

«Паттерны объектно-ориентированного проектирования»

1. Цель работы

Главной задачей данной лабораторной работы является ознакомление и практическое применение паттернов проектирования при разработке программного обеспечения.

2. Вариант задания

Вариант 1. Паттерн Abstract Factory.

В качестве второго паттерна был выбран паттерн Façade.

Код программы (консольного приложения) на языке C# приведен в Приложении А.

3. Описание предметной области

Предметная область описывает футбольный мир, в котором присутствуют два популярных футбольных клуба – ФК Спартак Москва и ФК Торпедо. Эти клубы участвуют в различных турнирах, таких как кубок и чемпионат. Недавно они сыграли важный матч за суперкубок, где определялся обладатель этого трофея. Нам необходимо получать информацию о забитых голах, а также о полученных очках в каждом из турниров, чтобы отслеживать прогресс и достижения клубов.

4. Аргументация выбора паттернов

Паттерн Abstract Factory позволяет выбрать конкретную реализацию фабрики из семейства возможных фабрик. В заданной предметной области возможно две фабрики (кубок и чемпионат), а также конкретная реализация каждой фабрики в зависимости от футбольного клуба (Спартак Москва или Торпедо). Данный паттерн позволяет выбрать результаты конкретного клуба в конкретном турнире.

Паттерн Façade скрывает сложную систему классов приводя все вызовы к одному объекту. Данный паттерн помещает вызов нескольких сложных объектов в один объект. Благодаря этому пользователь сразу может получить итоговый результат дерби матча за Супер кубок России вместо создания нескольких объектов и вызова разных методов.

Для класса Log, используемого для логирования всех происходящих действий в программе, используется паттерн Singleton, который ограничивает создание одного экземпляра класса, обеспечивает доступ к его единственному объекту. Конструктор класса приватный.

5. Описание классов

Рассмотрим классы, реализующие интерфейс SpartakMoscow.

Класс champFCSM описывает реализацию фабрики для получения результатов ФК Спартак Москва по чемпионату России. Данный класс состоит из трех методов и не имеет полей. Описание методов данного класса приведено в табл. 1.

Таблица 1. Методы класса champFCSM

Имя	Тип	Описание	Область видимости	Входные параметры	Выходные параметры
fcsmderbi	int	Метод получения числа голов в дерби за супер кубок	public	-	int число голов
fcsmGoals	int	Метод получения числа голов в чемпионате	public	-	int число голов
fcsmPoints	int	Метод получения количества очков в чемпионате	public	-	int кол-во очков

Класс supFCSM описывает реализацию фабрики для получения результатов ФК Спартак Москва по кубку России. Данный класс состоит из трех методов и не имеет полей. Описание методов данного класса приведено в табл. 2.

Таблица 2. Методы класса supFCSM

Имя	Тип	Описание	Область видимости	Входные параметры	Выходные параметры
fcsmderbi	int	Метод получения числа голов в дерби за супер кубок	public	-	int число голов
fcsmGoals	int	Метод получения числа голов в кубке	public	-	int число голов
fcsmPoints	int	Метод получения количества очков в кубке	public	-	int кол-во очков

Рассмотрим классы, реализующие интерфейс Torpedo.

Класс champTOR описывает реализацию фабрики для получения результатов ФК Торпедо по чемпионату России. Данный класс состоит из трех методов и не имеет полей. Описание методов данного класса приведено в табл. 3.

Таблица 3. Методы класса champTOR

Имя	Тип	Описание	Область видимости	Входные параметры	Выходные параметры
torDerbi	int	Метод получения числа голов в дерби за супер кубок	public	-	int число голов
torGoals	int	Метод получения числа голов в чемпионате	public	-	int число голов
torPoints	int	Метод получения количества очков в чемпионате	public	-	int кол-во очков

Класс supTOR описывает реализацию фабрики для получения результатов ФК Торпедо по кубку России. Данный класс состоит из трех методов и не имеет полей. Описание методов данного класса приведено в табл. 4.

