Тимофеева Наталья

М8О-202Б-19

ООП C#

Лабораторная работа № 4

**Описание работы**

НАСЛЕДОВАНИЕ: РАСШИРЕНИЕ, СПЕЦИФИКАЦИЯ, СПЕЦИАЛИЗАЦИЯ, КОНСТРУИРОВАНИЕ, КОМБИНИРОВАНИЕ

1. Представьте граф древовидной структуры как иерархию “Общее-Частное”. Пусть объект суперкласса – класс А. Поставьте соответствующие стрелки на графе.

2. Рассматривайте в одном проекте категории: расширение, cпецификация, cпециализация, конструирование выделив их на графе. В чем достоинства и недостатки каждой из этих категорий?

3. Реализуйте в отдельном проекте, только для вершин соединенных соответствующим образом на графе укажите соответствующие стрелки и реализуете категорию наследование комбинирование (множественное наследование). Какие конфликты возникают при множественном наследовании?

4. Оформите работу. Каждому проекту должен соответствовать свой граф, на которых должен быть указан вид соответствующих категорий. Напишите ответы на вопросы к пункту 2 и 3. Сохраните результаты лабораторной.

**Программа**

using System;

namespace Lab4\_1 {

class Program {

interface Animal {

void protection\_status();

}

interface Habits {

void food();

void place();

}

class Tiger : Animal {

public Tiger() { }

public void protection\_status() { Console.WriteLine("Endangered species"); }

virtual public void color() { Console.WriteLine("Red"); }

}

class WhiteTiger : Tiger {

public WhiteTiger() { }

override public void color() { Console.WriteLine("White"); }

}

class Kitten : WhiteTiger {

public Kitten() { }

public void noise() { Console.WriteLine("Meow"); }

public int quantity { get; set; }

}

class Bird {

public int speed { get; set; }

public Bird() { }

public Bird(int num) {

speed = num;

}

public void features() {

Console.WriteLine("Birds have wings and beak");

}

}

class Goose : Bird, Animal {

public Goose() { }

public Goose(int num):base(num) { }

public void protection\_status() { Console.WriteLine("Out of danger"); }

virtual public void swim() { Console.WriteLine("Goose like to swim"); }

}

class Sparrow : Goose, Habits {

public Sparrow() { }

public void food() { Console.WriteLine("Bread"); }

public void place() { Console.WriteLine("City"); }

override public void swim() { Console.WriteLine("Sparrow doesn't like to swim"); }

public void size() { Console.WriteLine("Small"); }

}

static void Main() {

Tiger Amur = new Tiger();

Amur.protection\_status();

Amur.color();

WhiteTiger Timur = new WhiteTiger();

Timur.protection\_status();

Timur.color();

Kitten Mur = new Kitten();

Mur.protection\_status();

Mur.color();

Mur.noise();

Mur.quantity = 3;

Console.WriteLine("This quantity of kitten of white tiger is {0}", Mur.quantity);

Goose Grey = new Goose(10);

Grey.protection\_status();

Grey.features();

Console.WriteLine(Grey.speed);

Grey.swim();

Sparrow Captain = new Sparrow();

Captain.protection\_status();

Captain.features();

Captain.speed = 7;

Console.WriteLine("Speed of Captain is {0}", Captain.speed);

Captain.food();

Captain.place();

Captain.swim();

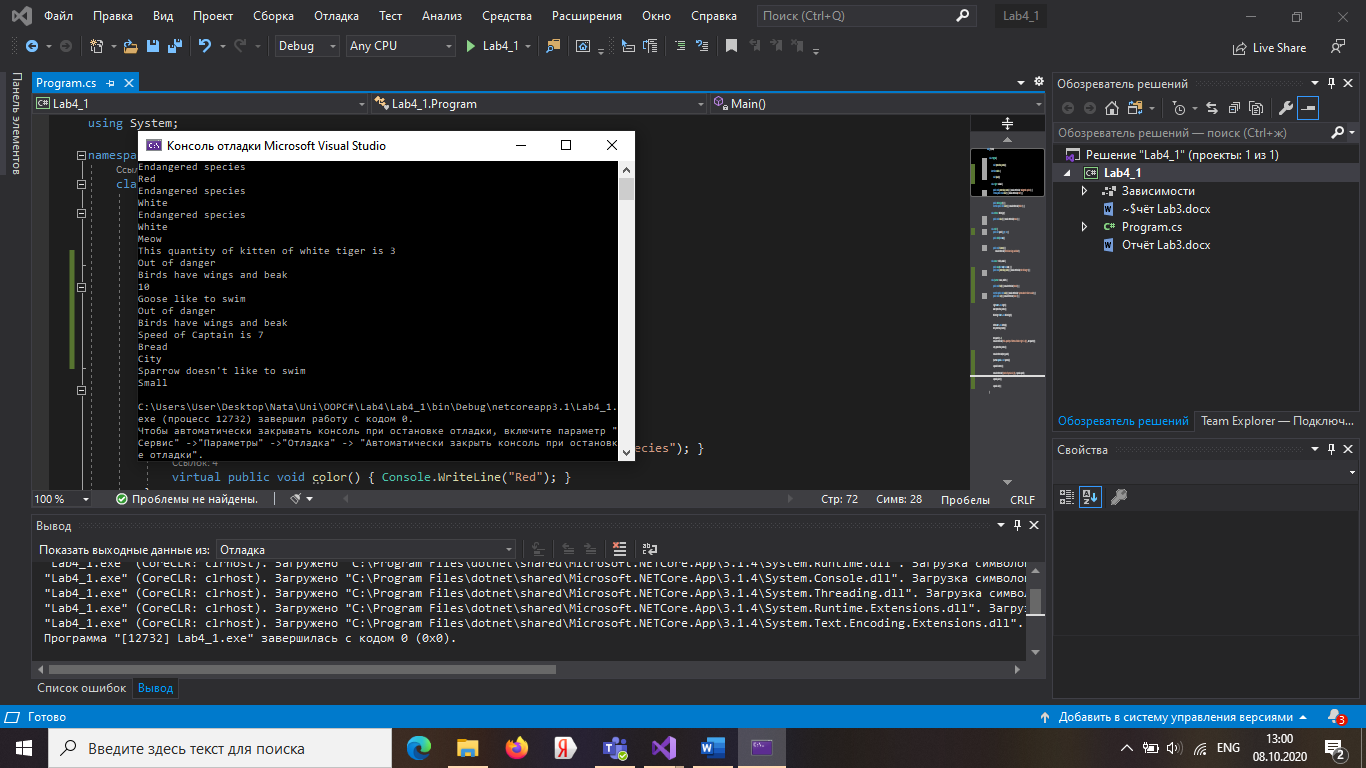
Captain.size();

}

}

}

**Вывод с консоли**



**Вывод**

Специализация: дочерний класс удовлетворяет спецификациям родителя во всех существенных моментах, т.е. его можно использовать вместо родительского класса.

Спецификация: родительский класс описывает поведение, которое реализуется в дочернем классе, но оставлено нереализованным в родительском.

Расширение: дочерний класс добавляет новые функциональные возможности к родительскому классу, но не меняет наследуемое поведение. В отличие от специализации при расширении дочерний класс не переопределяет ни одного метода базового класса, а добавленные методы слабо связаны с существующими методами родителя.

Комбинирование: для комбинирования классов используется механизм множественного наследования.

Конструирование: используются все методы, описанные выше.