCustom player)
    {
        Players.Add(player);
    }

    public void DisplaySelectedPlayers()
    {
        var eligiblePlayers = Players.Where(p => p.TotalPenaltyTime <
10).OrderBy(p => p.TotalPenaltyTime).ToList();
        Console.WriteLine("Список кандидатов в сборную по хоккею:");
        Console.WriteLine("Фамилия\t\tШтрафное время");
        foreach (var player in eligiblePlayers)
        {
            Console.WriteLine($"{player.LastName}\t\t{player.TotalPenaltyTime}
мин.");
        }
    }
}

// Задание 9: Результаты соревнований фигуристов
class FigureSkaterCustom
{
    public string LastName { get; set; }
    public List<double> JudgeRanks { get; set; } = new();

    public FigureSkaterCustom(string lastName)
    {
        LastName = lastName;
    }

    public void AddJudgeRank(double rank)
    {
        JudgeRanks.Add(rank);
    }
```

```
    public double TotalPlace => JudgeRanks.Sum();
}

class FigureSkatingCompetitionCustom
{
    public List<FigureSkaterCustom> Skaters { get; set; } = new();

    public void AddSkater(FigureSkaterCustom skater)
    {
        Skaters.Add(skater);
    }

    public void DisplayResults()
    {
        var sortedSkaters = Skaters.OrderBy(s => s.TotalPlace).ToList();
        Console.WriteLine("Результаты соревнований фигуристов:");
        Console.WriteLine("Место\tФамилия\t\tСумма мест");
        for (int i = 0; i < sortedSkaters.Count; i++)
        {
            var skater = sortedSkaters[i];
            Console.WriteLine($"{i +
1}\t{skater.LastName}\t\t{skater.TotalPlace:F2}");
        }
    }
}

// 3 Уровень

// Задание 3: Определение победителя среди команд
class TeamMemberCustom
{
    public string Name { get; set; }
    public int Position { get; set; }

    public TeamMemberCustom(string name, int position)
    {
        Name = name;
        Position = position;
    }

    public int GetPoints()
    {
        switch (Position)
        {
            case 1: return 5;
            case 2: return 4;
            case 3: return 3;
            case 4: return 2;
            case 5: return 1;
            default: return 0;
        }
    }
}

class TeamCustom
{
    public string TeamName { get; set; }
    public List<TeamMemberCustom> Members { get; set; } = new();

    public TeamCustom(string teamName)
    {
        TeamName = teamName;
```

```
    }

    public void AddMember(TeamMemberCustom member)
    {
        Members.Add(member);
    }

    public int TotalPoints()
    {
        return Members.Sum(member => member.GetPoints());
    }

    public TeamMemberCustom GetFirstPlaceMember()
    {
        return Members.OrderBy(m => m.Position).First();
    }
}

class TeamCompetitionCustom
{
    public List<TeamCustom> Teams { get; set; } = new();

    public void AddTeam(TeamCustom team)
    {
        Teams.Add(team);
    }

    public void DisplayWinner()
    {
        var sortedTeams = Teams.OrderByDescending(t => t.TotalPoints()).ToList();
        var winningTeam = sortedTeams.First();
        var firstPlaceMember = winningTeam.GetFirstPlaceMember();
        Console.WriteLine($"Победитель: Команда {winningTeam.TeamName}");
        Console.WriteLine($"Первое место: {firstPlaceMember.Name}, очки:
{firstPlaceMember.GetPoints()}");
    }
}

// Задание 4: Лыжные гонки
class SkiRaceParticipantCustom
{
    public string Name { get; set; }
    public int Position { get; set; }

    public SkiRaceParticipantCustom(string name, int position)
    {
        Name = name;
        Position = position;
    }
}

class SkiRaceGroupCustom
{
    public string GroupName { get; set; }
    public List<SkiRaceParticipantCustom> Participants { get; set; } = new();

    public SkiRaceGroupCustom(string groupName)
    {
        GroupName = groupName;
    }

    public void AddParticipant(SkiRaceParticipantCustom participant)
```

```
    {
        Participants.Add(participant);
    }

    public void DisplayResults()
    {
        var sortedParticipants = Participants.OrderBy(p => p.Position).ToList();
        Console.WriteLine($"Результаты {GroupName}:");
        Console.WriteLine("Место\tФамилия");
        foreach (var participant in sortedParticipants)
        {
            Console.WriteLine($"{participant.Position}\t{participant.Name}");
        }
    }
}

class CombinedSkiRaceResultsCustom
{
    public List<SkiRaceParticipantCustom> AllParticipants { get; set; } = new();

    public void AddParticipant(SkiRaceParticipantCustom participant)
    {
        AllParticipants.Add(participant);
    }

    public void DisplayCombinedResults()
    {
        var sortedParticipants = AllParticipants.OrderBy(p =>
p.Position).ToList();
        Console.WriteLine("Общие результаты лыжных гонок:");
        Console.WriteLine("Место\tФамилия");
        foreach (var participant in sortedParticipants)
        {
            Console.WriteLine($"{participant.Position}\t{participant.Name}");
        }
    }
}

