

ООП. Лекция *2

1. Объектно-ориентированная модель
2. Отношения между объектами
3. Классы

1. Объектно-ориентированная модель

Модель

- Основа объектно-ориентированной модели = классы и объекты.
- **Объект** моделирует часть окружающей действительности, существует во времени и пространстве.
- В ООП объектом может считаться также объект, который вводится на стадии проектирования.
- **Объект \Leftrightarrow экземпляр класса.**



Состояние

- Состояние зависит от события, которое может произойти.
- Состояние объекта характеризуется перечнем всех свойств объекта и текущими значениями для каждого из ЭТИХ СВОЙСТВ.
- Состояние – это суммарный результат его поведения.
- Состояние – это когда объект хранит информацию.

Поведение

- Когда объект подвергается воздействию.
- И сам воздействует на другие объекты.
- Поведение выражается в терминах состояния объекта и передачи сообщений.
- В ООП действия над объектом – это методы.
- Пять основных видов операций над объектами:
 1. *Модификатор* – изменение состояния объекта путем записи или доступа.
 2. *Селектор* - операция, позволяет определить состояния объекта ничего не меняя (операция чтения).
 3. *Итератор* – доступ к содержанию объекта по частям (в определенной последовательности).
 4. *Конструктор* - операция создания (и/или инициализация объекта).
 5. *Деструктор* - операция разрушения объекта (и/или) освобождение занимаемой им памяти.

```
public MyClass(string s, byte b)
{
    Name = s;
    Age = b;
}
```

```
~MyClass()
{
    Console.WriteLine("Объект уничтожен");
}
```



Создаются автоматически.

Операции

- **Операция** – это услуга, которую класс может предоставить своим экземплярам.
- Конструктор и деструктор – универсальные операции.
- В объектно-ориентированных языках операциями могут быть только методы класса, т.к. нельзя операцию определить вне класса.
- В языках C++ и т.д. допускается описать операцию как независимую от объектов. Операции, не являющиеся членами класса – это свободные подпрограммы, которые выполняют роль операции высокого уровня над объектом одного или разных классов. Могут группироваться в соответствие с классами, с которыми они созданы. В этом случае их называют утилитами.
- Все методы – это операции, но не все операции – это методы.
- Все методы и свободные подпрограммы, которые относятся к одному и тому же классу, образуют протокол. Протокол определяет поведение объекта, охватывая его статические и динамические аспекты.

Идентичность

- Это свойство объекта, которое отличает его от других объектов.
- Возможны две ситуации:
 1. Два различных объекта могут иметь абсолютно одинаковые поля.
 2. Напротив, поля данного объекта могут изменяться в процессе выполнения системы, но это не влияет на идентификацию объекта.

Активный / пассивный объект

Активный

Имеет свой поток управления.

Объект автономен, т.е. может проявлять свое поведение без воздействия со стороны другого объекта – самостоятельно поведение.

Пассивный

Не имеет своего потока управления.

Объект изменяет состояние только под воздействием другого объекта – поведение по заказу.

2. Отношения между объектами

Отношения между объектами

- Отношения между двумя объектами основываются на взаимной информации о разрешенных операциях и ожидаемом поведении.

Особо интересны 2 типа отношений:

- 1) **Связи** (равноправные отношения)
- 2) **Агрегация** (отношения целого и части)

1. СВЯЗЬ

- **Связь** – это физическое или понятийное **равноправное** соединение между объектами.
- Объект сотрудничает с другими объектами через соединяющие их связи.
- Связь между объектами и передача сообщений обычно односторонняя, но технически она может быть взаимной.

Участник связи играет 1 из 3-х ролей:

Актер – объект воздействует на другие объекты, но сам никогда не подвергается воздействию других объектов (деятельный объект).

Сервер – объект, который никогда не воздействует на другие объекты, он только используется другими объектами (пассивный объект)

Агент – объект, который может быть как пассивным, так и активным. Он создается для выполнения работы от имени актера или другого агента.

2. Агрегация

- Отношения целого и части => иерархия объектов.
- **Агрегат** – это целое, **атрибут** – это часть.
- Объект, являющийся атрибутом другого объекта-агрегата, имеет связь со своим агрегатом. Через эту связь может посылать агрегату сообщения.
- Агрегация – частный случай ассоциации.
- Может означать физическое вхождение одного объекта в другое .

3.Классы

Класс

- Понятие **класс** и **объект** тесно связаны, но существует важное различие.
- В ООП **класс** – это некоторое множество объектов, имеющих общую структуру и общее поведение.
- **Объект** – это экземпляр класса.
- **Объект не является классом, НО класс может быть объектом!**

Типы отношения между классами

1. От частного к общему

2. От целого к части

3. Ассоциация

