

## **Лабораторная работа №5**

### **Использование коллекций**

#### **Цель работы:**

получить навыки проектирования приложения, состоящих из нескольких взаимосвязанных классов.

#### **Время выполнения – 4 академических часа**

#### **Задание к работе**

- Для заданной предметной области спроектировать программную структуру, состоящую из 3-5 классов.
- В соответствии с разработанной диаграммой классов выполнить программную реализацию.
- Предусмотреть использование типа данных – перечисление.
- Ввод/вывод должен быть реализован вне проектируемого класса.

**Примечание:** меню для взаимодействия с пользователем реализовывать не нужно. Для демонстрации работы приложения достаточно инициализацию исходных данных и проверку функционала приложения реализовать в коде

#### **Индивидуальные задания**

##### **1. Предметная область: АТС.**

На АТС хранится информация о всех клиентах станции. АТС имеет список тарифов на междугородние разговоры. Клиент АТС может совершать множество звонков в различные города.

Система должна:

- позволять вводить информацию о тарифах;
- вводить информацию о клиентах и регистрировать звонки;
- по введенной фамилии о клиенте определять стоимость всех сделанных им звонков в соответствии с действующими тарифами;
- вычислять общую стоимость всех выполненных на АТС звонков.

##### **2. Предметная область: Вокзал.**

Касса вокзала имеет список тарифов на различные направления. При покупке билета регистрируются паспортные данные пассажира. Пассажир покупает билеты на различные направления.

Система должна:

- позволять вводить данные о тарифах;

- позволять вводить паспортные данные пассажира и регистрировать покупку билета;
- рассчитывать стоимость купленных пассажиром билетов;
- после ввода наименования направления, выводить список всех пассажиров, купивших на него билет;

### **3. Предметная область: ЖЭС.**

В ЖЭС хранятся тарифы на коммунальные услуги. ЖЭС имеет информацию обо всех жильцах. При потреблении жильцами коммунальных услуг информация регистрируется в системе.

Система должна позволять выполнять следующие задачи:

- ввод тарифов
- ввод информации о жильцах и потребленных услугах;
- после ввода фамилии, выводить сумму всех потребленных услуг;
  - выводить стоимость всех оказанных услуг;

### **4. Предметная область: Аэропорт.**

Касса аэропорта имеет список тарифов на различные направления. При покупке билета регистрируются паспортные данные.

Система должна:

- позволять вводить данные о тарифах;
- позволять вводить паспортные данные пассажира и регистрировать покупку билета;
- рассчитывать стоимость купленных пассажиром билетов;
- рассчитывать стоимость всех проданных билетов.

### **5. Предметная область: Банк.**

Информационная система банка хранит описание процентов по различным вкладам. Система хранит информацию о вкладчиках и сделанных ими вкладах. Каждый клиент может поместить в банк только один вклад.

Система должна позволять выполнять следующие задачи:

- хранить информацию о процентах по вкладам;
- хранить информацию о клиентах;
- пополнять клиенту величину вклада;
- вычислять общую сумму выплат по процентам для всех вкладов.

### **6. Предметная область: Отдел расчета зарплаты.**

Информационная система отдела расчета зарплаты на предприятии хранит данные о величине оплаты различных видов работ. Система хранит информацию о работниках предприятия.

Система должна позволять выполнять следующие задачи:

- вводить информацию о различных видах работ;
- вводить информацию о работниках и выполненных ими работах;
- после ввода фамилии выводить для работника зарплату;
- выводить сумму выплат всем работникам;

#### **7. Предметная область: Фирма грузоперевозок.**

Фирма имеет список тарифов по перевозке грузов. Клиент регистрируется в системе, после чего может заказать перевозку определенного объема груза.

Система должна позволять выполнять следующие задачи:

- ввод тарифов;
- регистрация клиента и заказ на перевозку грузов;
- вывод суммы заказа для определенного клиента;
- подсчет суммарной стоимости всех заказов.

#### **8. Предметная область: Гостиница.**

Информационная система гостиницы хранит информацию обо всех номерах и их стоимости. Система регистрирует клиентов. Каждый клиент может заказать один номер. При попытке заказа номера, который занят, выводится предупреждение.