Таблица 4. Методы класса supTOR

Имя	Тип	Описание	Область видимости	Входные параметры	Выходные параметры
torDerbi	int	Метод получения числа голов в дерби за супер кубок	public	-	int число голов
torGoals	int	Метод получения числа голов в кубке	public	-	int число голов
torPoints	int	Метод получения количества очков в кубке	public	-	int кол-во очков

Рассмотрим классы, реализующие интерфейс AbstractFactory.

Класс champFactory описывает фабрику для получения результатов по чемпионату России. Данный класс состоит из двух методов и не имеет полей. Описание методов данного класса приведено в табл. 5.

Таблица 5. Методы класса champFactory

Имя	Тип	Описание	Область видимости	Входные параметры	Выходные параметры
getFCSM	SpartakMoscow	Реализация фабрики для Спартака	public	-	Экземпляр класса champFCSM
getTOR	Torpedo	Реализация фабрики для Торпедо	public	-	Экземпляр класса champTOR

Класс supFactory описывает фабрику для получения результатов по кубку России. Данный класс состоит из двух методов и не имеет полей. Описание методов данного класса приведено в табл. 6.

Таблица 6. Методы класса supFactory

Имя	Тип	Описание	Область видимости	Входные параметры	Выходные параметры
getFCSM	SpartakMoscow	Реализация фабрики для Спартака	public	-	Экземпляр класса supFCSM
getTOR	Torpedo	Реализация фабрики для Торпедо	public	-	Экземпляр класса supTOR

Далее рассмотрим классы, которые не реализовывали никакой из интерфейсов.

Класс Façade описывает паттерн Façade, с помощью которого пользователь может получить информацию о результате дерби матча за супер кубок России. Данный класс состоит из двух полей и двух методов. Описание полей и методов данного класса приведено в табл. 7.

Таблица 7. Поля и методы класса Façade

Имя	Тип	Описание	Область видимости	Входные параметры	Выходные параметры
Поля					
spartak	SpartakMoscow	Экземпляр класса cupFCSM	private	-	-
torpedo	Torpedo	Экземпляр класса cupTOR	private	-	-
Методы					
Facade	-	Конструктор класса	public	-	-
derbi	void	Выводит в консоль результат дерби матча за супер кубок России	public	-	-

Класс Log используется для логирования всех происходящих действий в программе. Данный класс состоит из трех методов и одного поля. Описание полей и методов данного класса приведено в табл. 8.

Таблица 8. Поля и методы класса Log

Имя	Тип	Описание	Область видимости	Входные параметры	Выходные параметры
Поля					
goals_team1	int	Количество забитых голов первой командой	private	-	-
Методы					
Log	-	Конструктор класса	private	-	-
getLog	Log	Создает только один экземпляр класса.	public	-	Экземпляр класса Log -
AddLog	void	Записывает строку в лог файл	public	string msg – сообщение для записи в лог файл	-

6. Диаграмма классов

Диаграмма классов представлена на рис. 1.

