Министерство образования и науки Украины

Харьковский национальный университет радиоэлектроники

Кафедра Системотехники

Отчет по лабораторной работе №3

с курса: «Программирование под платформу .NET»

на тему: «Классы. Перегрузка методов. Полиморфизм. Наследование»

Выполнил: Проверила:

Ст. гр. ИТКНу-17-1 Жернова П. Е.

Гончаренко В.А.

Харьков 2019

Цель: Научиться создавать собственные классы и объекты; реализовать принцип подражания и перегрузки методов.

Ход выполнения работы

Задание 1. Создать класс Телевизор, в котором есть поле текущий канал. предусмотрите в нем возможность переключения каналов (методы класса): следующий канал, предыдущий канал, переход к каналу по номеру. Учтите, что канал не может иметь отрицательный номер.

Код программы:

public class TV {

private int \_currentChannel = 0;

public enum Channels {

Channel1, Channel2, Channel3, Channel4, Channel5

}

public TV() {

}

public Channels NextChannel() {

if (\_currentChannel < 4)

\_currentChannel++;

else

\_currentChannel = 0;

return (Channels)\_currentChannel;

}

public Channels PreviousChannel() {

if (\_currentChannel > 0)

\_currentChannel--;

else

\_currentChannel = 4;

return (Channels)\_currentChannel;

}

public Channels GetChannelsById(int id) {

if (id >= 0 && id <= 4)

\_currentChannel = id;

else

Console.WriteLine("Channel not found!");

return (Channels)\_currentChannel;

}

}

...

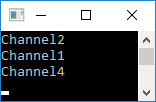
var tel = new TV();

Console.WriteLine(tel.NextChannel());

Console.WriteLine(tel.PreviousChannel());

Console.WriteLine(tel.GetChannelsById(3));

Результат:



Задание 2. Создать класс Студент, задайте в нем поля: имя, курс, есть ли у него Стипендия. Создайте у класса несколько конструкторов, для возможности задания сразу всех Указанным параметров или нескольких при создании объекта класса.

Код программы:

public class Student {

private string \_name;

private int \_course;

private bool \_grant;

public Student(string name) {

\_name = name;

}

public Student(int course) {

\_course = course;

}

public Student(bool grant) {

\_grant = grant;

}

public Student(string name, int course, bool grant) {

\_name = name;

\_course = course;

\_grant = grant;

} }

Задание 3. Создать класс Аудиоплеер, в котором есть поле громкость звука, для доступа к этому полю реализуйте свойство. Громкость в диапазоне от 0 до 100.

Код программы:

public class Audioplayer {

private int \_volume;

public int Volume {

get {

return \_volume;

}

set {

if(value > 100) {

\_volume = 100

} else if(value < 0) {

\_volume = 0;

} else {

\_volume = value;

}

}

}

var ipod = new Audioplayer();

ipod.Volume = 20;

ipod.Volume = 120;

Задание 4. Напишите программу - эмулятор холодильника. Программа должна позволять выполнять через меню в консольном приложении те или иные действия с холодильником, например, включить или выключить его, и т.п. При этом после каждого действия пользователя, программа должна отобразить на экране текущее состояние холодильника и меню всех функций. Класс холодильника должен иметь метод, который отображает текущее состояние, а также методы, которые позволяют изменить это состояние. Режимы работы холодильника должны быть описаны с помощью перечисления (enum). главный метод программы (Main) должен выполнять задачу отображения меню и передачу команд пользователя объекта класса холодильник.

