

Лабораторная работа № 1

Основы объектно-ориентированного программирования

Цель работы

Изучить основы объектно-ориентированного программирования (ООП), ознакомиться с интерфейсом среды разработки MS Visual Studio, научиться программировать простейшие классы и использовать основные принципы ООП.

Основные принципы

Любой объектно-ориентированный язык должен отвечать трем основным принципам:

- 1) Инкапсуляция: Как данный язык скрывает детали внутренней реализации объектов и предохраняет целостность данных?
- 2) Наследование: Как данный язык стимулирует многократное использование кода?
- 3) Полиморфизм: Как данный язык позволяет трактовать связанные объекты сходным образом?

Инкапсуляция

Инкапсуляция — это механизм программирования, объединяющий вместе код и данные, которыми он манипулирует, исключая как вмешательство извне, так и неправильное использование данных. В объектно-ориентированном языке данные и код могут быть объединены в совершенно автономный черный ящик. Внутри такого ящика находятся все необходимые данные и код. Когда код и данные связываются вместе подобным образом, создается **объект**.

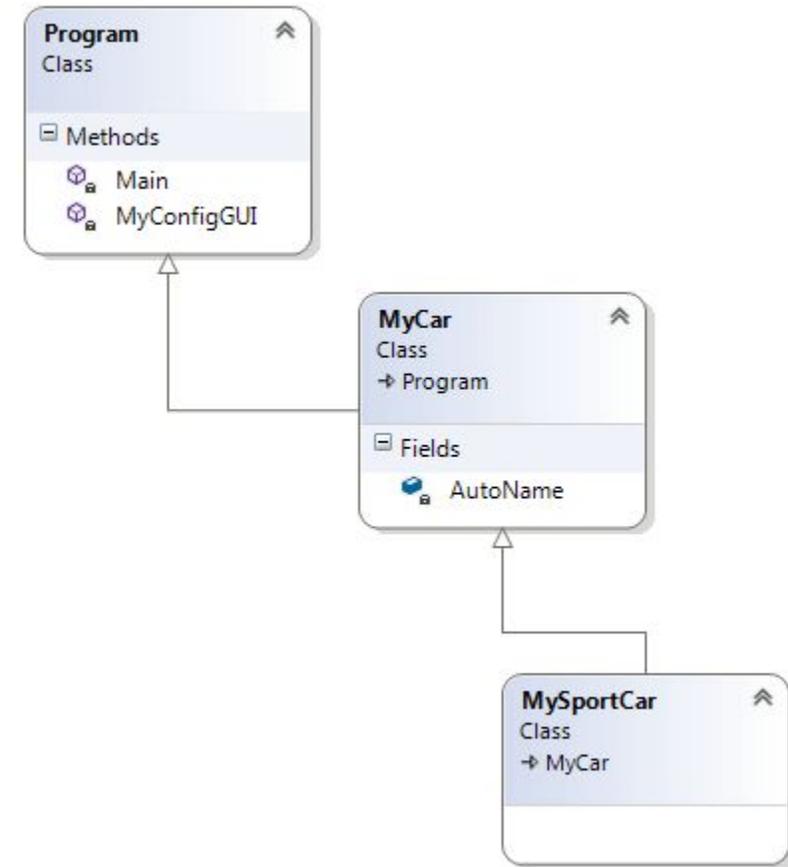
Основной единицей инкапсуляции в C# является **класс**, который определяет форму объекта. Он описывает данные, а также код, который будет ими оперировать.

Код и данные, составляющие вместе класс, называют *членами*. Данные, определяемые классом, называют *полями*, или *переменными экземпляра*. А код, оперирующий данными, содержится в *функциях-членах*. самым типичным представителем

Наследование

Следующий принцип ООП — **наследование** — касается способности языка позволять строить новые определения классов на основе определений существующих классов. По сути, наследование позволяет расширять поведение базового (или родительского) класса, наследуя основную функциональность в производном подклассе (также именуемом дочерним классом).

