

Министерство образования Республики Беларусь
Учреждение образования
“Брестский государственный технический университет”
Кафедра ИИТ

Лабораторная работа №5

По дисциплине: "СПП"

Выполнил:
Студент 3 курса
Группы ПО-3
Лущ М. Г.
Проверил:
Крощенко А. А.

Брест 2019

Цель работы: научиться создавать и использовать классы в программах на языке программирования C#.

Вариант 14

Задание №1:

Реализовать абстрактные классы или интерфейсы, а также наследование и полиморфизм для следующих классов:

Абстрактный класс Книга (Шифр, Автор, Название, Год, Издательство). Подклассы Справочник и Энциклопедия.

Код программы:

```
using System;
```

```
using System.Text;
```

```
namespace lab5_1
```

```
{
```

```
    class Program
```

```
    {
```

```
        static void Main(string[] args)
```

```
        {
```

```
            Book book = new Encyclopedia();
```

```
            book.Read();
```

```
            book = new Guide();
```

```
            book.Read();
```

```
            Guide guide = (Guide)book;
```

```
            guide.Read();
```

```
        }
```

```
    }
```

```
    abstract class Book
```

```
    {
```

```
        public Encoding Encoding { get; set; }
```

```
        public string Author { get; set; }
```

```
        public string Name { get; set; }
```

```

        public int PublishYear { get; set; }

        public string Publisher { get; set; }

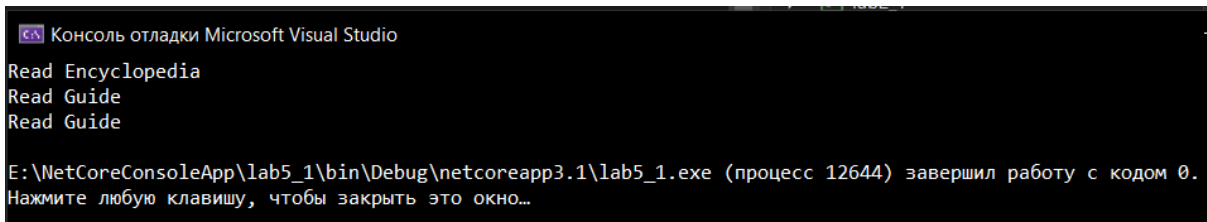
        public abstract void Read();
    }

    class Encyclopedia : Book
    {
        public override void Read()
        {
            Console.WriteLine("Read Encyclopedia");
        }
    }

    class Guide : Book
    {
        public override void Read()
        {
            Console.WriteLine("Read Guide");
        }
    }
}

```

Результаты работы программы:



```

Консоль отладки Microsoft Visual Studio
Read Encyclopedia
Read Guide
Read Guide

E:\NetCoreConsoleApp\lab5_1\bin\Debug\netcoreapp3.1\lab5_1.exe (процесс 12644) завершил работу с кодом 0.
Нажмите любую клавишу, чтобы закрыть это окно...

```

Задание №2:

В следующих заданиях требуется создать суперкласс (абстрактный класс, интерфейс) и определить общие методы для данного класса. Создать подклассы, в которых добавить специфические свойства и методы. Часть методов переопределить. Создать массив объектов суперкласса и заполнить объектами подклассов. Объекты подклассов идентифицировать конструктором по имени или идентификационному номеру. Использовать объекты подклассов для моделирования реальных ситуаций и объектов.

Создать суперкласс Домашнее животное и подклассы Собака, Кошка, Попугай. С помощью конструктора установить имя каждого животного и его характеристики.

Код программы:

```
using System;

namespace lab5_2
{
    class Program
    {
        static void Main(string[] args)
        {
            //Dog dog = new Dog("Владос", 5000, 0.4f);
            //Cat cat = new Cat("Смэртъ", 5000, 0.4f);
            //dog.Damage = 100;
            //cat.Damage = 90;
            //Fight(dog, cat);

            HomeAnimal[] animals = new HomeAnimal[3];
            animals[0] = new Dog("Master", 100, 0.2f, 10);
            animals[1] = new Cat("Shadow", 100, 0.5f, 10);
            animals[2] = new Parrot("Death");

            animals[1].Hit(animals[2]);
            animals[0].Hit(animals[1]);
            animals[1].Heal(10);

        }

        static void Fight(HomeAnimal first, HomeAnimal second)
        {

            while (first.IsAlive && second.IsAlive)
            {
                if (first.IsAlive)
                {
                    first.Hit(second);
```

```

    }
    if (second.IsAlive)
    {
        second.Hit(first);
    }
    Console.WriteLine("~~~~~");
}
}
}

```

abstract class HomeAnimal

```

{
    public readonly string name;
    private float _health;
    private float _maxHealth;
    private bool _isAlive;

    public float Health { get => _health; }
    public bool IsAlive { get => _isAlive; }
}

```

public HomeAnimal(string name, float health)

```

{
    this.name = name;
    _isAlive = true;
    _health = health;
    _maxHealth = health;
}

public void Heal(int health)
{
    _health += health;
    if (_health > _maxHealth)
        _health = _maxHealth;
}