Код программы:

public class Fridge {

public string Name { get; set; }

public FridgeMode Mode { get; set; }

public List<string> Food { get; set; }

public enum FridgeMode {

SwitchOn, SwitchOff, SwitchCold

}

public Fridge() {

Food = new List<string>();

}

public Fridge(string name, FridgeMode mode, params string[] food) {

Food = new List<string>();

Name = name;

Mode = mode;

Food = food.ToList();

}

public void FridgeStatus() {

Console.WriteLine($"Fridge name = {Name}");

Console.WriteLine($"Fridge mode = {Mode}");

Console.WriteLine($"Frigde storage: ");

foreach(var item in Food) {

Console.WriteLine($"{item}");

}

Console.WriteLine();

}

public void SwitchOn() {

Mode = FridgeMode.SwitchOn;

FridgeStatus();

}

public void SwitchOff() {

Mode = FridgeMode.SwitchOff;

FridgeStatus();

}

public void SwitchCold() {

Mode = FridgeMode.SwitchCold;

FridgeStatus();

}

public void PutFood(params string[] food) {

Food.AddRange(food);

FridgeStatus();

}

public void GetFood(params string[] food) {

foreach(var item in food) {

if (Food.Contains(item))

Food.Remove(item);

else

Console.WriteLine($"There are no {item} in the fridge!");

}

FridgeStatus(); }

}

....

var pobeda = new Fridge();

pobeda = new Fridge("Pobeda", Fridge.FridgeMode.SwitchOn, "Beer", "Cake", "Fish");

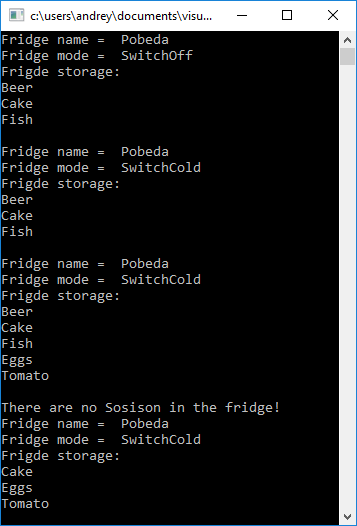
pobeda.SwitchOff();

pobeda.SwitchCold();

pobeda.PutFood("Eggs", "Tomato");

pobeda.GetFood("Beer", "Fish", "Sosison");

Результат:



Задание 5. Создать класс Круга и реализуйте следующий функционал:

1. Перегрузить конструктор:

- пустой конструктор;

- спрашивает координаты центра круга, его радиус и инициализирует объект.

2. Перегрузить груз метод получения длины круга

- метод без параметров возвращает длину круга для текущего объекта;

- метод принимает радиус и возвращает длину круга для заданного радиуса.

3. Перегрузить метод получения объекта-круга

- метод без параметров возвращает копию текущего объекта круга;

- метод принимает координаты центра круга, его радиус и возвращает объект круга с заданными параметрами.

4. Метод проверки попадания точки в круг.

5. Метод преобразования текущего состояния объекта в символьную строку.

Код программы:

public class Circle {

public int X { get; set; }

public int Y { get; set; }

public int Radius { get; set; }

public Circle() {

}

public Circle(int x, int y, int radius) {

X = x;

Y = y;

Radius = radius;

}

public double GetLenght() {

return 2 \* Math.PI \* Radius;

}

public double GetLenght(int radius) {

return 2 \* Math.PI \* radius;

}

public Circle GetCircleObject() {

return this;

}

public Circle GetCircleObject(int x, int y, int radius) {

return new Circle(x, y, radius);

}

public string CheckDot(int x, int y) {

double result = Math.Pow(X - x, 2) + Math.Pow(Y - y, 2);

if (result <= Math.Pow(Radius, 2)) {

return "Point is in the circle!";

}

return "Point is NOT in the circle!";

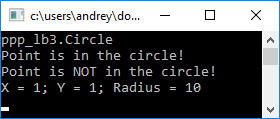
}

public string GetInfo() {

return "X = " + X + "; Y = "+ Y + "; Radius = " + Radius;

} }

Результат:



Задание 6. Создать класс Геометрическая фигура. Создайте в нем общие поля / свойства, например, координаты центра фигуры (с помощью конструктора должна быть возможность задать центр). На основе этого класса создайте два новых - Треугольник и круг. В этих классах должны быть свои особые поля. В оба класса добавьте метод Draw (), в котором существует специфическая логика рисования фигуры.

