Министерство образования Республики Беларусь
Учреждение образования «Белорусский государственный университет информатики и радиоэлектроники»

Факультет компьютерных систем и сетей
Кафедра Информатики
Дисциплина «Программирование»

ОТЧЕТ
к лабораторной работе №6
на тему:
«НАСЛЕДОВАНИЕ»
БГУИР 6-05-0612-02 113

Выполнил студент группы 453503
ХАЛАМОВ Николай Андреевич

_____
(дата, подпись студента)

Проверил ассистент каф. Информатики
РОМАНЮК Максим Валерьевич

_____
(дата, подпись преподавателя)

Минск 2025

1 ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ

Задание 1. Вариант 13. Предметная область: Алкоголь.

Для заданной предметной области реализовать следующие задачи:

– выделить в предметной области 2-3 варианта сущности, 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;

2 ВЫПОЛНЕНИЕ РАБОТЫ

Перед выполнением работы следует разработать диаграмму классов для наглядного выполнения поставленной задачи (см. рисунок 1).

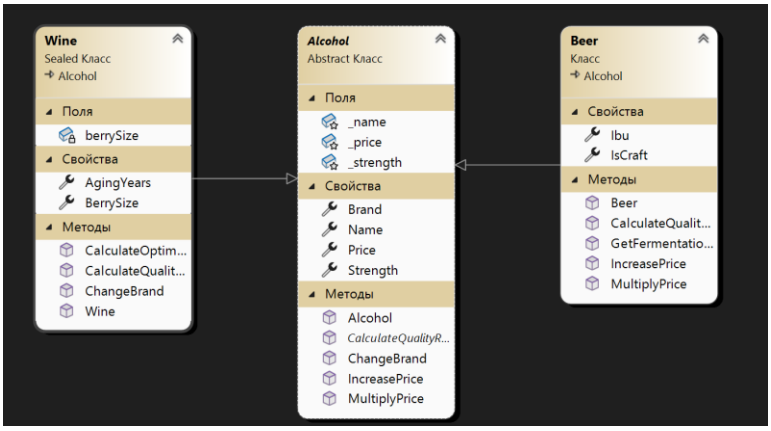

Рисунок 1 – Диаграмма классов

Для выполнения задания в проект были добавлены классы Alcohol, Beer, Wine, Demonstration(см. рисунок 2).

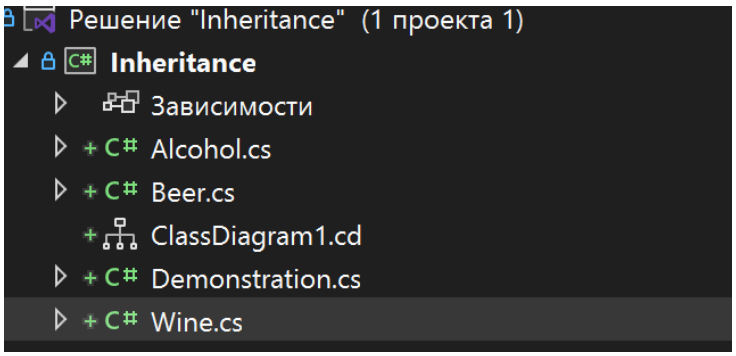

Рисунок 2 – Обозреватель решений

Рассмотрим реализацию абстрактного базового класса Alcohol. Он содержит поля крепость, цена, вид алкоголя и свойства для доступа, свойство бренд(марка) алкоголя. Их принимает конструктор класса. Метод для увеличения стоимости мы перегрузим в наследуемом классе Beer и оставим реализацию из базового класса для наследуемого класса Wine. Метод для изменения названия бренда мы переопределим в Wine и оставим реализацию из базового класса в Beer. Метод для вычисления рейтинга качества напитка мы сделаем абстрактным и реализуем отдельно в каждом классе-наследнике. Метод для увеличения цены с помощью умножения на число мы сделаем скрытым в Beer и оставим реализацию из базового класса в Wine.

```
namespace Inheritance
{
    public abstract class Alcohol
    {
        // Поля с protected доступом (видны наследникам)
        protected double _strength;
        protected double _price;
        protected string _name;

        // Публичные свойства для контролируемого доступа
        public string Brand
        {
            get;
            set;
        }
        public double Strength
        {
            get;
            set;
        }

        public string Name
        {
            get;
            set;
        }

        public double Price
        {
            get;
            set;
```

```
        }

        //Конструктор родительского класса
        public Alcohol(double strength,double price, string name,string
brand)
        {
            Strength = strength;
            Price = price;
            Name = name;
            Brand = brand;
        }

        //Перегрузим в Beer и оставим в Wine
        public void IncreasePrice(double amount)
        {
            Price+=amount;
        }

        //Виртуальный метод, который переопределим в Wine и оставим в
Beer
        public virtual void ChangeBrand(string newBrand)
        {
            Brand = newBrand;
        }

        //Абстрактный метод, которые реализуем в наследниках по
отдельности
        public abstract double CalculateQualityRating();

        //Метод, с помощью которого продемонстрируем скрытие
        public void MultiplyPrice(double multiplier)
        {
            Price *= multiplier;
        }
    }
}
```

