|  |
| --- |
| Федеральное государственное бюджетное образовательное учреждение высшего образования |
| Югорский государственный университет |
| Институт цифровой экономики  (ИЦЭ) |
|  |

ОТЧЕТ

по лабораторной работе №4

Тема:

«Полиморфизм и перегрузка»

По дисциплине:

«Языки программирования высокого уровня»

Выполнила: студентка группы 1182б

Гаджиева Айсель Лазим кызы

Проверил: доцент

Сафонов Егор Иванович

Ханты-Мансийск – 2019 г.

# Цель работы

Выполнение заданий по полиморфизму и перегрузке! Определение класса, создание виртуального метода с дальнейшим переопределением в других классах. Написать «перегрузку методов».

# Постановка задачи

1. Задачи на полиморфизм
2. Определить два класса Кит и Корова.
3. Переопределить метод getName в классе Whale(Кит), чтобы программа выдавала: Я не корова, Я – кит.
4. Написать метод, который определяет, объект какого класса ему передали, и выводит на экран одну из надписей: Кошка, Собака, Птица, Лампа.
5. Создать два класса Cat(кот) и Dog(собака), класс Dog(собака) происходит от Cat(кот). Определить метод getChild в классах Cat(кот) и Dog(собака), чтобы кот порождал кота, а собака – собаку.
6. Задачи на перегрузку методов
7. Написать два метода: print(int) и print(String).
8. Написать два метода: print(int) и print(Integer).Написать такой код в методе main, чтобы вызвались они оба.
9. Написать пять методов print с разными параметрами.
10. Написать public static методы: int min(int, int), long min(long, long), double min(double, double). Каждый метод должен возвращать минимальное из двух переданных в него чисел.
11. Написать public static методы: int max(int, int), long max (long, long), double max (double, double). Каждый метод должен возвращать максимальное из двух переданных в него чисел.

# Выполнение работы

1.

1. Определение 2-х классов.
2. Переопределение метода GetName в классе Whale.
3. Написание метода для определения имени класса.
4. Создание классов и методов для них.

2.

1. Создание нескольких методов с одним и тем же именем, но разной сигнатурой

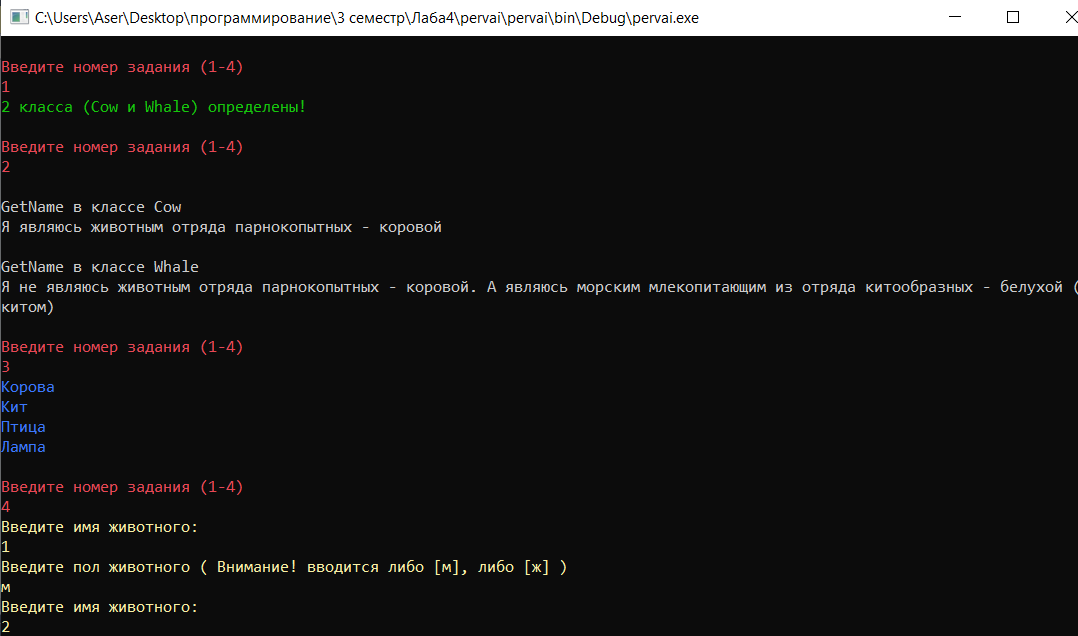


Рисунок 1. Вывод на консоль.