Система должна позволять выполнять следующие задачи:

- ввод информации о номерах и их стоимости;
- регистрация клиента и заказ номера;
- вывод списка не занятых номеров;
- после ввода фамилии клиента, вывод стоимости проживания.

#### **9. Предметная область: Интернет оператор.**

Провайдер имеет различные тарифы доступа в Интернет за 1Мбайт в зависимости от величины абонентской платы. Информационная система провайдера хранит данные о клиентах.

Система должна позволять выполнять следующие задачи:

- ввод тарифов;
- регистрация пользователя;
- ввод данных о потребленном трафике для конкретного пользователя;
- подсчет общей стоимости реализованного трафика;
- поиск клиента, заплатившего наибольшую стоимость за услуги.

#### **10 Предметная область: Интернет магазин.**

В информационной системе хранятся данные о товарах. Клиент звонит в магазин и оставляет заказ на товар.

Система должна позволять выполнять следующие задачи:

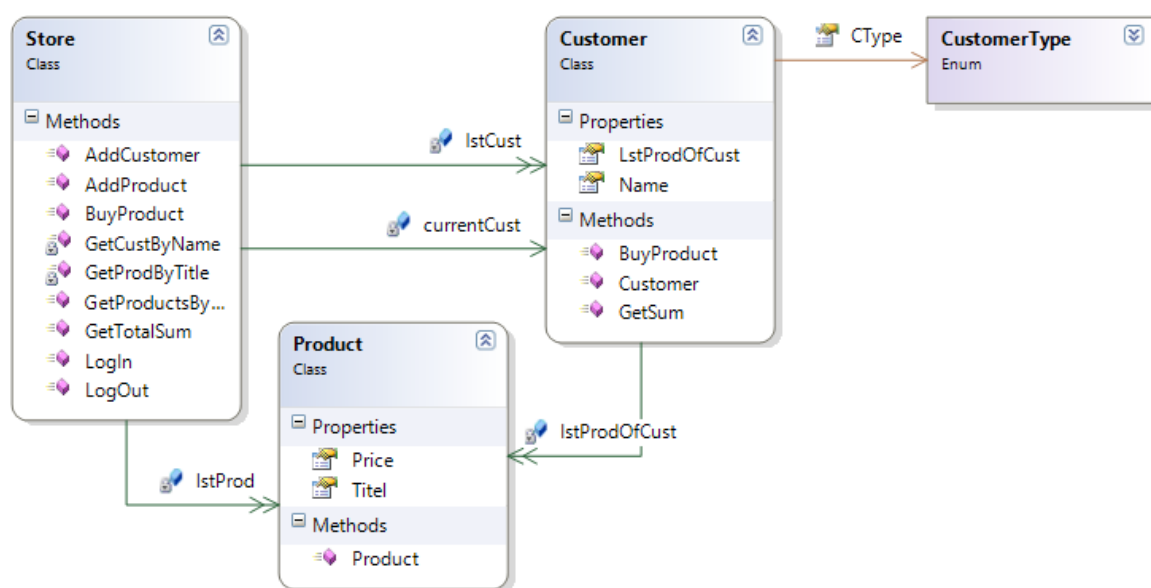
- ввод информации о товарах;
- регистрация заказа клиента на покупку определенного товара;
- после ввода фамилии покупателя вывод списка заказанных им товаров;
- после ввода фамилии покупателя вывод суммы заказа.

### Пример выполнения работы

#### Предметная область: Торговая система.

В информационной системе хранятся данные о товарах и покупателях. Некоторые покупатели имеют статус “VIP”. Для приобретения товара вводится фамилия, после чего регистрируются купленные товары. Система должна высчитывать общую сумму проданных товаров.

