

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра информационной безопасности

ОТЧЕТ
по лабораторной работе №9
по дисциплине «Объектно-ориентированное
программирование»
Тема: «Использование наследования с применением
интерфейсов»

Студент гр. 3363

Минко Д.А.

Преподаватель:

Новакова Н.Е.

Санкт-Петербург

2024

1. Формулировка задания

Вариант 22

Разработать программу для обеспечения продажи туров.

2. Цель работы

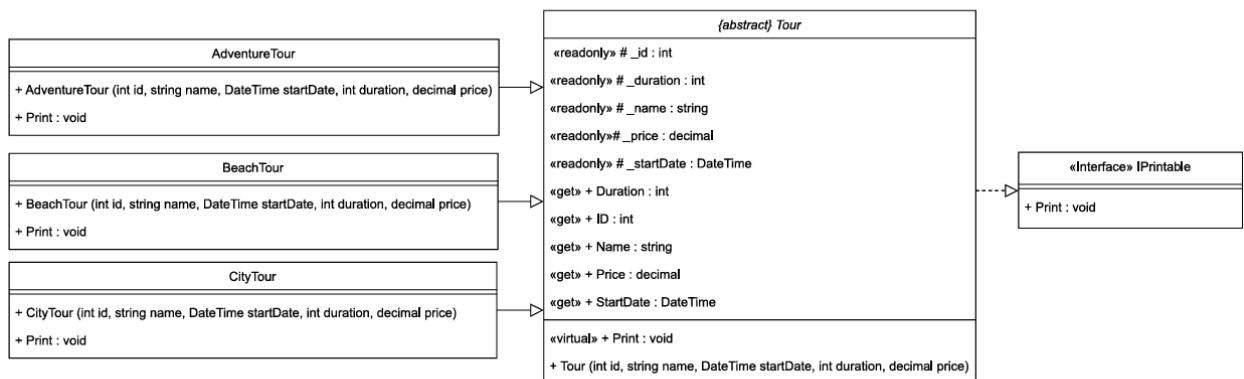
Целью данной лабораторной работы является изучение механизмов наследования классов и реализации интерфейсов в языке C#; изучение концепции абстрактных классов. Во время выполнения лабораторной работы необходимо разработать структуру данных для предметной области (предметной областью данного варианта являются туры), а также на основе этих структур написать программу, которая будет обеспечивать продажу туров.

3. Теоретический аспект задачи

Перед реализацией программы необходимо теоретически определить структуру. Для этого принято решение создать абстрактный класс «Tour», который будет представлять общую модель тура. Этот класс будет содержать свойства, определяющие тур как сущность: его идентификационный номер, название тура, дата начала, продолжительность тура и его цена. Кроме того, класс «Tour» будет реализовывать метод «Print», который будет выводить информацию о туре в презентабельном виде, а также реализует интерфейс «IPrintable».

Далее будут созданы три наследующих класса от «Tour», каждый из которых будет представлять отдельный тип тура (пляжный тур, городской тур, приключенческий тур). Эти классы будут содержать дополнительную информацию о типе тура, а метод «Print» будет переопределен для каждого из них. В каждом из этих классов будет добавлено статическое поле, представляющее цену тура для конкретного типа, и это поле будет иметь уникальное значение для каждого типа тура.

4. Формализация задачи



Основой программы является абстрактный класс «Tour», который описывает общую модель тура. Поля этого класса включают в себя основные свойства тура (идентификационный номер, название, дата начала, продолжительность и цена). Конструктор этого класса инициализирует эти поля значениями, и они могут быть доступны только для чтения.

Кроме того, в классе «Tour» есть метод «Print», который реализует интерфейс «IPrintable» и выводит информацию о туре в презентабельном виде. Также реализована логика для определения типа тура через методы классов-наследников.

Для конкретных типов туров (пляжный тур, городской тур, приключенческий тур) создаются производные классы «BeachTour», «CityTour» и «AdventureTour». Эти классы содержат статическое поле «Price», которое задает стоимость тура для каждого типа. Конструктор каждого из этих классов передает параметры в конструктор базового класса, поскольку все эти параметры одинаковы для всех типов туров. Метод «Print» в каждом из классов перегружается, чтобы дополнительно выводить цену тура, после вызова метода «Print» базового класса.

Программа выводит информацию о турах, включая их цену, и позволяет пользователю легко различать типы туров, которые предлагаются для продажи.