```

```
        Console.WriteLine($"Животное по кличке {name} излечили на {health} hp. Текущее  
здоровье: {_health}");
```

```
    }
```

```
    public virtual void ApplyDamage(float damage)
```

```
    {
```

```
        _health -= damage;
```

```
        if (_health <= 0)
```

```
        {
```

```
            Die();
```

```
        }
```

```
    }
```

```
    protected virtual void Die()
```

```
    {
```

```
        _isAlive = false;
```

```
    }
```

```
    public abstract void Hit(HomeAnimal animal);
```

```
}
```

```
class Dog : HomeAnimal
```

```
{
```

```
    private float _damage;
```

```
    private float _resist;
```

```
    public float Damage
```

```
    {
```

```
        get => _damage;
```

```
        set
```

```
        {
```

```
            if (value > 0)
```

```
                _damage = value;
```

```
            else
```

```
                Console.WriteLine("Урон должен быть больше нуля");
```

```
        }
```

```
    }
```

```

public float Resist
{
    set
    {
        if (value >= 0 && value < 1f)
            _resist = value;
        else
            Console.WriteLine("Сопротивление должно быть в промежутке: [0, 1)");
    }
}

public Dog(string name, float health, float resist, float damage) : base(name, health)
{
    Resist = resist;
    Damage = damage;
}

public override void ApplyDamage(float damage)
{
    damage = damage - _resist * damage;

    Console.WriteLine($"Собака получает {damage} урона с учётом сопротивления. Текущее
здоровье: {Health}");
    base.ApplyDamage(damage);
}

protected override void Die()
{
    Console.WriteLine($"Собаку по кличке {name} съели");
    base.Die();
}

public override void Hit(HomeAnimal animal)
{
    Console.WriteLine($"Собака {name} наносит урон животному по кличке {animal.name}. Урон:
{Damage}");
}

```

```
        animal.ApplyDamage(Damage);
    }
}
```

```
class Cat : HomeAnimal
```

```
{
    private float _damage;
    private float _evasionChance;
    public float Damage
    {
        get => _damage;
        set
        {
            if (value > 0)
                _damage = value;
            else
                Console.WriteLine("Урон должен быть больше нуля");
        }
    }
    public float EvasionChance
    {
        set
        {
            if (value > 0 && value < 1f)
                _evasionChance = value;
            else
                Console.WriteLine("Шанс уворота должен быть в промежутке: [0, 1)");
        }
    }
    public Cat(string name, float health, float evasionChance, float damage) : base(name, health)
    {
        EvasionChance = evasionChance;
        Damage = damage;
    }
}
```



```

public override void ApplyDamage(float damage)
{
    Random random = new Random();
    if (random.NextDouble() < _evasionChance)
    {
        Console.WriteLine($"Кот уворачивается от удара. Текущее здоровье: {Health}");
    }
    else
    {
        Console.WriteLine($"Кот получает {damage} урона. Текущее здоровье: {Health}");
        base.ApplyDamage(damage);
    }
}

protected override void Die()
{
    Console.WriteLine($"Кота по кличке {name} съели");
    base.Die();
}

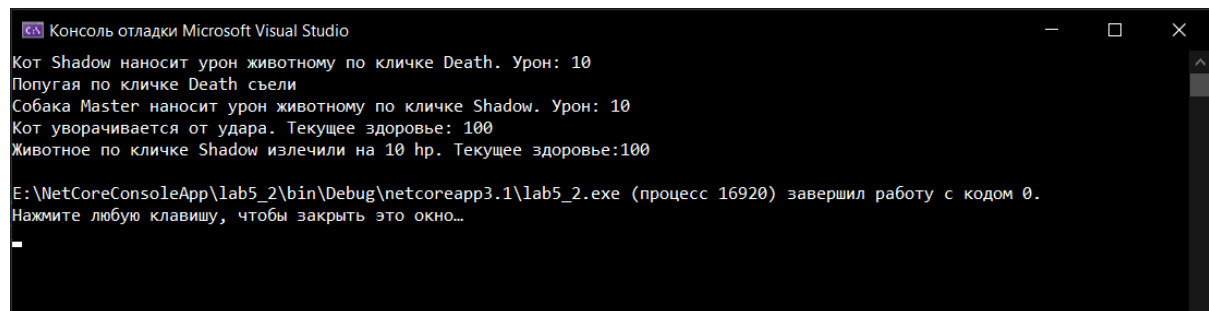
public override void Hit(HomeAnimal animal)
{
    Console.WriteLine($"Кот {name} наносит урон животному по кличке {animal.name}. Урон: {Damage}");
    animal.ApplyDamage(Damage);
}

class Parrot : HomeAnimal
{
    public Parrot(string name) : base(name, health: 1) { }
    protected override void Die()
    {
        Console.WriteLine($"Попугая по кличке {name} съели");
        base.Die();
    }
}

```

```
public override void Hit(HomeAnimal animal)
{
    Console.WriteLine("У попугая слишком мало урона");
}
}
```

Результаты работы программы:

A screenshot of the 'Консоль отладки Microsoft Visual Studio' (Visual Studio Debug Console) window. The window has a title bar with the Visual Studio icon and standard minimize, maximize, and close buttons. The console text is as follows:
Кот Shadow наносит урон животному по кличке Death. Урон: 10
Попугая по кличке Death съели
Собака Master наносит урон животному по кличке Shadow. Урон: 10
Кот уворачивается от удара. Текущее здоровье: 100
Животное по кличке Shadow излечили на 10 hp. Текущее здоровье:100

E:\NetCoreConsoleApp\lab5_2\bin\Debug\netcoreapp3.1\lab5_2.exe (процесс 16920) завершил работу с кодом 0.
Нажмите любую клавишу, чтобы закрыть это окно...
A small cursor is visible on the line following the prompt.

Задание №1:

В задании 3 ЛР №4, где возможно, заменить объявления суперклассов объявлениями абстрактных классов или интерфейсов.

В прошлой лабораторной работе при выполнении задания везде где можно использовать абстрактные классы или интерфейсы они использовались.