**Виртуальные функции**

Использование **виртуальных** функций в программах является

одним из способов реализации принципа полиморфизма ООП, когда в

дочернем классе переопределяется метод родительского класса.

**Этот механизм основывается на возможности хранения в переменной, являющейся ссылкой на базовый класс, адреса объекта производного класса.**

По умолчанию выбор вызываемой функции осуществляется в

соответствии с типом ссылки. В случае вызова виртуальной функции

через ссылку на базовый класс программа будет использовать метод,

определенный для типа объекта, а не метод, определенный для типа

ссылки (реализация принципа позднего связывания).

Объявление виртуального метода в базовом классе Base Class

осуществляется добавлением в начало объявления метода ключевого

слова virtual, например:

Продемонстрируем работу виртуального метода на примере:

BaseClass bc = new BaseClass();

DerivedClass dc= new DerivedClass();

BaseClass bcRef; // ссылка на базовый класс

bcRef = bc; // ссылке на базовый класс присваивается

// адрес объекта класса BaseClass

bcRef.Method(); // вызов метода Method() класса BaseClass

bcRef = dc; // ссылке на базовый класс

// присваивается адрес объекта

// класса DerivedClass

bcRef.Method(); // вызов метода Method()

// класса DerivedClass

В данном случае две одинаковые строки кода будут приводить к

вызову двух разных методов: базового и производного классов.

Конструкторы классов не могут являться виртуальными

функциями, поскольку производным классом не наследуется

конструктор базового класса.

Виртуальные функции могут быть перекрыты в производных

классах только виртуальными функциями. В этом случае метод

производного класса не считается переопределенным и будет

вызываться только для объекта производного класса.

**Абстрактные классы**

Некоторые сущности, которые мы хотим выразить с помощью языка программирования, могут не иметь конкретного воплощения. Например, в реальности не существует геометрической фигуры как таковой. Есть круг, прямоугольник, квадрат, но просто фигуры нет. Однако же и круг, и прямоугольник имеют что-то общее и являются фигурами. И для описания подобных сущностей, которые не имеют конкретного воплощения, предназначены абстрактные классы.

Абстрактный класс похож на обычный класс. Он также может иметь переменные, методы, конструкторы, свойства. Единственное, что при определении абстрактных классов используется ключевое слово abstract.