*Класс* представляет собой шаблон, по которому определяется форма объекта. В нем указываются данные и код, который будет оперировать этими данными. В С# используется спецификация класса для построения *объектов,* которые являются *экземплярами* класса. Следовательно, класс, по существу, представляет собой ряд схематических описаний способа построения объекта. При этом очень важно подчеркнуть, что класс является логической абстракцией. Физическое представление класса появится в оперативной памяти лишь после того, как будет создан объект этого класса.

**Общая форма определения класса**

При определении класса объявляются данные, которые он содержит, а также код,

оперирующий этими данными. Вообще говоря, данные содержатся в *членах данных,* определяемых классом, а код — в *функциях-членах.* Следует сразу же подчеркнуть, что в C# предусмотрено несколько разновидностей членов данных и функций-членов. Например, к членам данных, называемым также *полями,* относятся переменные экземпляра и статические переменные, а к функциям-членам — методы, конструкторы, деструкторы, индексаторы, события,

операторы и свойства.

Класс создается с помощью ключевого слова ***class***. В ***отличии*** от ***С++,*** в ***С#*** ***модификатор*** доступа задается сразу перед типом члена класса. Спецификатор доступа определяет ***тип***разрешенного доступа. Указывать спецификатор доступа не обязательно, но если он отсутствует, то объявляемый

член считается закрытым в пределах класса. Члены с закрытым доступом могут использоваться

только другими членами их класса.

*доступ тип имя\_переменной;*

class Building

{

public int Floors;

public int Area;

public int Occupants;

}

Все переменные объявлены в классе ***Building*** с предваряющим их модификатором доступа ***public***. Как пояснялось выше, благодаря этому они становятся доступными за пределами класса ***Building***.

Для того чтобы создать конкретный объект типа ***Building***, придется воспользоваться

следующим оператором.

Building house = new Building();

Эта строка объявления выполняет три функции. Во-первых, объявляется переменная

***house***, относящаяся к типу класса ***Building***. Сама эта переменная не является

объектом, а лишь переменной, которая может ***ссылаться***на объект. Во-вторых, создается

конкретная, физическая, копия объекта. Это делается с помощью оператора

new. И наконец, переменной ***house*** присваивается ссылка на данный объект. Таким

образом, после выполнения анализируемой строки объявленная переменная ***house***

ссылается на объект типа ***Building***.

Оператор new динамически (т.е. во время выполнения) распределяет память для объекта

и возвращает ссылку на него, которая затем сохраняется в переменной. Следовательно,

в C# для объектов всех классов должна быть динамически распределена память.

Building house; //объявить ссылку на объект

House = new Building(); //распределить память для объекта типа

После выполнения этого оператора объект house станет экземпляром класса ***Building***, т.е. обретет "физическую" реальность.

Для доступа к этим переменным служит оператор доступа к члену класса, который принято называть ***оператором -* *точкой****.* Оператор-точка связывает имя объекта с именем члена класса.

house.Floors = 2;

Пример в glava6\_1.