**ООП**

Объектно-ориентированное программирование - это подход при котором программа строится из объектов.

Объект - это некая сущность которая существует в пространстве и времени. Объекты можно описывать.

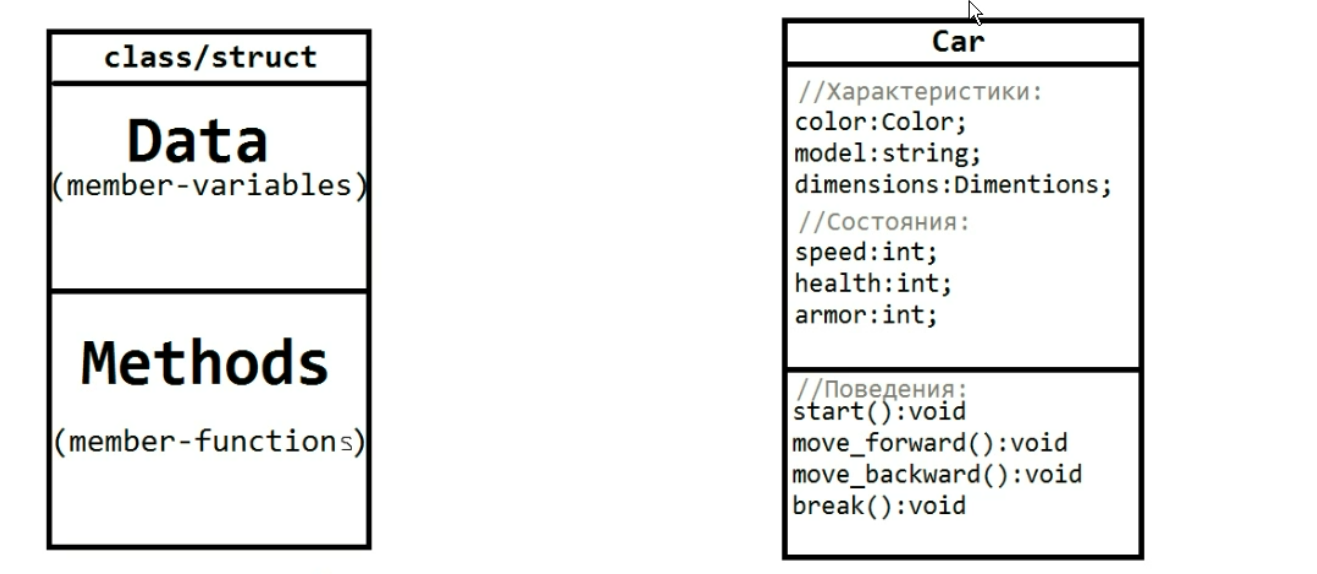
Объекты могут существовать в объективной субъективной или виртуальной реальность.

ООП позволяет проецировать объекты из объективной и субъективной в виртуальную реальность. ООП позволяет описать любую модель мира.

Объекты можно описывать. Каждый объект обладает какими-то характеристиками состояниями и поведением, кроме того каждый объект уникален, даже если он похож на другие.

Объекты принято классифицировать, т.е. разделять на классы.

Класс – это множество объектов обладающих одинаковым набором характеристик состояний и поведений.

С точки зрения языка программирования класс – это синтаксическая конструкция позволяющая описывать объекты. Классы состоят из полей или членов(members). Поля класса могут быть двух типов: переменные члены класса (member – variables) они описывают характеристики и состояния объектов, и методы (member - functions) – функции внутри класса. Метод может быть вызван только для какого-то объекта. Класс всегда можно представить в виде OML- диаграммы. 

Создавая структуру или класс мы создаем новый тип данных. Структуры или классы называют пользовательскими типами данных. Пользовательские типы данных хранят набор значений в отличии от примитивных типов данных, которые хранят одно значение из множества возможных.

**Доступ к полям объектов**

Для доступа к полям объектов используют операторы ‘.’ или ‘->’:

‘.’ – оператор прямого доступа, используется для доступа к полям объекта по имени объекта

‘->’ – оператор прямого доступа, используется для доступа к полям объекта по адресу объекта

**Концепции ООП**

ООП базируется на трёх основных концепциях:

1. Инкапсуляция (Encapsulation) – сокрытие определённой части класса от внешнего мира
2. Наследование (Inheritance)
3. Полиморфизм (Polymorphism)

На данный момент интересует только Инкапсуляция (Encapsulation). Инкапсуляция характеризуется модификаторами доступа и get/set-методами – позволяют получить доступ к закрытым переменным в классе. get-методы – открывают доступ к закрытым переменным на чтение. Обязательно должны быть const. const называется метод, который не изменяет объект для которого вызывается. Set-методы – открывают доступ к закрытым переменным на запись, т.е. позволяют задать значение закрытых переменных. Кроме того set-методы обеспечивают фильтрацию данных.

Для const-объекта могут быть вызваны только const-методы.

this – указатель на объект для которого вызывается метод.

В языке С++ есть три модификатора доступа:

1. private: закрытие поля, доступ можно получить только внутри класса
2. public: открытые поля, доступ можно получить откуда угодно
3. protected: защищенные поля, доступен внутри нашего класса и его дочерних классах (используется только при наследовании классов)

Переменные члены класса обязательно должны быть приватными, их никогда и не при каких обстоятельствах не размещают в public-секции. Переменная в public-секции – грубейшее нарушение инкапсуляции. Сокрытие переменных членов класса защищает их от случайной перезаписи.

В public-секции как правило размещаются методы, хотя они могут быть написаны и private-секции.

Ключевым отличием структуры от класса является то, что у структуры все поля открыты если не указано обратное, а у класса все поля закрыты если не указано обратное.

**Особые методы класса**

В любом классе обязательно должны быть следующие методы

1)Constructor – cоздаёт объект: выделяет память под объект и инициализирует поля

2) ~Destructor – уничтожает объект по завершении его времени жизни.

3)Assigment operator

Конструктор и деструктор никогда не возвраща