// Задание 5: Результаты первенства по футболу
class FootballTeamCustom
{
    public string TeamName { get; set; }
    public int Points { get; set; }
    public int GoalsScored { get; set; }
    public int GoalsConceded { get; set; }

    public FootballTeamCustom(string teamName)
    {
        TeamName = teamName;
        Points = 0;
        GoalsScored = 0;
        GoalsConceded = 0;
    }

    public void AddGameResult(int goalsScored, int goalsConceded)
    {
        GoalsScored += goalsScored;
        GoalsConceded += goalsConceded;
        if (goalsScored > goalsConceded)
        {
            Points += 3; // win
        }
```

```
        else if (goalsScored == goalsConceded)
        {
            Points += 1; // draw
        }
        // No points for a loss
    }

    public int GoalDifference => GoalsScored - GoalsConceded;
}

class FootballLeagueCustom
{
    public List<FootballTeamCustom> Teams { get; set; } = new();

    public void AddTeam(FootballTeamCustom team)
    {
        Teams.Add(team);
    }

    public void DisplayLeagueResults()
    {
        var sortedTeams = Teams.OrderByDescending(t => t.Points)
                               .ThenByDescending(t => t.GoalDifference)
                               .ToList();
        Console.WriteLine("Таблица футбольного первенства:");
        Console.WriteLine("Место\tКоманда\tОчки\tРазница мячей");
        for (int i = 0; i < sortedTeams.Count; i++)
        {
            var team = sortedTeams[i];
            Console.WriteLine($"{i +
1}\t{team.TeamName}\t{team.Points}\t{team.GoalDifference}");
        }
    }
}

// Главный класс программы
class Program
{
    static void Main(string[] args)
    {
        // Задача 1: Соревнования по прыжкам в длину
        Console.WriteLine("Задача 1: Соревнования по прыжкам в длину");
        var longJumpCompetition = new LongJumpCompetitionCustom();
        longJumpCompetition.AddParticipant(new
LongJumpParticipantCustom("Lewandowski", "Spartak", 6.5, 7.1));
        longJumpCompetition.AddParticipant(new LongJumpParticipantCustom("Yamal",
"Dynamo", 7.2, 6.9));
        longJumpCompetition.AddParticipant(new
LongJumpParticipantCustom("Raphinha", "Labor", 6.8, 7.3));
        longJumpCompetition.DisplayResults();

        // Задача 2: Кросс на 500 м для женщин
        Console.WriteLine("\nЗадача 2: Кросс на 500 м для женщин");
        var crossCountryCompetition = new CrossCountryCompetitionCustom();
        double normative = 90.0;
        crossCountryCompetition.AddParticipant(new
CrossCountryParticipantCustom("Lewandowski", "Group 1", "Smirnov", 85.4,
normative));
        crossCountryCompetition.AddParticipant(new
CrossCountryParticipantCustom("Yamal", "Group 2", "Petrov", 92.1, normative));
        crossCountryCompetition.AddParticipant(new
CrossCountryParticipantCustom("Raphinha", "Group 1", "Smirnov", 88.7, normative));
```

```
crossCountryCompetition.DisplayResults(normative);

// Задача 3: Опрос радиокомпании «Человек года»
Console.WriteLine("\nЗадача 3: Опрос радиокомпании 'Человек года'");
var survey = new SurveyCustom();
survey.AddVote("Lewandowski");
survey.AddVote("Yamal");
survey.AddVote("Raphinha");
survey.AddVote("Lewandowski");
survey.AddVote("Raphinha");
survey.AddVote("Lewandowski");
survey.AddVote("Yamal");
survey.AddVote("Yamal");
survey.AddVote("Raphinha");
survey.AddVote("Raphinha");
survey.DisplayTopCandidates(5);