Диаграмма классов:



**CustomerType** – перечисление. Описывает тип покупателя (VIP или Standard)

#### Класс Product – описывает товар

Свойства		
Price		Стоимость товара
Title		Наименование товара
Методы		

Product		Конструктор класса
---------	--	--------------------

### Класс Store – описывает магазин

Поля		
lstCust	private	Список покупателей
currentCust	private	Покупатель, совершающий покупки (вошедший в систему)
lstProd	private	Список товаров
Методы		
AddCustomer		Добавление покупателя в список
AddProduct		Добавление товара в список
BuyProduct		Покупка товара текущим покупателем (вошедший в систему)
GetCustByName	private	Поиск покупателя по имени
GetProdByTitle	private	Поиск товара по названию
GetProdutsByName		Получение списка товаров, купленных покупателем с заданным именем
GetTotalSum		Получение общей стоимости всех проданных товаров
LogIn		Вход в систему (по имени) для совершения покупок
LogOut		Выход из системы (завершение покупок)

### Класс Customer – описывает покупателя

Поля		
lstProd	private	Список купленных товаров
Свойства		
Name		Имя покупателя
LstProd		Список купленных товаров. Возвращает поле lstProd
CType		Тип покупателя - <b>CustomerType</b>
Методы		
Customer		Конструктор класса. Принимает имя покупателя и его тип
BuyProduct		Добавить товар в список покупок
GetSum		Получение общей стоимости всех купленных товаров

Код программы:

```
/// <summary>
```

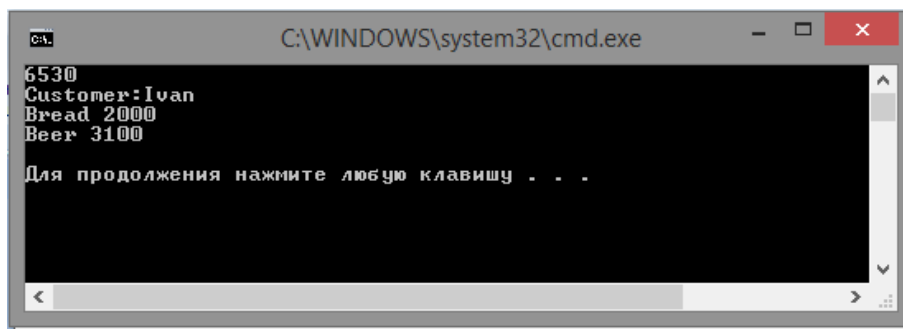
```

/// Основная программа
/// </summary>
class Program
{
    static void Main()
    {
        Store riga = new Store();
        riga.AddProduct("Milk", 1430);
        riga.AddProduct("Bread", 2000);
        riga.AddProduct("Beer", 3100);
        riga.AddCustomer("Masha", CustomerType.VIP);
        riga.AddCustomer("Ivan", CustomerType.usual);
        riga.LogIn("Ivan");
        riga.BuyProduct("Bread");
        riga.BuyProduct("Beer");
        riga.LogOut();
        riga.LogIn("Masha");
        riga.BuyProduct("Milk");
        riga.LogOut();

        Console.WriteLine(riga.GetTotalSum());
        Console.WriteLine(riga.GetProductsByName("Ivan"));
    }
}

```

Результат выполнения программы:



### Вопросы для самопроверки

1. Чем коллекция List отличается от массива?
2. Можно ли получить доступ к элементам коллекции List<T> по индексу?
3. Имеется массив **names**. Что означает выражение **names[1..4]** ?
4. Имеется массив **names**. Что означает выражение **names[^2]** ?
5. Приведите пример использования коллекции Queue
6. Какое преимущество коллекции **Dictionary** по сравнению с **List** ?
7. Что такое **HashSet** ? Отличие HashSet от других коллекций.
8. Что такое **Enum** ? Приведите пример использования.
9. Как создать кортеж?
10. Как получить доступ к элементам кортежа?

## Лабораторная работа №6

### Наследование

#### Цель работы:

Изучить механизмы наследования в C#.

**Время выполнения – 4 академических часа**

#### Задание к работе

- Выделить в предметной области 2-3 варианта сущности (например, "Геометрическая фигура", "Точка", "Треугольник"), отличающиеся несколькими полями и методами. Каждый класс имеет поля, свойства и методы.
- Спроектировать UML-диаграммы классов.
- **Базовый** класс для вашей иерархии объявите **абстрактным**. Он должен содержать **абстрактные методы** и **методы с реализацией**.
- Один из наследников должен **перегружать** метод родителя.
- Один из классов должен содержать **виртуальный метод**, который **переопределяется** в одном наследнике и **не переопределяется** в другом.
- Продемонстрировать работу всех объявленных методов.
- Продемонстрировать **вызов конструктора родительского класса** при наследовании.
- Продемонстрировать вызов метода родительского класса при его **скрытии**.
- Создать класс, **закрытый для наследования**.