Листинг кода 1 – Код:

using System;

namespace Laboratory4

{

class Program

{

static void Main(string[] args)

{

uint a;

uint A;

string s;

bool b;

Cow[] warrior = new Cow[4];

warrior[0] = new Cow();

warrior[1] = new Whale();

warrior[2] = new Bird();

warrior[3] = new Lamp();

while (true)

{

do

{

Console.ForegroundColor = ConsoleColor.Red;

Console.WriteLine(" \nВведите номер задания (1-4)");

s = Console.ReadLine();

b = uint.TryParse(s, out A);

Console.ResetColor();

} while (b == false || A == 0 || A > 4);

switch (A)

{

case 1:

Console.ForegroundColor = ConsoleColor.Green;

Console.WriteLine("2 класса (Cow и Whale) определены!");

Console.ResetColor();

break;

case 2:

Console.WriteLine("\nGetName в классе Cow");

warrior[0].GetName();

Console.WriteLine("\nGetName в классе Whale");

warrior[1].GetName();

break;

case 3:

Console.ForegroundColor = ConsoleColor.Blue;

GetName(warrior[0]);

GetName(warrior[1]);

GetName(warrior[2]);

GetName(warrior[3]);

Console.ResetColor();

break;

case 4:

Console.ForegroundColor = ConsoleColor.Yellow;

Cat[] army = new Cat[4];

army[0] = new Cat();

army[1] = new Cat();

army[2] = new Dog();

army[3] = new Dog();

if (army[0].logical\_gender != army[1].logical\_gender)

{

Console.WriteLine("Они могут иметь потомство");

if (army[0].logical\_gender == 'ж')

{

army[0].GetChild();

}

else

{ army[1].GetChild();}

}

else { Console.WriteLine("У них не может быть потомства! :(");}

if (army[2].logical\_gender != army[3].logical\_gender)

{

Console.WriteLine("Они могут иметь потомство");

if (army[2].logical\_gender == 'ж')

{

army[2].GetChild();

}

else

{ army[3].GetChild();}

}

else

Console.WriteLine("У них не может быть потомства! :(");

Console.WriteLine();

Console.WriteLine("Хотите поменять пол животному? ([Д]-Да, [Н]-Нет)");

string selection = Console.ReadLine();

switch (selection)

{

case "Д":

Console.WriteLine("Какому животному вы хотите поменять пол? Внимание! Введите номер животного! (от 0 до 3)!");

while (!uint.TryParse(Console.ReadLine(), out a) || a >= 4) ;

if (army[a].logical\_gender == 'м')

army[a].logical\_gender = 'ж';

else

army[a].logical\_gender = 'м';

break;

case "Н":

Console.WriteLine("Пол не был изменён! :(");

break;

default:

Console.WriteLine("Вы нажали неизвестную букву");

break;

}

Console.ResetColor();

break;

}

}

}

static public void GetName(Object o)

{

if (o.GetType().Name == "Cow")

{

Console.WriteLine("Корова");

}

if (o.GetType().Name == "Whale")

{

Console.WriteLine("Кит");

}

if (o.GetType().Name == "Bird")

{

Console.WriteLine("Птица");

}

if (o.GetType().Name == "Lamp")

{

Console.WriteLine("Лампа");

}

}

static void Print(int a)

{

Console.WriteLine(a + " - это целое число");

}

static void Print(string a)

{

Console.WriteLine(a + " - это строка");

}

static void Print(string a, int b)

{

Console.WriteLine($"{a} - это строка, {b} - это целое число");

}

static void Print(int a, string b)

{

Console.WriteLine($"{a} - это целое число, {b} - это строка");

}

static void Print(ref double a)

{

Console.WriteLine(a + " - это вещественное число");

}

}

class Cow

{

public virtual void GetName()

{

Console.WriteLine("Я являюсь животным отряда парнокопытных — коровой");

}

}

class Whale : Cow

{

public override void GetName()

{

Console.WriteLine("Я не являюсь животным отряда парнокопытных — коровой. А являюсь морским млекопитающим из отряда китообразных — белухой (китом)");

}

}

class Cat

{

public char logical\_gender;

protected string name;

public Cat()

{

Console.WriteLine("Введите имя животного:");

name = Console.ReadLine();

Console.WriteLine("Введите пол животного ( Внимание! вводится либо [м], либо [ж] )");

while (!char.TryParse(Console.ReadLine(), out logical\_gender) || (logical\_gender != 'ж' && logical\_gender != 'м')) ;

}

public virtual Cat GetChild()

{

return new Cat();

}

}

class Dog : Cat

{

public override Cat GetChild()

{

return new Dog();

}

}

class Bird : Cow

{

}

class Lamp : Cow

{

}

class MaximumMinimum

{

public static int Min(int a, int b)

{

if (a < b)

return a;

else return b;

}

public static long Min(long a, long b)

{

if (a < b)

return a;

else return b;

}

public static double Min(double a, double b)

{

if (a < b)

return a;

else return b;

}

public static int Max(int a, int b)

{

if (a > b)

return a;

else return b;

}

public static long Max(long a, long b)

{

if (a > b)

return a;

else return b;

}

public static double Max(double a, double b)

{

if (a > b)

return a;

else return b;

}

}

}

# Заключение

В ходе выполнения лабораторной работы была достигнута цель.