**Выполнили студент 215 группы**

**Марков К.А.**

**Практическое занятие 10**

**Тема: «Наследование классов с#»**

**Цель работы:** научиться разрабатывать классы, их основныекомпоненты: поля, методы, конструкторы, создавать объекты класса, обращаться к открытым полям и методам класса.

**Задание:**

**Вариант 7 и 13:** Разработать иерархию классов. Выделить родительский класс. Описать конструкторы в каждом классе. Создать перегруженные и виртуальные методы, а также методы скрывающие методы родителя. Продемонстрировать работу объектов классов в приложении. **7 -** местность, область, город;13 - млекопитающие, земноводные, животное;

7 задание

class Locality

{

public string Local { get; set; }

public Locality() { }

public Locality(string local)

{

Local = local;

}

virtual public void Info()

{

Console.WriteLine($"Местность: {Local}");

}

}

class Region : Locality

{

public string Rregin { get; set; }

public Region() { }

public Region(string local,string rregin):base(local)

{

Rregin = rregin;

}

public override void Info()

{

base.Info();

Console.WriteLine($"Регион:{Rregin}");

}

}

class City:Region

{

public string Town { get; set; }

public City() { }

public City(string local, string rregin,string town) : base(local,rregin)

{

Town = town;

}

public override void Info()

{

base.Info();

Console.WriteLine($"Город: {Town}");

}

}

internal class Program

{

static void Main(string[] args)

{

List<City> cities = new List<City>();

cities.Add(new City("Восточная Европа", "Челябинская область","Челябинск"));

foreach(var elem in cities)

{

elem.Info();

}

Console.ReadKey();

}

}

13 задание

class Animal

{

public string Name { get; set; }

public string Coating { get; set; }

public Animal() { }

public Animal(string name, string coating)

{

Name = name;

Coating = coating;

}

virtual public void Info()

{

Console.WriteLine($"Название: {Name}\nТип покрытия:{Coating}");

}

}

class Amphibians : Animal

{

public int Count { get; set; }

public Amphibians() { }

public Amphibians(string name, string coating, int count):base(name,coating)

{

Count = count;

}

public override void Info()

{

base.Info();

Console.WriteLine($"Количество пар лап: {Count}");

}

}

class Mammals : Animal

{

public int Speed { get; set; }

public Mammals() { }

public Mammals(string name, string coating, int speed) : base(name, coating)

{

Speed = speed;

}

public override void Info()

{

base.Info();

Console.WriteLine($"Скорость передвижения:{Speed} км/ч");

}

}

internal class Program

{

static void Main(string[] args)

{

List<Animal> list = new List<Animal>();

list.Add(new Amphibians("Лягушка","Кожа",2));

list.Add(new Mammals("Медведь","Шерсть",48));

foreach(var elem in list)

{

elem.Info();

}

Console.ReadKey();

}

}

**Вывод:** Разработал иерархию классов. Выделил родительский класс. Описал конструкторы в каждом классе. Создал перегруженные и виртуальные методы, а также методы скрывающие методы родителя. Продемонстрировал работу объектов классов в приложении.