**Примеры методов:** вывод полей класса, изменение числовых полей класса (скорости, дохода, численности штата и т.д.), изменение текстовых полей класса (переименование), какая-либо динамика (полет, ремонт, переезд, открытие/закрытие и.д.)

#### Индивидуальные задания

1. Геометрические фигуры
2. Здания
3. Транспорт
4. Автомобили
5. Товары
6. Игровые юниты
7. Компьютерная техника
8. Игрушки
9. Мебель

10. Фильмы
11. Техника
12. Еда
13. Алкогольные напитки
14. Коктейли
15. Фрукты или овощи
16. Десерты
17. Космические объекты
18. Спорт
19. Животные
20. Растения

### Вопросы для самопроверки

1. Поддерживает ли C# множественное наследование?
2. Есть базовый класс **A** и производный от него класс **B**. Будет ли вызываться конструктор класса **A** при создании объекта класса **B**?
3. Может ли абстрактный класс содержать неабстрактные члены?
4. Можно ли создать объект абстрактного класса?
5. Может ли наследник абстрактного класса тоже быть абстрактным?
6. Есть базовый класс **A** и два производных класса: **B** и **C**. Допустимо ли такое присваивание:

```
A a;  
a = new B();
```

7. Есть базовый класс **A** и два производных класса: **B** и **C**. Допустимо ли такое присваивание:

```
B b;  
b = new C();
```

8. Есть базовый класс **A** и производный класс **B**. В классе **B** добавлен метод **Execute**. Как из переменной **a** вызвать метод **Execute** :

```
A a;  
a = new B();
```

9. Есть базовый класс **A** и два производных класса: **B** и **C**. Допустимо ли такое присваивание:

```
A a;  
a = new B();  
B b = a;
```

10. Что такое «явное приведение типа»?
11. Для чего используется оператор **is** ?
12. Для чего используется оператор **as** ?
13. Как из производного класса вызвать метод базового класса?



14. Какой модификатор доступа используется для того, чтобы член класса был виден только в иерархии наследования классов?
15. **Базовый** класс имеет модификатор доступа **internal**. Может ли **производный** класс иметь модификатор доступа **public**?
16. Что такое сокрытие метода? Как реализуется?
17. Что такое переопределение метода? Как реализуется?
18. Чем отличается абстрактный метод от виртуального?

## Лабораторная работа №7

### Перегрузка операторов

#### Цель работы:

Изучить механизмы реализации полиморфизма в C#.

#### Задание к работе

- Спроектировать класс согласно варианту индивидуального задания. Для класса использовать отдельный модуль.
- Спроектировать конструкторы и свойства с контролем корректности вводимых значений.
- перегрузить метод `toString()`
- добавить индексирование для получения полей класса
- Перегрузить операции:
  - a) математические (имеющие смысл для объектов класса)
  - b) инкремент и декремент (изменить поля на 1)
  - c) отношения (`==`, `!=`, `<`, `>`)
  - d) `true` и `false`
  - e) преобразования типа

*В методе `main()`*

- Создать несколько объектов класса. Продемонстрировать использование конструкторов и свойств.
- Продемонстрировать работу всех методов и операций.

**Примечание:** меню для взаимодействия с пользователем реализовывать не нужно. Для демонстрации работы приложения достаточно инициализацию исходных данных и проверку функционала приложения реализовать в коде

#### Индивидуальные задания

1. Класс `complex` (Комплексное число). Тип полей – `double`. Перегрузить `+`, `-`, `/`, `*` для 2 комплексных, для комплексного и `double`. Сравнивать по модулю. (d) если обе части = 0, число = `false`. Преобразовать `complex` в `double` явно, назад – неявно.

