Основы Swift

Значимые и ссылочные типы

Значимые и ссылочные типы в Swift

- Классы — ссылочный тип
- Структуры и перечисления — значимые типы

Перечисления и структуры

- Перечисления (Enumeration) и структуры (Structures) — значимые типы (value types).

- Int, Float, String, Bool, Array, Dictionary — являются значимыми типами, к тому же они реализованы с помощью структур.

Значимые типы

```
var firstNumber = 5
var secondNumber = firstNumber

firstNumber = 10

print(firstNumber, secondNumber)
// firstNumber = 10, secondNumber = 5

let constantNumber = 5
constantNumber = 10 // Error
```

Классы — ссылочный тип

При присвоении значения не копируются,
а передается ссылка.
Объект объявленный как константа
позволяет менять свои поля.

```
class DecimalNumber {
    var value = 0
}

var numberOne = DecimalNumber()
numberOne.value = 5
var numberTwo = DecimalNumber()
numberTwo.value = 3
```

```
numberTwo = numberOne
numberOne.value = 7

print(numberOne.value,
numberTwo.value)
// numberOne = 7 numberTwo = 7
```

Операторы идентичности

- Идентичны (===)
- Не идентичны (!==)

```
if numberOne === numberTwo {
    print("tenEighty and alsoTenEighty refer
    to the same Decimal Number")
}
// "tenEighty and alsoTenEighty refer
to the same Decimal Number
```

- Идентичны — два ссылочных значения указывают на один и тот же экземпляр
- Равны — экземпляры могут иметь одинаковые значения, но указывают на разные экземпляры (определяется реализацией)

Операторы сравнения

Созданные вами классы и структуры не получают реализацию оператора эквивалентности по-умолчанию.

```
extension DecimalNumber {
    static func == (left: DecimalNumber,
    right: DecimalNumber) -> Bool {
        return (left.value == right.value)
    }

    static func != (left: DecimalNumber,
    right: DecimalNumber) -> Bool {
        return !(left == right)
    }
}
```

Классы или структуры?

- Структуры используем для строк, URL, дат, изображений.
- Не требуется мутабельность из разных блоков программы.
- Классы: для объектов с жизненным циклом, для реализации наследования.