

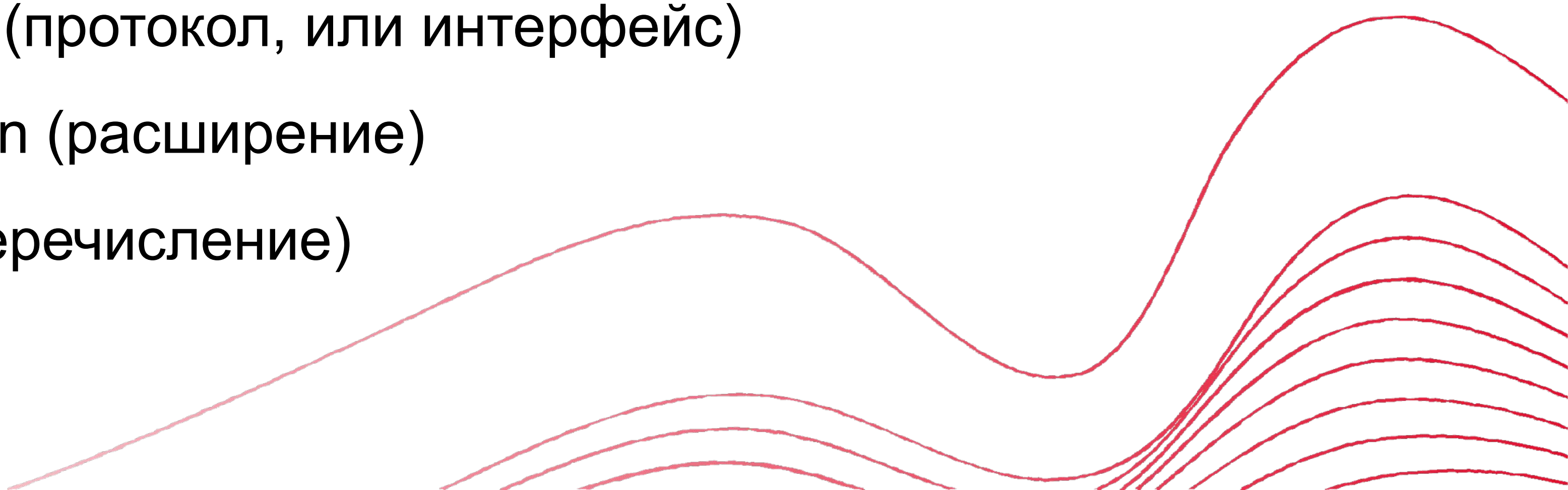


# Лекция 4





# Agenda

- Повторение классов и структур. Простая задача.
  - Управление памятью MRC / ARC
  - Используем оператор switch и guard
  - protocol (протокол, или интерфейс)
  - extension (расширение)
  - enum(перечисление)
- 



# Классы - практика

а) Создайте класс Матрос (sailor), в котором будет один метод `func sayHello() { }`. Этот метод должен выводить в консоль приветствие. И класс Корабль, в котором будут свойства: название, массив "матросы".

б) Создайте несколько экземпляров класса Матрос, и с их помощью экземпляр класса Корабль. С использованием цикла `for` выведите в консоль приветствие от имени всех матросов.

а) Создайте класс Игрок, в котором будет один метод `func playWell() { }`. Этот метод должен выводить в консоль "Играю как профи!". И класс Команда, в котором будут свойства: название, массив "игроки".

б) Создайте несколько экземпляров класса Игрок, и с их помощью экземпляр класса Команда. С использованием цикла `for` выведите в консоль реплику "Играю как профи!" от имени всех игроков.





# ARC и управление памятью

- Область видимости переменной,
- Цикл ссылок на примере User и Phones
- Weak ссылка (всегда optional)
- Циклы ссылок в замыканиях

Подробная статья на тему ARC и управления памятью:

<https://habr.com/ru/post/451130/>

Про ARC на английском:

<https://docs.swift.org/swift-book/LanguageGuide/AutomaticReferenceCounting.html>

Про ARC на русском:

<https://swiftbook.ru/content/languageguide/automatic-reference-counting>



# Protocol

- Как реализовать простой протокол
- Объединим несколько элементов в 1 массиве. Фермер, корова и трава.
- Создаем цикл, выводящий названия объектов с условием.
- Решаем ту же задачу с использованием протоколов.

Канал на youtube, где толково изложена информация. Но важное условие - смотреть нужно по порядку и все:

<https://www.youtube.com/c/AlexSkutarenko/videos>

Документация по протоколам:

<https://swiftbook.ru/content/languageguide/protocols/>



# Protocol - практика

- а) Создать несколько структур (и/или классов) обозначающих продукты. Создать протокол, на который будут подписаны продукты. Протокол реализует свойство name для продукта и метод taste(), который выводит в консоль информацию - вкусный продукт или нет.
- б) Добавить все продукты в массив и с помощью цикла for вывести в консоль название продукта и вызвать метод taste.

Статья с более глубоким разбором протоколов:

<https://habr.com/ru/company/acronis/blog/420239/>

POP (protocol oriented programming) - тема со звездочкой. Я бы советовал ее для самостоятельного изучения, если вы хорошо разобрались во всех понятиях из последних 4х уроков. Если чувствуете, что это слишком тяжело - это не страшно, на позицию junior developer обычно это не требуется.