2. Класс `rational` (рациональное число). Поля – `int` (числитель и знаменатель). Добавить метод нормализации (сокращение) и НОД. Перегрузить `+`, `-`, `/`, `*` для 2 `rational`, для `rational` и `int`. (d) если числитель = 0, число = `false`. Преобразовать `rational` в `int` явно, назад – неявно.

3. Класс `interval` (интервал). Поля – `double` (2 границы). Добавить метод вычисления длины. Перегрузить `+`, `-`, `*` (пересечение) – проверить возможность выполнения операции. `++` и `--` - расширение и сужение интервала.

Сравнивать по длине. (d) если длина интервала = 0, интервал = false. Преобразовать интервал в double явно (длина), назад – также явно.

4. Класс квадратное уравнение. Поля – int (a, b, c). Добавить метод нахождения корней. Перегрузить +, -, ++, --, \* на число, / на число. Сравнить на == и !=. (d) если уравнение не имеет корней, уравнение = false. Преобразовать в число (a) и назад ( $ax^2$ ) – в обоих случаях явно.

5. Класс полином  $ax+by+cz$ . Поля – int (a, b, c). Перегрузить +, -, ++, --, \* на число, / на число. Сравнить на == и !=. (d) если  $a=b=c=0$ , полином = false. Преобразовать в число (a) и назад ( $ax$ ) – в обоих случаях явно.

6. Класс vector. Поля – int (a, b, c). Перегрузить +, -, ++, --, \*, \* на число, / на число. Сравнить на == и !=. (d) если  $a=b=c=0$ , вектор = false. Преобразовать в число (модуль) и назад ( $ax$ ) – в обоих случаях явно.

7. Класс квадратная матрица 2\*2 типа int. К полям обращаться через индексатор. Перегрузить +, -, ++, --, \*, \* на число, / на число. Сравнить на == и !=. (d) если определитель = 0, матрица = false. Преобразовать в число (опредетитель) и назад ( $a,0,0,a$ ) – в обоих случаях явно.

8. Класс – множество символов. В качестве поля – массив символов. Добавить метод, проверяющий корректность множества (нет повторений) – нужен сеттеру. Перегрузить +, -, \* (пересечение). Сравнить на == и !=. (d) если множество пустое, оно false. Преобразовать в String – неявно, назад – явно.

9. Класс – окружность. Задаются координатами центра и радиусом (double). Добавить методы вычисления периметра и площади. Перегрузить ++, --, -, \*, / - изменяя соответственно радиус. Сравнивать окружности по площади. (d) если центр в начале координат, окружность = false. Преобразовать double в окружность – с центром в начале координат и радиусом double (неявно). Назад – явно.

10. Класс – треугольник. Задаются длинами сторон (double). Добавить методы вычисления периметра и площади. Перегрузить ++, --, -, \*, / - изменяя соответственно все 3 стороны. Сравнивать по площади. (d) если равносторонний = false. Преобразовать double в треугольник – равносторонний (неявно). Назад – явно.

### Вопросы для самопроверки

1. Для чего используется перегрузка операторов?
2. Какие операторы перегружаются обязательно попарно?
3. Для чего используется оператор **Implicit** ?
4. Для чего используется оператор **Explicit** ?
5. То такое индексатор?
6. Методы перегрузки операторов являются статическими или экземпляльными?
7. Что такое «метод расширения (расширяющий метод, extension method)?

## Лабораторная работа №8 Полиморфизм

### Цель работы:

Изучить механизмы реализации полиморфизма в C#.

Ознакомиться с основными подходами при использовании интерфейсов.

**Время выполнения – 4 академических часа**

### Задание к работе

- Составить диаграмму классов проектируемой системы.
- Запрограммировать классы в соответствии с новой диаграммой.
- Проиллюстрировать использование интерфейсов.
- Показать вызов метода интерфейса через интерфейсную ссылку.
- Применить в программе шаблон проектирования Strategy.

**Примечание:** меню для взаимодействия с пользователем реализовывать не нужно. Для демонстрации работы приложения достаточно инициализацию исходных данных и проверку функционала приложения реализовать в коде

### Индивидуальные задания

1. Предметная область: АТС. АТС имеет список тарифов на междугородние разговоры. Есть два типа тарифов: обычный и льготный. В классе АТС реализовать методы добавления обычного тарифа и добавления льготного тарифа. Класс АТС должен выполнять вычисление средней стоимости тарифов с учетом скидки.

