

Основы объектно-ориентированного программирования (ООП)

Объект и класс

- Основными элементами программы являются не переменные и методы (процедуры), а объекты.
- Объекты – это программные конструкции, включающие набор логически связанных свойств (данных) и методов.
- Объекты создаются на основе шаблонов, которые называются классами и являются экземплярами этих классов.
- Пример описания класса автомобилей:

```
class Автомобиль  
{  
  // описание данных  
  // описание методов  
}
```

Создание экземпляров класса

- При создании экземпляра класса в памяти создается копия этого класса. Созданный таким образом экземпляр класса называют объектом.
- Экземпляр класса можно создать посредством специальной операции new, например, так:

// Объявление переменной типа Автомобиль

Автомобиль myAuto, RedCar;

// переменная это не объект класса!

// Создание экземпляра класса Автомобиль

// и сохранение ссылки на него в переменной myAuto

myAuto = new Автомобиль ();

RedCar = new Автомобиль ();

*Экземпляры класса не зависят друг от друга и являются отдельными программными конструкциями.

Поля, свойства, методы, события

- Объекты состоят из полей, свойств, методов и событий.
- Поля содержат данные объекта
- Свойства – предоставляют управляемый способ доступа к данным объекта
- Методы – определяют действия, которые объект способен выполнять.
- События уведомляют заинтересованных пользователей (другие класса, которые подпишутся на эти события), если в объекте происходит что-то важное.
- Свойства объектов класса Автомобиль: *Цвет, Модель, Расход_топлива*. Методы объектов класса Автомобиль: *Нажать_акселератор, Переключить_передачу, Повернуть_руль*.
- **Методы позволяют реализовать поведение объекта. События представляют собой уведомления о важных происшествиях, например о том, что количество бензина стало ниже заданной величины, или объект класса Двигатель может уведомлять объект класса Автомобиль о событии Перегрев_двигателя.**

Пример описания класса

- `class Автомобиль`
- `{`
- `// описание свойств`
- `public string Модель;`
- `public float Расход_топлива;`
- `private int Число_цилиндров;`
- `// описание методов`
- `public void Повернуть_руль(){...};`
- `private Регулировка_датчика(){...};`
- `// описание события`
- `event Перегрев_двигателя();`
- `}`

Доступ к полям и методам

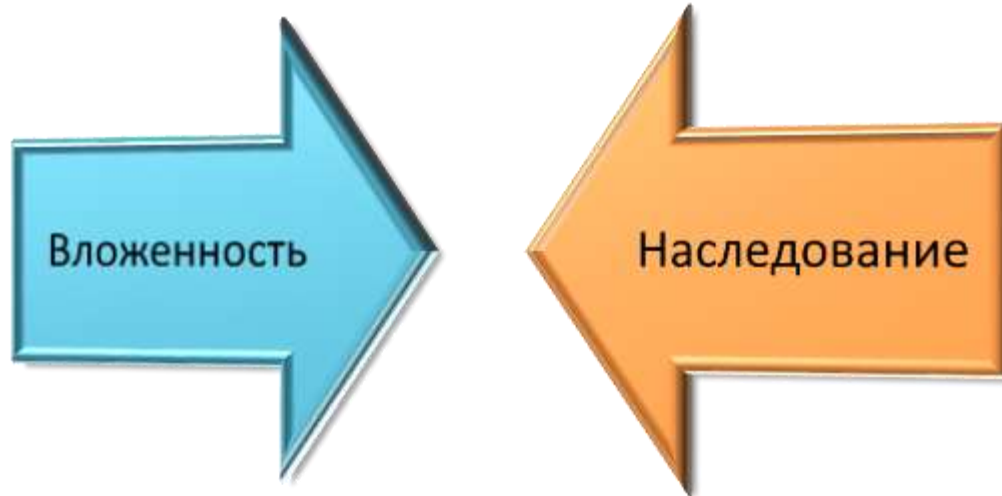
- Для доступа к полям, свойствам, методам элементам объектов используется специальная операция точка (.).

- Например:

`myAuto.Модель = "Reno Logan" // определение Модели`
`myAuto.Повернуть_руль() // поворот руля автомобиля`

- Однако к элементам объекта имеется доступ не из всех частей программы

Отношения между классами



- **Вложенность** – это включение объектов одного класса в объекты другого класса.
- **Наследование** – это описание одного класса на основе другого класса.