В классе Beer добавим поля и свойства горечи пива и его крафтовости. Добавим их в конструктор класса, также вызывается конструктор родительского класса с его полями и свойствами.Реализуем перегруженный метод увеличения цены и скрытый метод(с помощью new) умножения цены на число. Также реализуем абстрактный метод рейтинга качества с использованием новых полей и свойств класса. Добавим метод расчета времени ферментации.

```
namespace Inheritance
{
    internal class Beer : Alcohol
    {
        // Горечь пива
        public double Ibu
        {
            get;
            set;
        }
    public bool IsCraft
        {
            get;
            set;
```

```
        }
        public Beer(double ibu,double strength, double price, string
name,string brand,bool isCraft) : base(strength, price, name,brand)
        {
            Ibu = ibu;
            IsCraft = isCraft;
        }

        // Перегрузка метода IncreasePrice
        public void IncreasePrice(double amount, string reason, out
string operationDetails)
        {
            base.IncreasePrice(amount);
            operationDetails = $"Перегрузка метода.Цена увеличена на
{amount} по причине: {reason}";
        }

        // Скрытие метода MultiplyPrice
        public new void MultiplyPrice(double multiplier,out string
operationDetails)
        {
            double oldPrice = Price;
            base.MultiplyPrice(multiplier);  //  Вызов  родительского
метода
             operationDetails = $"Цена изменена: {oldPrice} -> {Price}
(x{multiplier})";
        }

    public override double CalculateQualityRating()
        {
            // Формула: базовая оценка + бонусы за крафт и горечь
            double baseRating = 3.0 + (Strength / 10); // 3-6 баллов за
крепость
            double craftBonus = IsCraft ? 2.5 : 0;
            double ibuBonus = Math.Min(Ibu / 20, 3); // + до 3 баллов
за горечь
            return Math.Round(baseRating + craftBonus + ibuBonus, 1);
        }

        // Расчёт времени ферментации
        public int GetFermentationTime() => IsCraft ? 14 : 7; // недели
    }
}
```

В еще одном классе-наследнике от Alcohol Wine добавим поля и свойства размера ягод и выдержки вина.Их добавим в конструктор класса и используем при подсчете рейтинга.Переопределим метод изменения бренда и реализуем метод для подсчета оптимального срока выдержки. Класс нельзя наследовать, использовано ключевое слово sealed.

```
namespace Inheritance
{
    //Класс, закрытый для наследования
    sealed class Wine : Alcohol
    {
        private double berrySize;
        public double BerrySize
        {
            get;
            set;
```

```
        }
        public int AgingYears
        {
            get;
            set;
        }
        public Wine(double strength, double price, string name,string
brand,int agingYears,double berrySize) : base(strength, price, name,brand)
        {
            AgingYears=agingYears;
            this.berrySize=berrySize;
        }

        public override void ChangeBrand(string newBrand)
        {
            Brand="Premium "+ newBrand;
        }

        public override double CalculateQualityRating()
        {
            // Базовый рейтинг (крепость и выдержка)
            double baseRating = 4.0 + (Strength / 20) +
Math.Min(AgingYears * 0.3, 4);

            // Влияние размера ягод:
            // - Мелкие ягоды (8-10мм) = +2 балла (более
концентрированный сок)
            // - Средние (10-12мм) = +1 балл
            // - Крупные (12-15мм) = -0.5 балла (разбавленный вкус)
            double berryImpact = berrySize switch
            {
                < 10 => 2.0,
                < 12 => 1.0,
                _ => -0.5
            };

            // Дополнительный бонус для идеального сочетания:
            // Мелкие ягоды + высокая крепость = +1.5
            if (berrySize < 10 && Strength > 13.5)
                berryImpact += 1.5;

            return Math.Round(baseRating + berryImpact, 1);
        }

        // Рассчитывает оптимальный срок выдержки
        public int CalculateOptimalAging()
            => (int)(AgingYears * 1.5); // Например, для молодого вина
    }
}
```