2. Предметная область: Вокзал. Класс “вокзал” имеет список тарифов на различные направления. На некоторые тарифы может быть предоставлена скидка, заданная в процентах. В классе “вокзал” реализовать методы добавления нового тарифа со скидкой и без скидки, поиск направления с минимальной стоимостью.

3. Предметная область: ЖЭС. ЖЭС имеет информацию обо всех жильцах. Имеется два типа жильцов со льготами и без льгот. В классе ЖЭС реализовать метод добавления нового жильца имеющего льготы и без льгот, а так же метод подсчета стоимости всех оказанных услуг.

4. Предметная область: Аэропорт. Касса аэропорта имеет список тарифов на различные направления. Тариф содержит название направления и стоимость перевозки. На некоторые направления предоставляется фиксированная скидка. В классе аэропорт реализовать метод добавления нового тарифа и метод поиска направления с максимальной стоимостью.

5.Предметная область: Банк. Система хранит информацию о вкладчиках и сделанных ими вкладах. Класс “вкладчик” содержит имя вкладчика и величину вклада. Некоторым вкладчикам при создании вклада на счет может дополнительно перечисляться фиксированная сумма. В классе банк реализовать методы добавления нового вкладчика и метод вычисления общей суммы вкладов.

6. Предметная область: Отдел расчета зарплаты. Информационная система отдела расчета зарплаты на предприятии хранит данные о величине оплаты за различные виды работ. На некоторые виды работ предоставляется надбавка, заданная в процентах. В классе “отдел расчета зарплаты” реализовать методы добавления нового типа работ и метод вычисления средней величины оплаты.

7.Предметная область: Фирма грузоперевозок. Фирма имеет список тарифов по перевозке грузов. Класс тариф хранит наименование тарифа и цену. На не которые тарифы предоставлена скидка, заданная в процентах. В классе фирма реализовать методы добавления нового тарифа и метод поиска тарифа с минимальной стоимостью.

8.Предметная область: Гостиница. Информационная система гостиницы хранит информацию обо всех номер ах и их стоимости. На проживание в некоторых номерах предоставляется скидка, заданная в процентах. В классе “гостиница” реализовать метод добавления информации о номере и метод вычисления средней стоимости.

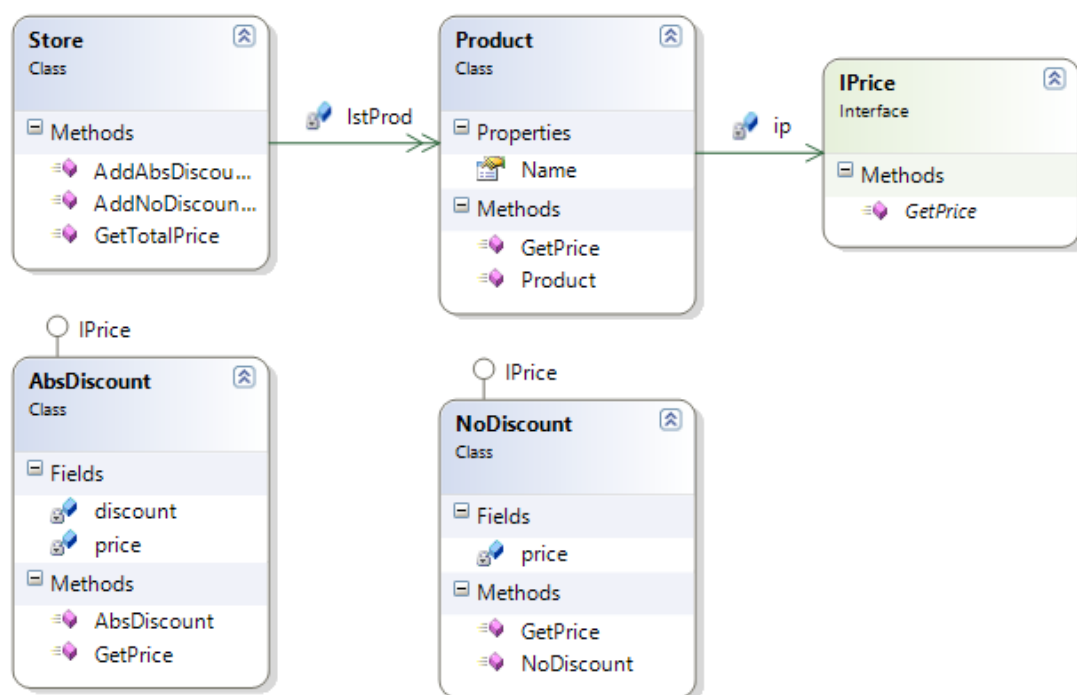
9. Предметная область: Интернет оператор. Информационная система провайдера хранит данные о клиентах. Некоторым клиентам предоставляется фиксированная скидка. В классе “оператор” реализовать метод добавления нового клиента и метод вычисления суммарной стоимости оказанных услуг.