<https://msdn.microsoft.com/ru-ru/library/ms173149.aspx>



Полиморфизм

Последний принцип ООП — **полиморфизм**. Он обозначает способность языка трактовать связанные объекты в сходной манере. В частности, этот принцип ООП позволяет базовому классу определять набор членов (формально называемый *полиморфным интерфейсом*), которые доступны всем наследникам. Полиморфный интерфейс класса конструируется с использованием любого количества виртуальных или абстрактных членов.

По сути, *виртуальный член* — это член базового класса, определяющий реализацию по умолчанию, которая может быть изменена (или, говоря более формально, переопределена) в производном классе. В отличие от него, *абстрактный метод* — это член базового класса, который не предусматривает реализации по умолчанию, а предлагает только сигнатуру. Когда класс наследуется от базового класса, определяющего абстрактный метод, этот метод обязательно должен быть переопределен в производном классе. В любом случае, когда производные классы переопределяют члены, определенные в базовом классе, они по существу переопределяют свою реакцию на один и тот же запрос.

Поиск информации

MSDN - <https://msdn.microsoft.com/ru-ru/dn308572.aspx>

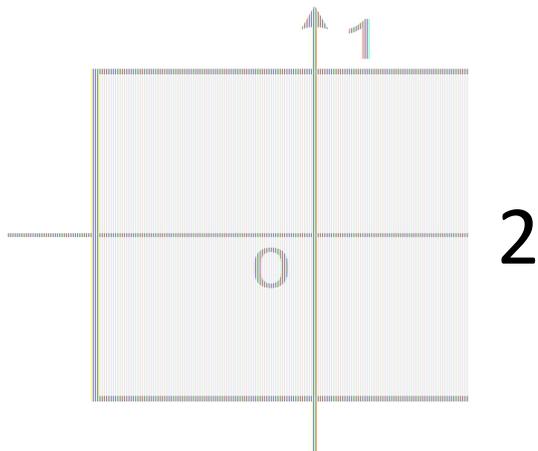
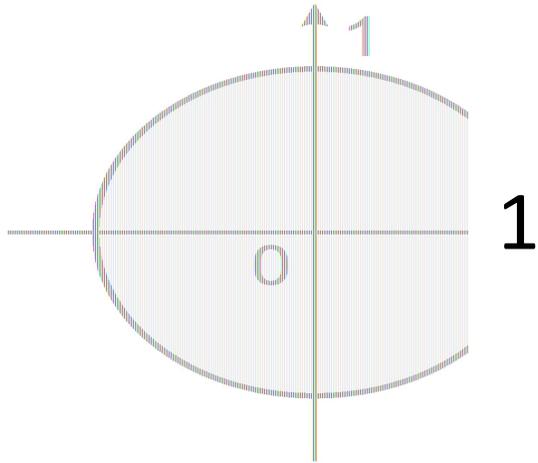
Самоучитель 1 -

<http://nullpro.info/2013/samouchitel-po-c-dlya-nachinayushhix-01-osnovy-yazyka-peremennye-logika-cikly/>

Справочник с примерами -

http://professorweb.ru/my/csharp/charp_theory/level3/3_2.php

Задание



Составить две программы, выполняющие следующее: Пользователь вводит координаты точки X и Y . Программа проверяет, входит ли данная точка в область определения, отмеченную на графике серой областью. Необходимо использовать класс для графика, его конструктор и методы.

1) Дополнительное задание (+1): объединить две программы, предоставив пользователю выбор между областями определения «Круг» и «Квадрат».

2) Дополнительное задание (+2): объединить две программы как в 2, реализовать иерархию областей определения через абстрактный класс `Function`

Требования к оформлению отчета

- 1) Шапка (рис. ниже)
- 2) Цель работы
- 3) Постановка задачи
- 4) Алгоритм программы
- 5) Исходный код программы
- 6) Экранные формы

	ФИО	
	Дата	
	Отчет	