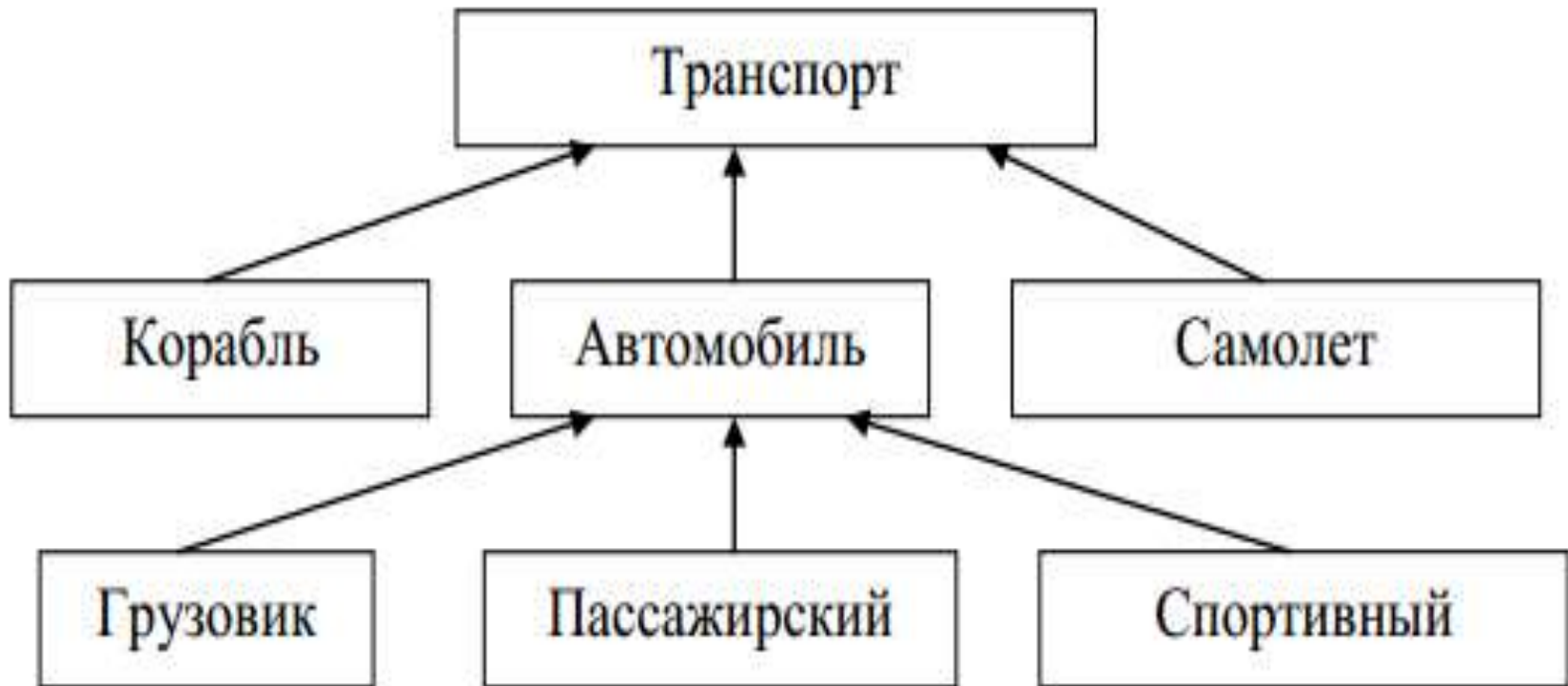
Объектные модели

- Объекты одних классов могут включать объекты других классов в качестве своих полей и предоставлять к ним доступ, как и к другим своим элементам.
- Иерархия вложенности объектов друг в друга называется **объектной моделью** (object model).
- Например, в случае с автомобилем, объект класса *Автомобиль*, который сам по себе является объектом, состоит из ряда вложенных объектов, например, таких как объект класса *Двигатель*, четырех объектов класса *Колесо*, объект класса *Трансмиссия* и т.д.

Наследование

- Один класс может быть описан на основе уже имеющегося описания другого класса. В этом случае между классами задается отношение наследования.
- Наследование позволяет создавать новые классы на основе существующих, при этом новые классы обладают всей функциональностью старых и при необходимости могут модифицировать их.
- Класс, объявленный на основе некоторого (базового) класса, называется производным или классом-потомком.
- У любого класса может быть только один прямой предок – его базовый класс.

Схем наследования классов



Описание класса потомка

- На основе описания класса *Транспортное_средство* (базовый класс) можно описать класс *Автомобиль* (производный класс) :

```
class Автомобиль : Транспорт  
{  
// описание свойств  
public string model;  
public float Расход_топлива;  
private int Число_цилиндров;  
// описание методов  
public void Повернуть_руль(){...};  
private Регулировка_датчика(){...};  
// описание события  
public event Перегрев_двигателя();  
}
```

Основные принципы ООП



Абстрагирование

- Описание объектов реального мира при помощи программных объектов называют абстрагированием.
- Абстрагирование — это способ выделить набор значимых характеристик объекта, исключая из рассмотрения незначимые. Соответственно, абстракция — это набор всех таких характеристик.

Инкапсуляция

- языковой механизм ограничения доступа к определённым компонентам объекта;
- языковая конструкция, способствующая объединению данных с методами (или другими функциями), обрабатывающими эти данные.
- *Смысл инкапсуляции состоит в отделении реализации объекта (его внутреннего содержания) от способа взаимодействия с ним. Другие объекты приложения взаимодействуют с рассматриваемым объектом посредством имеющихся у него открытых (**public**) свойств и методов, которые составляют его интерфейс.*

Полиморфизм

- **Полиморфизм** - возможность объектов с одинаковой спецификацией иметь различную реализацию.
- Полиморфизм позволяет вызывать методы и свойства объекта независимо от их реализации.
- Например, объект класса *Водитель* взаимодействует с объектом класса *Автомобиль* через открытый интерфейс. Если другой объект, например *Грузовик* или *Гоночный_автомобиль*, поддерживает такой открытый интерфейс, то объект класса *Водитель* сможет взаимодействовать и с ними (управлять ими), невзирая на различия в реализации интерфейса.