// Задание 7: Турнирная таблица по шахматам
Console.WriteLine("Задача 7: Турнирная таблица по шахматам");
var chessTournament = new ChessTournamentCustom();
chessTournament.AddPlayer(new ChessPlayerCustom("Petrov"));
chessTournament.AddPlayer(new ChessPlayerCustom("Ivanov"));
chessTournament.AddPlayer(new ChessPlayerCustom("Sidorov"));
chessTournament.AddGameResult("Petrov", "Ivanov", 1.0, 0.0);
chessTournament.AddGameResult("Sidorov", "Ivanov", 0.5, 0.5);
chessTournament.AddGameResult("Petrov", "Sidorov", 0.5, 0.5);
chessTournament.DisplayResults();

// Задание 8: Сборная по хоккею
Console.WriteLine("\nЗадача 8: Сборная по хоккею");
var hockeyTeam = new HockeyTeamSelectionCustom();
hockeyTeam.AddPlayer(new HockeyPlayerCustom("Petrov"));
hockeyTeam.AddPlayer(new HockeyPlayerCustom("Ivanov"));
hockeyTeam.AddPlayer(new HockeyPlayerCustom("Sidorov"));
hockeyTeam.AddPlayer(new HockeyPlayerCustom("Smirnov"));
hockeyTeam.Players[0].AddPenaltyTime(5);
hockeyTeam.Players[1].AddPenaltyTime(10);
hockeyTeam.Players[2].AddPenaltyTime(2);
hockeyTeam.Players[3].AddPenaltyTime(8);
hockeyTeam.DisplaySelectedPlayers();

// Задание 9: Результаты соревнований фигуристов
Console.WriteLine("\nЗадача 9: Результаты соревнований фигуристов");
var figureSkatingCompetition = new FigureSkatingCompetitionCustom();
figureSkatingCompetition.AddSkater(new FigureSkaterCustom("Petrov"));
figureSkatingCompetition.AddSkater(new FigureSkaterCustom("Ivanov"));
figureSkatingCompetition.AddSkater(new FigureSkaterCustom("Sidorov"));
figureSkatingCompetition.Skaters[0].AddJudgeRank(2.0);
figureSkatingCompetition.Skaters[0].AddJudgeRank(3.0);
figureSkatingCompetition.Skaters[1].AddJudgeRank(1.0);
figureSkatingCompetition.Skaters[1].AddJudgeRank(4.0);
figureSkatingCompetition.Skaters[2].AddJudgeRank(5.0);
figureSkatingCompetition.Skaters[2].AddJudgeRank(2.0);
figureSkatingCompetition.DisplayResults();

// Задание 3: Определение победителя среди команд
Console.WriteLine("Задача 3: Определение победителя среди команд");
var teamCompetition = new TeamCompetitionCustom();

var teamA = new TeamCustom("Команда А");
teamA.AddMember(new TeamMemberCustom("Игрок 1", 1));
teamA.AddMember(new TeamMemberCustom("Игрок 2", 3));
```

```
teamA.AddMember(new TeamMemberCustom("Игрок 3", 6));
teamA.AddMember(new TeamMemberCustom("Игрок 4", 8));
teamA.AddMember(new TeamMemberCustom("Игрок 5", 12));
teamA.AddMember(new TeamMemberCustom("Игрок 6", 18));

var teamB = new TeamCustom("Команда B");
teamB.AddMember(new TeamMemberCustom("Игрок 1", 2));
teamB.AddMember(new TeamMemberCustom("Игрок 2", 4));
teamB.AddMember(new TeamMemberCustom("Игрок 3", 5));
teamB.AddMember(new TeamMemberCustom("Игрок 4", 7));
teamB.AddMember(new TeamMemberCustom("Игрок 5", 9));
teamB.AddMember(new TeamMemberCustom("Игрок 6", 11));

var teamC = new TeamCustom("Команда C");
teamC.AddMember(new TeamMemberCustom("Игрок 1", 10));
teamC.AddMember(new TeamMemberCustom("Игрок 2", 13));
teamC.AddMember(new TeamMemberCustom("Игрок 3", 14));
teamC.AddMember(new TeamMemberCustom("Игрок 4", 15));
teamC.AddMember(new TeamMemberCustom("Игрок 5", 16));
teamC.AddMember(new TeamMemberCustom("Игрок 6", 17));

teamCompetition.AddTeam(teamA);
teamCompetition.AddTeam(teamB);
teamCompetition.AddTeam(teamC);

teamCompetition.DisplayWinner();