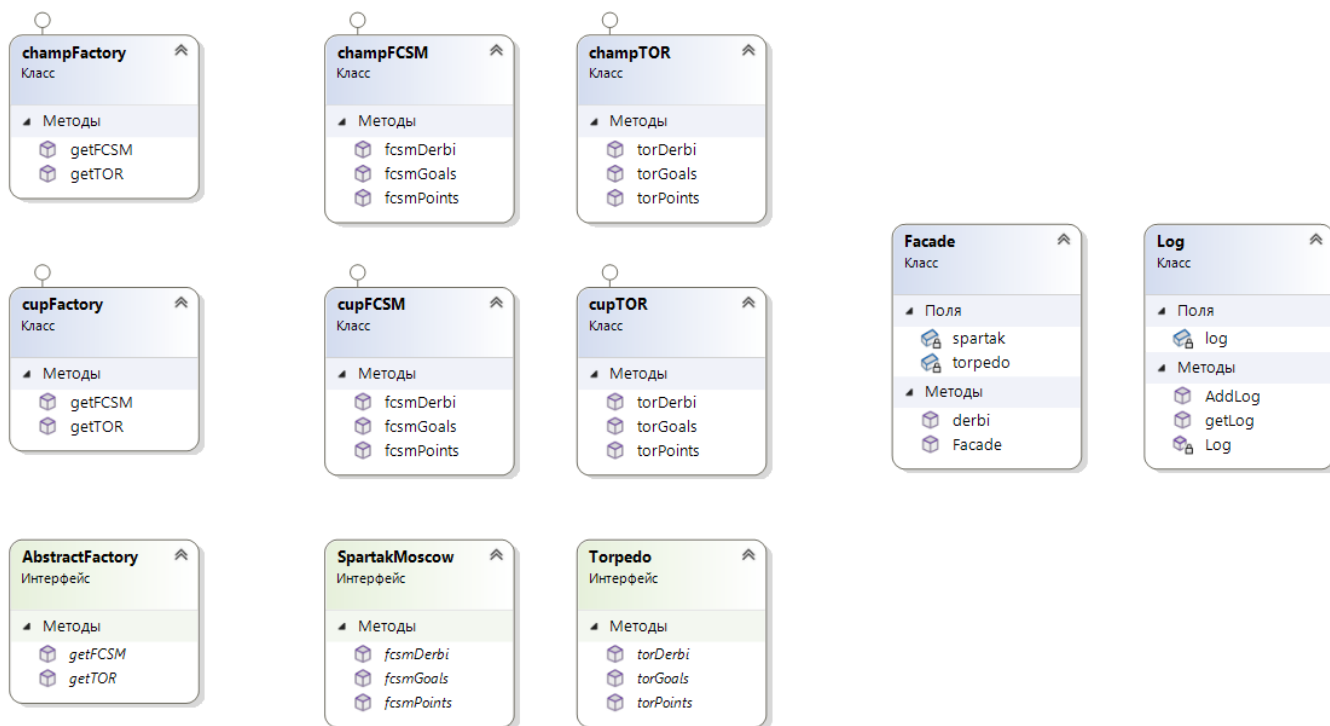


Рис. 1. Диаграмма классов

7. Результат работы программы

Результат работы программы представлен на рис. 2 – 6.

Выберите турнир:
1 - Кубок России
2 - Чемпионат России

Рис. 2. Начальное меню

После выбора первого пункта начального меню пользователь переходит в меню раздела Кубок России.

D:\Материалы магистратуры\2nd sem\ТРПО\Программирование\ЛР 1.1\

Кубок России

Выберите интересующую информацию:

- 1 - Количество голов ФК Спартак Москва в турнире
- 2 - Количество очков ФК Спартак Москва в турнире
- 3 - Количество голов ФК Торпедо Москва в турнире
- 4 - Количество очков ФК Торпедо Москва в турнире
- 5 - Результат дерби в финале Супер кубка
- 6 - Завершить

Количество голов ФК Спартак Москва в турнире: 15

Выберите интересующую информацию:

- 1 - Количество голов ФК Спартак Москва в турнире
- 2 - Количество очков ФК Спартак Москва в турнире
- 3 - Количество голов ФК Торпедо Москва в турнире
- 4 - Количество очков ФК Торпедо Москва в турнире
- 5 - Результат дерби в финале Супер кубка
- 6 - Завершить

Количество очков ФК Спартак Москва в турнире: 12

Выберите интересующую информацию:

- 1 - Количество голов ФК Спартак Москва в турнире
- 2 - Количество очков ФК Спартак Москва в турнире
- 3 - Количество голов ФК Торпедо Москва в турнире
- 4 - Количество очков ФК Торпедо Москва в турнире
- 5 - Результат дерби в финале Супер кубка
- 6 - Завершить

