



Занятие №2



# Привет



Многие, как оказалось, хотят послушать базовое введение в ООП, так что вот оно

- Объектно-ориентированное программирование
  - Это значит что всё мы пытаемся представить в виде **ОБЪЕКТОВ**
  - Сама программа строится как последовательность взаимодействий объектов

# Что такое “объект”

---



- Объект — сущность, которой можно посылать сообщения (или вызывать методы), и которая может на них реагировать в соответствии со своим состоянием
- Объект — сущность класса
- Данные (состояние) объекта скрыто от остального кода (см. Инкапсуляция)

# Что такое “класс”?

---



- Класс задаёт внутреннюю структуру значений и набор операций, которые над этими значениями могут быть выполнены
- Таким образом класс это абстрактный тип данных, который описывает свойства категории объектов

# Основные понятия ООП

---



- Наследование
- Инкапсуляция
- Полиморфизм

# Наследование

---



Отношение наследования между классами означает фактически включение в *наследника* всех свойств *родителя (или базового класса)*

Наследование нужно для расширения функциональности классов

Инкапсуляция — объединение данных и методов для их работы в объекте. Иногда в это понятие включается сокрытие этих данных и реализации методов.

Если “на пальцах”, то инкапсуляция это то, что позволяет говорить объекту что делать без уточнения “как именно”.



# Полиморфизм

---



Полиморфизм — возможность в наследниках заменять реализацию методов родителя так, чтобы внешняя система не видела различий между их вызовами

# That's all, folks!



---

Собственно это вся теория на эту тему. Давайте посмотрим на некоторые детали

# Конструкторы

---



Конструктор — метод, который создаёт объект класса и предварительно заполняет его данными

# Поле, метод

---



- Поле — данные, ассоциированные с объектом
- Метод — функция, или просто код, ассоциированные с объектом

- Интерфейс, он же протокол
- Протокол — описание (контракт), по которому система может взаимодействовать с объектом
- Разные классы могут реализовывать один и тот же протокол.

# Конструкции swift

---



- Теперь кратко поговорим про некоторые конструкции swift, которые, как правило, вызывают больше всего вопросов

- Пишется как название типа с “?” на конце.
- Означает, что значение может быть или данного типа, или пустым (nil)
- Нужно для того, чтобы компилятор мог быть уверенным в том, какое где значение и не было ошибок в рантайме
- Для использования optional его надо развернуть, это делается написанием ? после имени переменной
- Если переменная nil, то ничего после ? Не выполнится

# Explicitly/Force unwrapped optional

---



- Это значит, что там всё же может быть nil, но компилятор этого проверять не будет
- Их можно проверять через ?, но не обязательно



# struct/class

---



- Структуры передаются по значению
- Классы передаются по ссылке
- Структуры по умолчанию immutable

# Extension

---



- Способ добавить методы к уже существующему классу

# Protocol Extension

---



- Добавляет реализацию “по умолчанию” для части протокола

# guard



- Это if с перевернутым условием
- Используется просто для того, чтобы сематически было понятно, что это проверка условий начальных