Код программы:

public class Figure {

public int X { get; set; }

public int Y { get; set; }

public Figure(int x, int y) {

X = x;

Y = y;

} }

public class Triangle : Figure {

public int Side { get; set; }

public Triangle(int side, int x, int y) : base(x,y) {

Side = side;

}

public void Draw() {

Console.WriteLine($"\*treugolnek s centrom v x = {X}, y = {Y} i storonoi = {Side}\*");

} }

public class Krug : Figure {

public int Radius { get; set; }

public Krug(int radius, int x, int y) : base(x, y) {

Radius = radius;

}

public void Draw() {

Console.WriteLine($"\*kruzhochek s centrom v x = {X}, y = {Y} i radiusom = {Radius}\*");

} }

...

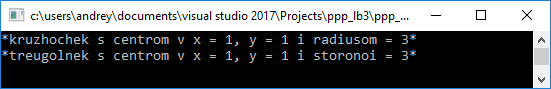
var triangle = new Triangle(3, 1, 1);

var mihail = new Krug(3, 1, 1);

mihail.Draw();

triangle.Draw();

Результат:



Задание 7. Создать класс Квадрат, создайте свойство для хранения значения стороне и виртуальный метод, который возвращает периметр. На основе этого класса создайте класс Куб и переопределите метод получения периметра.

Код программы:

public class Square {

public int Side { get; set; }

public Square(int side) {

Side = side;

}

public virtual int GetPerim() {

return Side \* 4;

} }

public class Coube : Square {

public Coube(int side) : base(side) {

}

public override int GetPerim() {

return Side \* 12; }

}

...

var square = new Square(3);

Console.WriteLine($"Square perimetr = {square.GetPerim()}.");

var coube = new Coube(3);

Console.WriteLine($"Coube perimetr = {coube.GetPerim()}.");

Console.ReadKey();

Задание 8. Создать класс Прямоугольник с полями координат верхнего левого (Х1, Y1) и нижнего правого (Х2, Y2) углов. Переопределите в нем методы ToString, Equals и GetHashCode. Прямоугольники равны, если у них одинаковые координаты левого верхнего и нижнего правого углов.

Код программы:

public class Rectangle {

private int X1, X2, Y1, Y2;

public Rectangle(int x1, int y1, int x2, int y2) {

X1 = x1;

Y1 = y1;

X2 = x2;

Y2 = y2;

}

public override int GetHashCode() {

return X1 + X2 + Y1 + Y2;

}

public override bool Equals(object obj) {

var currentRectangle = obj as Rectangle;

if(currentRectangle != null) {

if (currentRectangle.X1 == X1 && currentRectangle.Y1 == Y1 && currentRectangle.X2 == X2 && currentRectangle.Y2 == Y2)

return true;

else

return false;

}

return false;

}

public override string ToString() {

return $"Rectangle with X1 = {X1}, Y1 = {Y1}, X2 = {X2}, Y2 = {Y2}";

}

}

....

var rt1 = new Rectangle(1, 1, 5, 5);

var rt2 = new Rectangle(1, 1, 5, 5);

var rt3 = new Rectangle(1, 2, 3, 5);

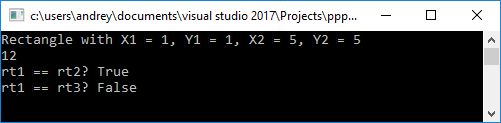
Console.WriteLine(rt1.ToString());

Console.WriteLine(rt1.GetHashCode());

Console.WriteLine($"rt1 == rt2? {rt1.Equals(rt2)}");

Console.WriteLine($"rt1 == rt3? {rt1.Equals(rt3)}");

Результат:



Выводы:

На данной практической работе я научился пользоваться классами, наследованием, полиморфизмом и перегрузкой методов.