Количество голов ФК Торпедо Москва в турнире: 7

Выберите интересующую информацию:

- 1 - Количество голов ФК Спартак Москва в турнире
- 2 - Количество очков ФК Спартак Москва в турнире
- 3 - Количество голов ФК Торпедо Москва в турнире
- 4 - Количество очков ФК Торпедо Москва в турнире
- 5 - Результат дерби в финале Супер кубка
- 6 - Завершить

Количество очков ФК Торпедо Москва в турнире: 6

Рис. 3. Выбор 1 - 4 пунктов меню

```
Выберите интересующую информацию:
1 - Количество голов ФК Спартак Москва в турнире
2 - Количество очков ФК Спартак Москва в турнире
3 - Количество голов ФК Торпедо Москва в турнире
4 - Количество очков ФК Торпедо Москва в турнире
5 - Результат дерби в финале Супер кубка
6 - Завершить

Результат дерби в финале Супер кубка России:
Спартак Москва 3 - 1 Торпедо

Выберите интересующую информацию:
1 - Количество голов ФК Спартак Москва в турнире
2 - Количество очков ФК Спартак Москва в турнире
3 - Количество голов ФК Торпедо Москва в турнире
4 - Количество очков ФК Торпедо Москва в турнире
5 - Результат дерби в финале Супер кубка
6 - Завершить

Программа завершила выполнение. Нажмите ENTER...
```

Рис. 4. Выбор 5 и 6 пункта меню

После выбора второго пункта начального меню пользователь переходит в меню раздела Чемпионат России.

■ D:\Материалы магистратуры\2nd sem\ТРПО\Программирование\ЛР 1.1\

Чемпионат России

```
Выберите интересующую информацию:
1 - Количество голов ФК Спартак Москва в турнире
2 - Количество очков ФК Спартак Москва в турнире
3 - Количество голов ФК Торпедо Москва в турнире
4 - Количество очков ФК Торпедо Москва в турнире
5 - Результат дерби в финале Супер кубка
6 - Завершить
```

Количество голов ФК Спартак Москва в турнире: 37

```
Выберите интересующую информацию:
1 - Количество голов ФК Спартак Москва в турнире
2 - Количество очков ФК Спартак Москва в турнире
3 - Количество голов ФК Торпедо Москва в турнире
4 - Количество очков ФК Торпедо Москва в турнире
5 - Результат дерби в финале Супер кубка
6 - Завершить
```

Количество очков ФК Спартак Москва в турнире: 31

```
Выберите интересующую информацию:
1 - Количество голов ФК Спартак Москва в турнире
2 - Количество очков ФК Спартак Москва в турнире
3 - Количество голов ФК Торпедо Москва в турнире
4 - Количество очков ФК Торпедо Москва в турнире
5 - Результат дерби в финале Супер кубка
6 - Завершить
```

Количество голов ФК Торпедо Москва в турнире: 25

```
Выберите интересующую информацию:
1 - Количество голов ФК Спартак Москва в турнире
2 - Количество очков ФК Спартак Москва в турнире
3 - Количество голов ФК Торпедо Москва в турнире
4 - Количество очков ФК Торпедо Москва в турнире
5 - Результат дерби в финале Супер кубка
6 - Завершить
```

Количество очков ФК Торпедо Москва в турнире: 22

Рис. 5. Выбор 1 - 4 пунктов меню

```
Выберите интересующую информацию:
1 - Количество голов ФК Спартак Москва в турнире
2 - Количество очков ФК Спартак Москва в турнире
3 - Количество голов ФК Торпедо Москва в турнире
4 - Количество очков ФК Торпедо Москва в турнире
5 - Результат дерби в финале Супер кубка
6 - Завершить

Результат дерби в финале Супер кубка России:
Спартак Москва 3 - 1 Торпедо

Выберите интересующую информацию:
1 - Количество голов ФК Спартак Москва в турнире
2 - Количество очков ФК Спартак Москва в турнире
3 - Количество голов ФК Торпедо Москва в турнире
4 - Количество очков ФК Торпедо Москва в турнире
5 - Результат дерби в финале Супер кубка
6 - Завершить

Программа завершила выполнение. Нажмите ENTER...
```

