# Homework: Encapsulation and Polymorphism

This document defines the homework assignments from the ["OOP" Course @ Software University](https://softuni.bg/trainings/coursesinstances/details/8). Please submit as homework a single zip / rar / 7z archive holding the solutions (source code) of all below described problems. The solutions should be written in C#.

## Shapes

* Define an interface **IShape** with two abstract methods: **CalculateArea()** and **CalculatePerimeter()**.
* Define an abstract class **BasicShape** implementing **IShape** and holding **width** and **height**. Leave the methods **CalculateArea()** and **CalculatePerimeter()** abstract.
* Define two new **BasicShape** subclasses: **Triangle** and **Rectangle** that implement the abstract methods **CalculateArea()** and **CalculatePerimeter()**.
* Define a class **Circle** with a suitable constructor and **IShape**.
* Create an array of different shapes (**Circle**, **Rectangle**, **Triangle**) and test the behavior of the **CalculateSurface()** and **CalculatePerimeter()** methods.

## Bank of Kurtovo Konare

A bank holds different types of accounts for its customers: **deposit accounts**, **loan accounts** and **mortgage accounts**.

* Customers could be **individuals** or **companies**.
* All accounts have **customer**, **balance** and **interest rate** (monthly based).
* **Deposit** accounts are allowed to deposit and withdraw money. **Loan** and **mortgage** accounts can only deposit money.
* All accounts can calculate their interest for a given period (in months) using the formula

***A = money \* (1 + interest rate \* months)***

* **Loan** accounts have no interest for the first **3 months** if held by **individuals** and for the first **2 months** if held by a **company**.
* **Deposit** accounts have no interest if their balance is positive and less than 1000.
* **Mortgage** accounts have ½ interest for the first 12 months for **companies** and no interest for the first 6 months for **individuals**.

Write a program to model the bank system with classes and interfaces. You should identify the classes, interfaces, base classes and abstract actions and implement the calculation of the interest functionality through overridden methods. Write a program to demonstrate that your classes work correctly.