# Lab: Polymorphism

Problems for exercises and homework for the ["C# OOP Basics" course @ SoftUni](https://softuni.bg/courses/csharp-oop-basics)".

You can check your solutions here: [https://judge.softuni.bg/Contests/Compete/Index/680#0](https://judge.softuni.bg/Contests/Compete/Index/680%230).

## MathOperations

Create a class **MathOperations**, which should have 3 times method Add(). Method Add() have to be invoked with:

* Add(int, int): **int**
* Add(double, double, double): **double**
* Add(decimal, decimal, decimal): **decimal**

You should be able to use the class like this:

|  |
| --- |
| StartUp.cs |
| public static void Main()  {  MathOperations mo = new MathOperations();  Console.WriteLine(mo.Add(2, 3));  Console.WriteLine(mo.Add(2.2, 3.3, 5.5));  Console.WriteLine(mo.Add(2.2m, 3.3m, 4.4m));  } |

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
|  | 5  11  9.9 |

### Solution

The MathOperation class should look like this:



## Animals

Create a class **Animal**, which hold **two properties**:

* Name: string
* FavouriteFood: string

Animal have one virtual method ExplainSelf()**: string**You should add two new classes **Cat** and **Dog.** There override ExplainSelf() method by adding concrete animal sound on new line. (Look at examples below)

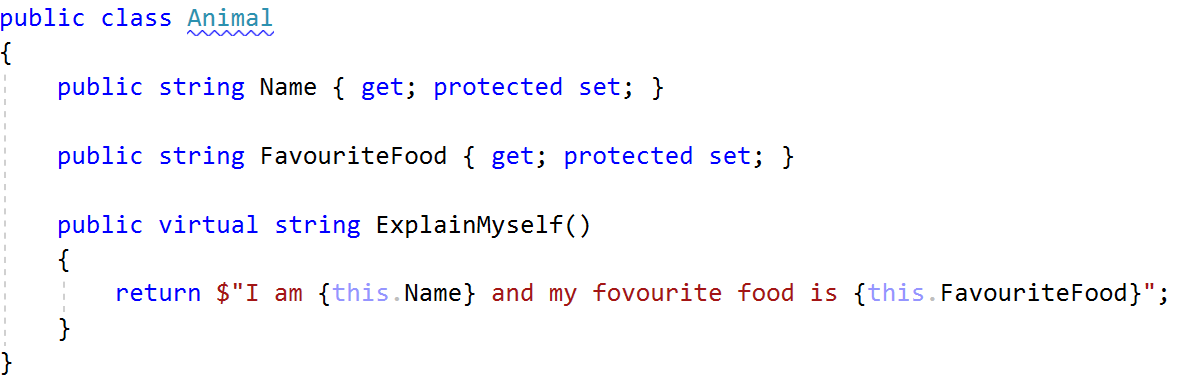
You should be able to use the class like this:

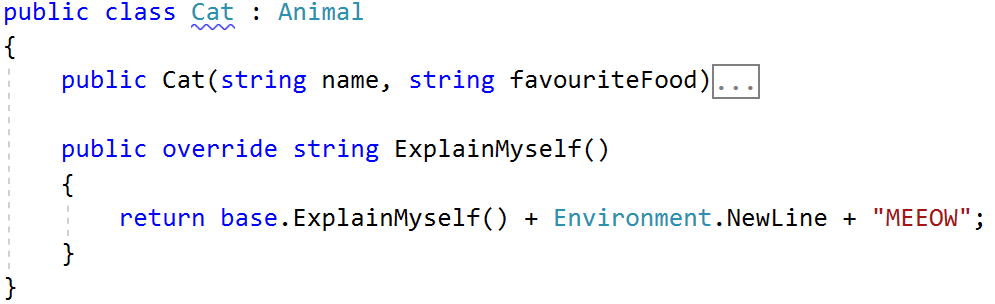
|  |
| --- |
| StartUp.cs |
| Animal cat = new Cat("Pesho", "Whiskas");  Animal dog = new Dog("Gosho", "Meat");  Console.WriteLine(cat.ExplainMyself());  Console.WriteLine(dog.ExplainMyself()); |

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
|  | I am Pesho and my favourite food is Whiskas  MEEOW  I am Gosho and my fovourite food is Meat  DJAAF |

### Solution





## Shapes

Create class hierarchy, starting with **abstract** class Shape:

* **Abstract methods:**
  + CalculatePerimeter(): double
  + CalculateArea(): double
* **Virtual methods**:
  + Draw(): string

Create **two children** of the Shape class:

* **Rectangle**
* **Circle**

Each of them need to have:

* **Fields:**
  + **height and width for Rectangle**
  + **radius for Circle**
* **Public properties with encapsulation for the fields**
* **Public constructor**
* **Concrete methods for calculations (perimeter and area)**
* **Override methods for drawing**

The finished class should be able to be used like this:

|  |
| --- |
| StartUp.cs |
| public static void Main()  {  var shapes = new List<Shape>();  shapes.Add(new Circle(3.5));  shapes.Add(new Rectangle(3.5, 1.2));  shapes.Add(new Rectangle(1.5, 1.5));  shapes.Add(new Rectangle(3.4, 1.5));  shapes.Add(new Circle(3.6));  foreach (var shape in shapes)  {  Console.WriteLine(shape.Draw());  }  } |