Рис. 6. Выбор 5 и 6 пункта меню

Результат логирования программы представлен на рис. 7.

log.txt – Блокнот	log.txt – Блокнот
Файл Правка Формат Вид Справка	Файл Правка Формат Вид Справка
30.03.2024 2:25:16 program started. log created	02.04.2024 2:44:13 program started. log created
30.03.2024 2:25:18 champFactory created	02.04.2024 2:44:39 cupFactory created
30.03.2024 2:25:18 Factory spartak created	02.04.2024 2:44:39 Factory spartak created
30.03.2024 2:25:18 Factory torpedo created	02.04.2024 2:44:39 Factory torpedo created
30.03.2024 2:25:19 method <fcsMGoals> called	02.04.2024 2:44:43 method <fcsMGoals> called
30.03.2024 2:25:19 method <torPoints> called	02.04.2024 2:44:45 method <fcsMPoints> called
30.03.2024 2:25:20 method <torGoals> called	02.04.2024 2:44:48 method <torGoals> called
30.03.2024 2:25:20 method <fcsMPoints> called	02.04.2024 2:44:49 method <torPoints> called
30.03.2024 2:25:21 Facade class created	02.04.2024 2:44:49 Facade class created
30.03.2024 2:25:21 method <derbi> called	02.04.2024 2:44:49 method <derbi> called
30.03.2024 2:25:22 program finished	02.04.2024 2:44:56 program finished
30.03.2024 2:26:42 program started. log created	02.04.2024 2:45:39 program started. log created
30.03.2024 2:26:42 cupFactory created	02.04.2024 2:45:41 cupFactory created
30.03.2024 2:26:43 Factory spartak created	02.04.2024 2:45:41 Factory spartak created
30.03.2024 2:26:43 Factory torpedo created	02.04.2024 2:45:41 Factory torpedo created
30.03.2024 2:26:43 method <torGoals> called	02.04.2024 2:45:42 method <fcsMGoals> called
30.03.2024 2:26:44 method <fcsMPoints> called	02.04.2024 2:45:43 method <fcsMPoints> called
30.03.2024 2:26:44 method <torPoints> called	02.04.2024 2:46:10 method <torGoals> called
30.03.2024 2:26:45 Facade class created	02.04.2024 2:46:11 method <torPoints> called
30.03.2024 2:26:45 method <derbi> called	02.04.2024 2:46:33 Facade class created
30.03.2024 2:26:46 method <fcsMGoals> called	02.04.2024 2:46:33 method <derbi> called
30.03.2024 2:26:47 program finished	02.04.2024 2:46:35 program finished
30.03.2024 2:45:37 program started. log created	02.04.2024 2:51:19 program started. log created
30.03.2024 2:45:43 cupFactory created	02.04.2024 2:51:20 champFactory created
30.03.2024 2:45:43 Factory spartak created	02.04.2024 2:51:20 Factory spartak created
30.03.2024 2:45:43 Factory torpedo created	02.04.2024 2:51:20 Factory torpedo created
30.03.2024 2:45:44 method <torPoints> called	02.04.2024 2:51:22 method <fcsMGoals> called
30.03.2024 2:45:45 program finished	02.04.2024 2:51:23 method <fcsMPoints> called

Рис. 7. Log-файл

8. Вывод

В рамках лабораторной работы были рассмотрены три паттерна проектирования.

1) Abstract Factory. Этот паттерн проектирования полезен, когда нужно создать семейство связанных объектов. Это позволяет выбрать конкретную реализацию фабрики из семейства возможных фабрик.