5. Спецификация классов

Имя	Тип	Модификатор доступа	Назначение
_id	int	protected	Идентификационный номер тура
_name	string	protected	Название тура
_startDate	DateTime	protected	Дата начала тура
_duration	int	protected	Продолжительность тура (в днях)
_price	decimal	protected	Цена тура

Таблица 1 – Описание полей класса Tour

Метод	Возвращаемый тип	Модификатор доступа	Входные параметры	Выходные параметры	Назначение
Tour	-	public	int id, string name, DateTime startDate, int duration, decimal price	-	Конструктор класса для инициализации полей
Print	void	public	-	-	Вывод информации о туре

Таблица 2 – Описание методов класса Tour

Имя	Тип	Модификатор доступа	Назначение
_price	decimal	private	Цена пляжного тура

Таблица 3 – Описание полей класса BeachTour

Метод	Возвращаемый тип	Модификатор доступа	Входные параметры	Выходные параметры	Назначение
BeachTour	-	public	int id, string name, DateTime startDate, int duration, decimal price	-	Конструктор для инициализации полей класса
print	void	public	-	-	Переопределение метода для вывода информации о пляжном туре

Таблица 4 – Описание методов класса BeachTour

Имя	Тип	Модификатор доступа	Назначение
_price	decimal	private	Цена городского тура

Таблица 5 – Описание полей класса CityTour

Метод	Возвращаемый тип	Модификатор доступа	Входные параметры	Выходные параметры	Назначение
CityTour	-	public	int id, string name, DateTime startDate, int duration, decimal price	-	Конструктор для инициализации полей класса
Print	void	public	-	-	Переопределение метода для вывода информации о городском туре

Таблица 6 – Описание методов класса CityTour

Имя	Тип	Модификатор доступа	Назначение
_price	decimal	private	Цена приключенческого тура

Таблица 7 – Описание полей класса AdventureTour

Метод	Возвращаемый тип	Модификатор доступа	Входные параметры	Выходные параметры	Назначение
AdventureTour	-	public	int id, string name, DateTime startDate, int duration, decimal price	-	Конструктор для инициализации полей класса
Print	void	public	-	-	Переопределение метода для вывода информации о приключенческом туре

Таблица 8 – Описание методов класса AdventureTour

6. Руководство оператора

Никаких входных данных от пользователя не требуется. Вся работа программы заранее определена.

7. Руководство программиста

Все логически отдельные структуры находятся в отдельных файлах. Название файлов интерфейсов начинается с буквы I. Для получения данных о полях экземпляров классов извне следует использовать реализованные

геттеры. Также поля используют стереотип «только для чтения», поэтому присваивание значений новому экземпляру извне происходит лишь однажды при инициализации. Класс `Tour` является абстрактным, поэтому создание экземпляра класса `Tour` является недопустимым.

8. Контрольный пример

Далее представлены результаты работы программы (Рисунок 1).



```
Консоль отладки Microsoft V × +  
  
Tour ID: 1  
Name: Maldives Getaway  
Start Date: 01.12.2024  
Duration: 7 days  
Price: 49999,99 P  
Type: Beach Tour  
  
Tour ID: 2  
Name: Paris Explorer  
Start Date: 15.05.2024  
Duration: 5 days  
Price: 34999,99 P  
Type: City Tour  
  
Tour ID: 3  
Name: Amazon Jungle Trek  
Start Date: 20.08.2024  
Duration: 10 days  
Price: 59999,99 P  
Type: Adventure Tour
```

Рисунок 1 - Контрольный пример для работы

В методе «Main» создаются три различных тура («BeachTour», «CityTour», «AdventureTour»), и через конструктора им присваиваются свои данные. Затем для каждого объекта по очереди вызывается метод Print, который выводит на экран пользователя информацию о созданных турах. Информация включает идентификационный номер тура, название, дату начала, продолжительность тура, цену и тип тура (например, «Beach Tour», «City Tour», «Adventure Tour»).