// Задание 4: Лыжные гонки
Console.WriteLine("\nЗадача 4: Лыжные гонки");
var skiGroupA = new SkiRaceGroupCustom("Группа A");
skiGroupA.AddParticipant(new SkiRaceParticipantCustom("Иванов", 1));
skiGroupA.AddParticipant(new SkiRaceParticipantCustom("Петров", 3));
skiGroupA.AddParticipant(new SkiRaceParticipantCustom("Смирнов", 2));

var skiGroupB = new SkiRaceGroupCustom("Группа B");
skiGroupB.AddParticipant(new SkiRaceParticipantCustom("Кузнецов", 1));
skiGroupB.AddParticipant(new SkiRaceParticipantCustom("Михайлов", 3));
skiGroupB.AddParticipant(new SkiRaceParticipantCustom("Федоров", 2));

var combinedResults = new CombinedSkiRaceResultsCustom();
foreach (var participant in skiGroupA.Participants)
{
    combinedResults.AddParticipant(participant);
}

foreach (var participant in skiGroupB.Participants)
{
    combinedResults.AddParticipant(participant);
}

skiGroupA.DisplayResults();
skiGroupB.DisplayResults();
combinedResults.DisplayCombinedResults();

// Задание 5: Результаты первенства по футболу
Console.WriteLine("\nЗадача 5: Результаты первенства по футболу");
var footballLeague = new FootballLeagueCustom();

var teamX = new FootballTeamCustom("Команда X");
teamX.AddGameResult(3, 1); // Победа
teamX.AddGameResult(2, 2); // Ничья
```

```
        var teamY = new FootballTeamCustom("Команда Y");
        teamY.AddGameResult(1, 3); // Проигрыш
        teamY.AddGameResult(2, 1); // Победа

        var teamZ = new FootballTeamCustom("Команда Z");
        teamZ.AddGameResult(0, 0); // Ничья
        teamZ.AddGameResult(1, 1); // Ничья

        footballLeague.AddTeam(teamX);
        footballLeague.AddTeam(teamY);
        footballLeague.AddTeam(teamZ);

        footballLeague.DisplayLeagueResults();
    }
}
```

Результат выполнения программы:

```
Задача 1: Соревнования по прыжкам в длину
Результаты соревнований по прыжкам в длину:
Место   Фамилия         Общество        Попытка 1       Попытка 2       Сумма
1       Yamal           Dynamo          7,20            6,90            14,10
2       Raphinha                Labor           6,80            7,30            14,10
3       Lewandowski             Spartak         6,50            7,10            13,60

Задача 2: Кросс на 500 м для женщин
Результаты кросса на 500 м для женщин:
Фамилия         Группа          Преподаватель   Результат       Норматив
Lewandowski             Group 1 Smirnov         85,4            Выполнен
Raphinha                Group 1 Smirnov         88,7            Выполнен
Yamal           Group 2 Petrov          92,1            Не выполнен
Всего участниц, выполнивших норматив: 2

Задача 3: Опрос радиокомпании 'Человек года'
Результаты опроса 'Человек года':
Кандидат        Голосов         Доля (%)
Raphinha                4               40,00
Lewandowski             3               30,00
Yamal           3               30,00
Задача 7: Турнирная таблица по шахматам
Турнирная таблица по шахматам:
Место   Фамилия         Очки
1       Petrov          1,50
2       Sidorov         1,00
3       Ivanov          0,50

Задача 8: Сборная по хоккею
Список кандидатов в сборную по хоккею:
Фамилия         Штрафное время
Sidorov         2 мин.
Petrov          5 мин.
Smirnov         8 мин.
```

```
Задача 9: Результаты соревнований фигуристов
Результаты соревнований фигуристов:
Место   Фамилия         Сумма мест
1       Petrov          5,00
2       Ivanov          5,00
3       Sidorov         7,00
Задача 3: Определение победителя среди команд
Победитель: Команда Команда A
Первое место: Игрок 1, очки: 5

Задача 4: Лыжные гонки
Результаты Группа A:
Место   Фамилия
1       Иванов
2       Смирнов
3       Петров
Результаты Группа B:
Место   Фамилия
1       Кузнецов
2       Федоров
3       Михайлов
Общие результаты лыжных гонок:
Место   Фамилия
1       Иванов
1       Кузнецов
2       Смирнов
2       Федоров
3       Петров
3       Михайлов

Задача 5: Результаты первенства по футболу
Таблица футбольного первенства:
Место   Команда Очки    Разница мячей
1       Команда X       4       2
2       Команда Y       3       -1
3       Команда Z       2       0
```

Вывод: в ходе работы были реализованы задачи с использованием классов и ООП. Применение наследования, методов и коллекций позволило структурировать код и упростить его расширение. Программа решает задачи корректно, демонстрируя преимущества ООП: читаемость, повторное использование и масштабируемость.