2) Facade. Паттерн Façade предоставляет унифицированный интерфейс к нескольким объектам, скрывая их сложную структуру и позволяя упростить взаимодействие с ними. Façade может упростить код, скрывая детали взаимодействия между подсистемами и предоставляя более простой интерфейс для работы с ними.

3) Singleton. Паттерн Singleton гарантирует, что у класса есть только один экземпляр, что полезно, когда нужно контролировать количество объектов и управлять их использованием. Singleton обеспечивает глобальную точку доступа к объекту, что может упростить доступ к нему и избежать проблем с многопоточностью.

В целом, работа с паттернами проектирования, такими как Abstract Factory, Facade и Singleton, может помочь разработчикам структурировать и организовать код, улучшить его читаемость и гибкость, а также предотвратить возможные ошибки и сложности в разработке программных систем.

Приложение А. Код программы.

```
// А-08м-23 Белоус Виктор ЛР 1.1
// Номер по списку 9 => Вариант 1

// Используемые паттерны:
// 1) Abstract Factory (по заданию)
// 2) Facade (выбранный)

using System;
using System.IO;

namespace lab1_1
{
    // Определим общие интерфейсы двух Футбольных клубов
    public interface SpartakMoscow
    {
        int fcsMGoals();
        int fcsMPoints();
        int fcsMDerbi();
    }
    public interface Torpedo
    {
        int torGoals();
        int torPoints();
        int torDerbi();
    }
    // =====

    // Кубок России – Спартак Москва
    public class cupFCSM : SpartakMoscow
    {
        public int fcsMGoals()
        {
            int goals = 15;
            return goals;
        }

        public int fcsMPoints()
        {
            int points = 12;
            return points;
        }

        public int fcsMDerbi()
        {
            int goal = 3;
            return goal;
        }
    }

    // Чемпионат России – Спартак Москва
    public class champFCSM : SpartakMoscow
    {
        public int fcsMGoals()
        {
            int goals = 37;
            return goals;
        }
    }
}
```

```

    public int fcsmpoints()
    {
        int points = 31;
        return points;
    }

    public int fcsmderbi()
    {
        int goal = 3;
        return goal;
    }
}
// =====

// Кубок России – Торпедо
public class cupTOR : Torpedo
{
    public int torGoals()
    {
        int goals = 7;
        return goals;
    }

    public int torPoints()
    {
        int points = 6;
        return points;
    }

    public int torDerbi()
    {
        int goal = 1;
        return goal;
    }
}

// Чемпионат России – Торпедо
public class champTOR : Torpedo
{
    public int torGoals()
    {
        int goals = 25;
        return goals;
    }

    public int torPoints()
    {
        int points = 22;
        return points;
    }

    public int torDerbi()
    {
        int goal = 1;
        return goal;
    }
}
// =====

interface AbstractFactory
{

```

```

        SpartakMoscow getFCSM();
        Torpedo getTOR();
    }

// =====

// Чемпионат России
public class champFactory : AbstractFactory
{
    public SpartakMoscow getFCSM()
    {
        return new champFCSM();
    }

    public Torpedo getTOR()
    {
        return new champTOR();
    }
}

// Кубок России
public class cupFactory : AbstractFactory
{
    public SpartakMoscow getFCSM()
    {
        return new cupFCSM();
    }

    public Torpedo getTOR()
    {
        return new cupTOR();
    }
}

// =====
// Паттерн facade
class Facade
{
    private SpartakMoscow spartak;
    private Torpedo torpedo;
    public Facade()
    {
        spartak = new cupFCSM();
        torpedo = new cupTOR();
    }
    public void derbi()
    {
        Console.WriteLine("Спартак Москва {0} - {1} Торпедо",
spartak.fcsMDerbi(), torpedo.torDerbi());
    }
}

// =====
// класс для логирования программы
class Log
{
    private static Log log;
    private Log() { }
}