10 Предметная область: Интернет магазин. В информационной системе хранятся данные о товарах. Класс «товар» содержит стоимость товара и его наименование. На некоторые товары предоставляется скидка, заданная в процентах. В классе магазин реализовать метод добавления нового товара имеющего скидку и без скидки, также метод поиска товара с минимальной стоимостью.

### **Пример выполнения работы**

Пусть есть магазин с множеством продуктов. На некоторые продукты предоставляется скидка. Требуется подсчитать суммарную стоимость товаров в магазине с учетом предоставляемых скидок.

## UML диаграмма классов:



### Интерфейс IPrice – описывает метод получения стоимости товара

Методы		
GetPrice		Возвращает стоимость товара

### Класс Product – описывает товар

Поля		
ip	private	Объект класса, реализующего интерфейс IPrice
Свойства		
Name		Название товара
Методы		
Product		Конструктор класса.
GetPrice		Возвращает стоимость товара

### Класс AbsDiscount – реализация интерфейса IPrice для подсчета стоимости товара со скидкой

Поля		
discount	private	Величина скидки
Price	private	Стоимость товара
Методы		
AbsDiscount		Конструктор класса. Принимает значения стоимости товара и скидки

GetPrice		Реализация интерфейса IPrice для подсчета стоимости товара со скидкой
----------	--	---

**Класс NoDiscount – реализация интерфейса IPrice для подсчета стоимости товара без скидки**

<b>Поля</b>		
Price	private	Стоимость товара
<b>Методы</b>		
AbsDiscount		Конструктор класса. Принимает значение стоимости товара
GetPrice		Реализация интерфейса IPrice для подсчета стоимости товара без скидки

**Класс Store – описывает магазин**

<b>Поля</b>		
LstProd	private	Список товаров
<b>Методы</b>		
AddAbsDiscount		Добавление товара со скидкой
AddNoDiscount		Добавление товара без скидки
GetTotalPrice		Получение общей стоимости всех товаров

**Код программы**

```
class Program
{
    static void Main(string[] args)
    {
        Store riga = new Store();
        riga.AddAbsDiscountProduct(" bread", 3200, 200);
        riga.AddNoDiscountProduct("milk", 1200);
        Console.WriteLine(riga.GetTotalPrice());
    }
}
```

**Вопросы для самопроверки**

1. Что такое «Полиморфизм» в ООП?
2. Допускается ли в С# множественное наследование интерфейсов?

3. Может ли класс, реализующий интерфейс, реализовать только часть методов, описанных в интерфейсе?
4. Может ли интерфейс содержать методы с модификатором **private**?
5. Может ли интерфейс содержать статические методы
6. Как описать конструктор интерфейса?
7. Может ли интерфейс содержать реализация методов и свойств по умолчанию?
8. Что такое «**коллизия имен**»? Когда она возникает?
9. Что такое «**склеивание**» при решении коллизии имен?
10. Что такое «**явная реализация интерфейса (кастинг)**» при решении коллизии имен?
11. Как получить доступ к методу интерфейса при кастинге?
12. Чем отличается интерфейс от абстрактного класса?
13. Для чего используется интерфейс **Comparable**?