Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет

информатики и радиоэлектроники»

Факультет компьютерного проектирования

Кафедра Информатики

Дисциплина «Программирование»

**ОТЧЕТ**

к лабораторной работе №4

на тему:

**«Конструкторы. Статические члены класса.**

**Шаблон проектирования Singleton.»**

БГУИР 6-05-0612-02 36

|  |
| --- |
| Выполнил студент группы 353505  ДАНИЛОВ Дмитрий Игоревич |
|  |
| (дата, подпись студента) |
| Проверил ассистент каф. Информатики  РОМАНЮК Максим Валерьевич |
|  |
| (дата, подпись преподавателя) |

Минск 2024

# 1 Индивидуальное задание

Задание 1. Вариант 8.Спроектировать классы для выбранной предметной области: класс-контейнер, управляющий контейнеризируемым классом, и контейнеризируемый класс. Предусмотреть у классов наличие полей, методов и свойств. Предусмотреть перегрузку методов. Предусмотреть наличие статических методов. Применить к классу-контейнеру шаблон проектирования Singleton. Предметная область: Предметная область: Гостиница - Тариф. В классе хранить информацию о названии гостиницы, числе заселенных мест, общем числе мест, оплате за день проживания (для всех жильцов одинаковая стоимость – класс Тариф). Реализовать метод для подсчета общей выручки гостиницы. Реализовать возможность изменения (увеличения и уменьшения) тарифа.

# 2 Выполнение работы

Для выполнения данной лабораторной работы первым делом был создан проект, к которому позже добавлялись классы, реализующие данное задание. Сначала был реализован контейнеризируемый класс “Tariff”. В нем было создано статическое поле “instance”, хранящее информацию о названии гостиницы, числе заселенных мест, общем числе мест, оплате за день проживания. Для поля “instance” был применен модификатор static, так как для всех жильцов цена одинаковая. Были реализованы 2 конструктора: базовый и устанавливающий значение обоих полей. Так же для обоих полей были созданы свойства, которые позволяют получать и устанавливать значение полей.

public class Tariff

{

public decimal DailyRate { get; private set; }

public Tariff(decimal rate)

{

SetDailyRate(rate);

}

public void SetDailyRate(decimal rate)

{

DailyRate = rate;

}

public decimal CalculateTotalRevenue(int occupiedRooms)

{

return DailyRate \* occupiedRooms;

}

}

После этого был создан класс-контейнер с названием “Hotel”. Он содержит поля “instance”, “hotelName”, “availableRooms”, “occupiedRooms”, “totalRooms”, “tariff”. Поле “hotel” является объектом класса “Hotel”. Так же для реализации шаблона Singleton был создан приватный конструктор класса, инициализирующий все поля, и пустой объект класса. Для применения этого конструктора был создан статический метод “GetInstance”, который проверяет, созданный в классе объект на “ null ”. В случае, если поле “null”, то создается новый объект класса и возвращается ссылка на него. Иначе – только возвращается ссылка. То есть можно создать только 1 независимый объект класса, а все остальные лишь будут указывать на ту же область памяти. Для изменения стоимости тарифа было создано свойство “SetDailyRate”, которое обращается к соответствующему свойству класса “Hotel”. Также были реализованы методы для получения информации о количестве комнат и заселившихся людей, также был добавлен метод, который позволяет выселять либо же, наоборот, заселять людей.

public class Hotel

{

private static Hotel instance;

private string hotelName;

private int availableRooms;

private int occupiedRooms;

private int totalRooms;

private Tariff tariff;

private Hotel(string name, int totalRooms, decimal dailyRate)

{

hotelName = name;

this.totalRooms = totalRooms;

availableRooms = totalRooms;

occupiedRooms = 0;

tariff = new Tariff(dailyRate);

}

public static Hotel GetInstance()

{

if (instance == null)

{

Console.Write("Введите название гостиницы: ");

string name = Console.ReadLine();

int totalRooms;

while (true)

{

Console.Write("Введите общее количество комнат: ");

if (int.TryParse(Console.ReadLine(), out totalRooms) && totalRooms > 0)

break;

else

Console.WriteLine("Некорректный ввод. Пожалуйста, введите положительное целое число.");

}

int numberOfGuests;

while (true)

{

Console.Write("Введите количество гостей для заселения: ");

if (int.TryParse(Console.ReadLine(), out numberOfGuests) && numberOfGuests > 0 && numberOfGuests <= totalRooms)

break;

else if (numberOfGuests > totalRooms)

Console.WriteLine("Недостаточное количество комнат. Пожалуйста, введите другое количество гостей.");

else

Console.WriteLine("Некорректный ввод. Пожалуйста, введите положительное целое число.");

}

decimal dailyRate;

while (true)

{

Console.Write("Введите стоимость проживания за день: ");

if (decimal.TryParse(Console.ReadLine(), out dailyRate) && dailyRate > 0)

break;

else

Console.WriteLine("Некорректный ввод. Пожалуйста, введите положительное число.");

}

instance = new Hotel(name, totalRooms, dailyRate);

instance.CheckIn(numberOfGuests);

}

return instance;

}

public decimal CalculateTotalRevenue()

{

return tariff.CalculateTotalRevenue(occupiedRooms);

}

public void ChangeTariff(decimal newRate)

{

tariff.SetDailyRate(newRate);

}

public void CheckIn(int numberOfGuests)

{

if (numberOfGuests <= availableRooms)

{

occupiedRooms += numberOfGuests;

availableRooms -= numberOfGuests;

Console.WriteLine($"Успешное заселение {numberOfGuests} гостей.");

}

else

{

Console.WriteLine("Извините, нет достаточного количества свободных комнат.");

}

}

public void CheckOut(int numberOfGuests)

{

if (numberOfGuests <= occupiedRooms)

{

occupiedRooms -= numberOfGuests;

availableRooms += numberOfGuests;

Console.WriteLine($"Выселение {numberOfGuests} гостей прошло успешно.");

}

else

{

Console.WriteLine("Недостаточное количество заселенных гостей.");

}

}

public int GetTotalRooms()

{

return totalRooms;

}

public int GetOccupiedRooms()

{

return occupiedRooms;

}

public int GetNumberOfGuests()

{

return occupiedRooms;

}

}

Основной код программы запрашивает у пользователя данные о названии гостиницы, общем числе комнат и количестве гостей, также стоимость проживания за день. (рисунок 1).

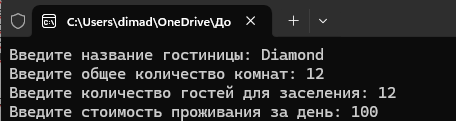


Рисунок 1 – запрос программой входных данных

После этого программа предлагает пользователю меню, в котором можно выбрать один из 7 пунктов меню (пользователь может заселить либо выселить гостя, узнать информацию о количестве гостей и комнат, также узнать общую выручку и, в случае необходимости, поменять стоимость проживания). (рисунок 2).

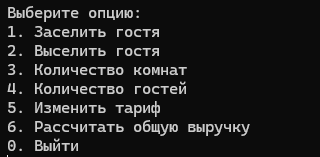


Рисунок 2 – меню выбора представленных опций

# Вывод

В ходе лабораторной работы я углубил свои знания в теми создания классов. Научился создавать конструкторы и описывать свойства классов. Изучил основу работы с шаблоном проектирования Singleton.