```

```

public static Log getLog()
{
    if (log == null)
        log = new Log();
    return log;
}
public async void AddLog(string msg)
{
    string path = "log.txt";
    using (StreamWriter writer = new StreamWriter(path, true))
    {
        await writer.WriteLineAsync(DateTime.Now.ToString() + " " + msg);
    }
}
}
// =====

// Основная программа
internal class Program
{
    static void Main(string[] args)
    {
        var logger = Log.getLog();
        logger.AddLog("program started. log created");
        bool Flag = true;
        AbstractFactory ifactory = null;

        while (Flag)
        {
            Console.WriteLine("Выберите турнир: ");
            Console.WriteLine("1 - Кубок России ");
            Console.WriteLine("2 - Чемпионат России ");
            var start = Console.ReadLine();

            if (start.Equals("1"))
            {
                ifactory = new cupFactory();
                logger.AddLog("cupFactory created");
                Flag = false;
                Console.Clear();
                Console.WriteLine("Кубок России");
            }
            else if (start.Equals("2"))
            {
                ifactory = new champFactory();
                logger.AddLog("champFactory created");
                Flag = false;
                Console.Clear();
                Console.WriteLine("Чемпионат России");
            }
            else
            {
                Console.WriteLine("\nОшибка ввода\n");
            }
        }

        SpartakMoscow spartak = ifactory.getFCSM();
        logger.AddLog("Factory spartak created");
        Torpedo torpedo = ifactory.getTOR();
        logger.AddLog("Factory torpedo created");
    }
}

```

```

Flag = true;
while (Flag)
{
    Console.WriteLine("\nВыберите интересующую информацию: ");
    Console.WriteLine("1 - Количество голов ФК Спартак Москва в
турнире ");
    Console.WriteLine("2 - Количество очков ФК Спартак Москва в
турнире ");
    Console.WriteLine("3 - Количество голов ФК Торпедо Москва в
турнире ");
    Console.WriteLine("4 - Количество очков ФК Торпедо Москва в
турнире ");
    Console.WriteLine("5 - Результат дерби в финале Супер кубка");
    Console.WriteLine("6 - Завершить");

    var res = Console.ReadLine();          // выбранный пункт меню

    // удаление из консоли выбранного пункта для красивого
отображения
    // -----
    // Capture current cursor position
    var cursorTop = Console.CursorTop;
    var cursorLeft = Console.CursorLeft;

    // Clear the previous line (above the current position)
    Console.SetCursorPosition(0, cursorTop - 1);
    Console.Write(new string(' ', Console.BufferWidth));

    // Resume cursor at it's original position
    Console.SetCursorPosition(cursorLeft, cursorTop);
    // -----

    switch (res)
    {
        case "1":
            Console.WriteLine("Количество голов ФК Спартак Москва в
турнире: {0}", spartak.fcsmGoals().ToString());
            logger.AddLog("method <fcsmGoals>      called");
            break;
        case "2":
            Console.WriteLine("Количество очков ФК Спартак Москва в
турнире: {0}", spartak.fcsmPoints().ToString());
            logger.AddLog("method <fcsmPoints>      called");
            break;
        case "3":
            Console.WriteLine("Количество голов ФК Торпедо Москва в
турнире: {0}", torpedo.torGoals().ToString());
            logger.AddLog("method <torGoals>      called");
            break;
        case "4":
            Console.WriteLine("Количество очков ФК Торпедо Москва в
турнире: {0}", torpedo.torPoints().ToString());
            logger.AddLog("method <torPoints>      called");
            break;
        case "5":
            Console.WriteLine("Результат дерби в финале Супер кубка
России: ");
            Facade facade = new Facade();
            logger.AddLog("Facade class created");

```

```

        facade.derbi();
        logger.AddLog("method <derbi> called");
        break;
    case "6":
        Console.WriteLine("Программа завершила выполнение.
Нажмите ENTER...");
        logger.AddLog("program finished\n");
        Flag = false;
        break;
    default:
        Console.WriteLine("\nОшибка ввода\n");
        break;
    }
}
Console.ReadLine();
}
}
}

```