9. Листинг программы

Program.cs

```
namespace TourSales
{
    internal class Program
    {
        static void Main(string[] args)
        {
            BeachTour beachTour = new BeachTour(1, "Maldives Getaway", new
DateTime(2024, 12, 1), 7, 49999.99M);
            CityTour cityTour = new CityTour(2, "Paris Explorer", new
DateTime(2024, 5, 15), 5, 34999.99M);
            AdventureTour adventureTour = new AdventureTour(3, "Amazon Jungle
Trek", new DateTime(2024, 8, 20), 10, 59999.99M);

            beachTour.Print();
            Console.WriteLine();
            cityTour.Print();
            Console.WriteLine();
            adventureTour.Print();
        }
    }
}
```

IPrintable.cs

```
namespace TourSales
{
    internal interface IPrintable
    {
        void Print();
    }
}
```

Tour.cs

```
namespace TourSales
{
    internal abstract class Tour : IPrintable
    {
        protected readonly int _id; // ID тура
        protected readonly string _name; // Название тура
        protected readonly DateTime _startDate; // Дата начала тура
    }
}
```

```

        protected readonly int _duration; // Продолжительность тура в днях
        protected readonly decimal _price; // Цена тура

        public int ID { get { return _id; } }
        public string Name { get { return _name; } }
        public DateTime StartDate { get { return _startDate; } }
        public int Duration { get { return _duration; } }
        public decimal Price { get { return _price; } }

        public Tour(int id, string name, DateTime startDate, int duration,
decimal price)
        {
            _id = id;
            _name = name;
            _startDate = startDate;
            _duration = duration;
            _price = price;
        }

        public virtual void Print()
        {
            Console.WriteLine("Tour ID: {0}\nName: {1}\nStart Date:
{2}\nDuration: {3} days\nPrice: {4} P",
                ID, Name, StartDate.ToShortDateString(), Duration, Price);
        }
    }
}

```

AdventureTour.cs

```

namespace TourSales
{
    internal class AdventureTour : Tour
    {
        public AdventureTour(int id, string name, DateTime startDate, int
duration, decimal price)
            : base(id, name, startDate, duration, price) { }

        public override void Print()
        {
            base.Print();
            Console.WriteLine("Type: Adventure Tour");
        }
    }
}

```

```
    }  
}
```

CityTour.cs

```
namespace TourSales  
{  
    internal class CityTour : Tour  
    {  
        public CityTour(int id, string name, DateTime startDate, int  
duration, decimal price)  
            : base(id, name, startDate, duration, price) { }  
  
        public override void Print()  
        {  
            base.Print();  
            Console.WriteLine("Type: City Tour");  
        }  
    }  
}
```

BeachTour.cs

```
namespace TourSales  
{  
    internal class BeachTour : Tour  
    {  
        public BeachTour(int id, string name, DateTime startDate, int  
duration, decimal price)  
            : base(id, name, startDate, duration, price) { }  
  
        public override void Print()  
        {  
            base.Print();  
            Console.WriteLine("Type: Beach Tour");  
        }  
    }  
}
```

10. Выводы

В ходе выполнения работы была разработана программа, создающая три различных тура. Для достижения результата были использованы следующие механизмы объектно-ориентированного программирования:

- Абстрактный класс: был создан абстрактный класс «Tour», который содержит общие свойства и методы для всех типов туров.
- Виртуальный метод: метод «Print» был реализован как виртуальный в базовом классе Tour и переопределён в производных классах для добавления специфической информации.
- Перегрузка метода: в производных классах («BeachTour», «CityTour», «AdventureTour») метод «Print» был перегружен для вывода дополнительных сведений, таких как тип тура.
- Интерфейс и его реализация: был создан интерфейс «IPrintable», реализованный классом «Tour» и его наследниками для унификации способа вывода информации.

Таким образом, программа демонстрирует использование принципов наследования, полиморфизма и интерфейсов для достижения поставленной задачи.

Список использованной литературы

1. Microsoft Learn – сеть разработчиков Microsoft. URL: <https://learn.microsoft.com/ru-ru/dotnet/csharp/> (дата обращения: 18.11.2024)
2. Полное руководство по языку программирования C# 12 и платформе .NET 8. URL: <https://metanit.com/sharp/tutorial/> (дата обращения: 18.11.2024)
3. Горячев А.В., Кравчук Д.К., Новакова Н.Е. Объектно-ориентированное моделирование. Учеб. Пособие. СПб.: Изд-во СПбГЭТУ “ЛЭТИ”, 2010. (дата обращения: 18.11.2024)
4. Язык UML = The Unified Modeling Language User Guide : руководство пользователя / Г. Буч, Д. Рамбо, А. Джекобсон (дата обращения: 18.11.2024)