Результат работы классов и работа всех методов описана в классе Demonstration, чтобы наглядно разобраться в структуре созданных классов.

```
namespace Inheritance
{
internal class Demonstration
{
static void Main()
{
Console.WriteLine("=== Демонстрация работы класса Beer ===");
```

```
        var beer = new Beer(ibu: 45, strength: 6.5, price: 300,name: "Эль",
brand: "Baltika", isCraft: true);
        Console.WriteLine($"Beer создан: {beer.Name}, {beer.Brand},
{beer.Price} руб.,коэффициент горечи: {beer.Ibu} , крепость:
{beer.Strength},крафтовое? {beer.IsCraft}");
        Console.WriteLine($"Рейтинг качества:
{beer.CalculateQualityRating()}"); //Абстрактный метод

        Console.WriteLine($"Повысим стоимость пива.Стоимость сейчас
{beer.Price}"); //Перегруженный метод
        beer.IncreasePrice(100, "повышение спроса", out string
increaseDetails);
        Console.WriteLine(increaseDetails);

        Console.WriteLine($"Вызовем родительский метод для умножения,без
скрытия.Начальная цена {beer.Price}");
        beer.MultiplyPrice(1.5);
        Console.WriteLine($"Конечная цена {beer.Price}");

        Console.WriteLine("Вызовем скрытый метод");
        beer.MultiplyPrice(2,out string operationDetails);
        Console.WriteLine(operationDetails);

        Console.WriteLine($"Поменяем бренд.Старый бренд {beer.Brand}");
//Метод из родителя
        beer.ChangeBrand("Zatezky Gus");
        Console.WriteLine($"Новый бренд: {beer.Brand}");

        Console.WriteLine($"Время ферментации: {beer.GetFermentationTime()}
дней"); // Собственный метод



        Console.WriteLine("\n=== Демонстрация работы класса Wine ===");
        var wine = new Wine(strength: 14.0, price: 5000,
        name: "Полусладкое", brand: "Massandra",
        agingYears: 5, berrySize: 9.5);

        Console.WriteLine($"Wine создан: {wine.Name}, {wine.Brand},
{wine.Price} руб.,размер ягод: {wine.BerrySize}, крепость: {wine.Strength},
выдержка {wine.AgingYears} лет");

        Console.WriteLine($"Рейтинг качества:
{wine.CalculateQualityRating()}");  //Абстрактный метод


        Console.WriteLine($"Поменяем бренд.Старый бренд {wine.Brand}"); //
Вызов переопределенного виртуального метода
        wine.ChangeBrand("Chardonnet");
        Console.WriteLine($"Новый бренд: {wine.Brand}");

        Console.WriteLine($"Повысим стоимость вина.Стоимость сейчас
{wine.Price}"); //Метод из родителя
        beer.IncreasePrice(1000);
        Console.WriteLine($"Стоимость после повышения {wine.Price}");

        Console.WriteLine("Продемонстрируем метод из родителя,который скрывали
для пива.Тут возьмем реализацию родителя.");
        Console.WriteLine($"Начальная цена {wine.Price}");
        wine.MultiplyPrice(3);
        Console.WriteLine($"Конечная цена {wine.Price}");
```

```
        Console.WriteLine($"Оптимальная выдержка:
{wine.CalculateOptimalAging()} лет"); // Уникальный метод Wine

        }
        }
        }
```

Результат работы программы продемонстрирован ниже (см. рисунок 3).

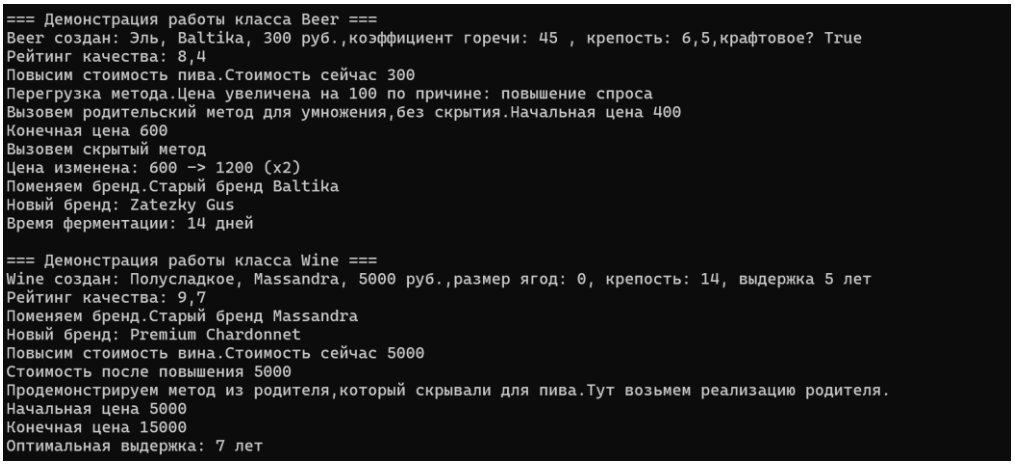

Рисунок 3 – Результат работы программы

ВЫВОД

В ходе лабораторной работы были изучены принципы построения диаграмм классов при наследовании классов. Изучены такие понятия как абстрактные классы, виртуальные методы, классы, закрытые для наследования, а также изменение работы метода родительского класса при его скрытии.