<https://habr.com/ru/post/358804/>

<http://bytepace.com/ru/blog/pop>



# Extension (расширение)

- Простой пример расширения
- Расширяем структуру Int и добавляем проверку isEven
- Добавляем функцию возведения в степень powTo
- Расширения протоколов

Документация по extension на русском:

<https://swiftbook.ru/content/languageguide/extensions/>



## Extension (расширение) - практика:

- a) Расширить тип Integer свойством isNegative и isPositive возвращающее true или false. Если Int равен нулю, возвращаем false в обоих методах
- b) Расширить тип String, методом addGreetings(greetings: String) { } Метод должен возвращать стринг с префиксом greetings. Например для строки "Как дела?" мы добавляем префикс "Привет!" и получаем строку "Привет! Как дела?". Не забудьте про пробел.
- c) Создать протокол Greetings, который требует реализации метода func sayHello(). Расширить протокол Greetings, добавив ему реализацию по умолчанию, которая выполняет print("Hello!").





# Оператор switch и guard

- Тестируем оператор switch на переменной age, проверяем диапазоны.
- Добавляем оператор fallthrough
- Добавляем в switch дополнительно условие ... where ... :

Подробнее про конструкцию switch:

<https://metanit.com/swift/tutorial/2.7.php>

Подробнее про guard и defer (про defer не плохо знать, но я не видел использования этого оператора на реальных проектах):

<https://oxozle.com/2016/01/10/nachalo-swift-guard-i-defer/>



# Enum (перечисление)

- Создаем простой Enum “Члены семьи”
- Добавляем ассоциативные значения, (age: Int ... )
- Создаем switch с перебором enum-а. Добавляем условия where вместо if для ассоциативных значений.
- Добавляем Enum Gender и используем его для детей внутри первого enum-а. Новый enum можно объявить внутри старого.
- “Подписываем” enum под примитивный тип. Создание с rawValue и вывод в консоль rowValue.



# Enum (перечисление) - практика

- а) Создать Enum “шахматные фигуры” в котором будут case-с: тура, конь, офицер, ладья, король и пешка.
- б) Для каждого case добавить ассоциативный тип “цвет”.
- в) Подписать шахматные фигуры под String и добавить rawValues - названия фигур. Создать array из элементов enum-а. Создать функцию, которая принимает array фигур и выводит в консоль имя фигуры и ее цвет.



## Термины:

**Protocol** (протокол) - набор свойств и методов без реализации. Тот класс/структура/энам, который подписывается на протокол, обязуется реализовать его свойства и методы.

**Enum** (перечисление) - определяем набор перечисленных значений, которые нельзя добавить, или удалить.

**Switch** - Конструкция работающая так же как цепочка операторов if else ... с некоторым дополнительным функционалом.

**Guard** - такой же условный оператор как и if, позволяющий завершить функцию при невыполнении главного условия.

**POP** (protocol oriented programming) - программирование основанное на протоколах и всех преимуществах языка swift, обычно противопоставляется OOP.





# Все ссылки:

Подробная статья на тему ARC и управления памятью:

<https://habr.com/ru/post/451130/>

Про ARC на английском:

<https://docs.swift.org/swift-book/LanguageGuide/AutomaticReferenceCounting.html>

Про ARC на русском:

<https://swiftbook.ru/content/languageguide/automatic-reference-counting>

Канал на youtube, где толково изложена информация. Но важное условие - смотреть нужно по порядку и все:

<https://www.youtube.com/c/AlexSkutarensko/videos>

Документация по протоколам:

<https://swiftbook.ru/content/languageguide/protocols/>

Статья с более глубоким разбором протоколов:

<https://habr.com/ru/company/acronis/blog/420239/>

POP (protocol oriented programming) - тема со звездочкой. Я бы советовал ее для самостоятельного изучения, если вы хорошо разобрались во всех понятиях из последних 4х уроков. Если чувствуете, что это слишком тяжело - это не страшно, на позицию junior developer обычно это не требуется.

<https://habr.com/ru/post/358804/>

<http://bytepace.com/ru/blog/pop>

Документация по extension на русском:

<https://swiftbook.ru/content/languageguide/extensions/>

Подробнее про конструкцию switch:

<https://metanit.com/swift/tutorial/2.7.php>

Подробнее про guard и defer (про defer не плохо знать, но я не видел использования этого оператора на реальных проектах):

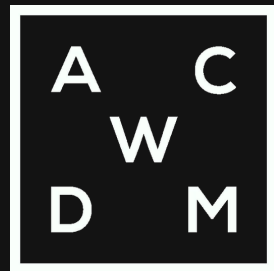
<https://oxozle.com/2016/01/10/nachalo-swift-guard-i-defer/>

Подробнее про конструкцию switch:

<https://metanit.com/swift/tutorial/2.7.php>

Подробнее про guard и defer (про defer не плохо знать, но я не видел использования этого оператора на реальных проектах):

<https://oxozle.com/2016/01/10/nachalo-swift-guard-i-defer/>



Web  
Academy

# CREATE YOUR IT FUTURE.

WITH WEB ACADEMY

