Цели домашнего задания

Научиться использовать:

* протоколы,
* расширения (extension).

Что нужно сделать

Предлагаем закрепить навыки работы с расширениями и протоколами в Swift. Вас ждёт несколько заданий, в которых нужно будет реализовать протоколы и работать с ними.

Для решения заданий создайте новый файл Playground, выполните в нём все задания и отправьте на проверку.

Задание 1

Объявите три сущности, которые реализует предоставленный вам протокол. После этого для всех сущностей вызовите все методы и свойства, объявленные в протоколе.

**Протокол:**

protocol CalorieCountProtocol {  
     var calories: Int { get }  
     func description() -> String  
}

Должны выполняться следующие условия:

1. Объявлены класс, структура и перечисления enum, которые реализует данный протокол.
2. Реализация протокола может быть любой.
3. Вызваны все свойства и методы протокола у каждого экземпляра из пункта 1.
4. Задание сделано без изменения протокола.

Задание 2

Поработайте с протоколом Equatable, который нужен для сравнения сущностей.

Вам предоставлены структура и класс, которые нужно сравнить. Добавьте в объявление этих сущностей протокол Equatable. При необходимости реализуйте недостающий метод.

**Структура:**

enum BalanceType {  
    case positive, negative, neutral  
}  
  
struct Balance {  
    let type: BalanceType  
    let amount: Int  
}

**Класс:**

class BalanceObject {  
    var amount: Int = 0  
}

Должны выполняться следующие условия:

1. Структура и класс реализуют протокол Equatable.
2. Проведена проверка, что протокол реализуется. Для этого сравните экземпляры этих сущностей с помощью оператора ‘==’.

Задание 3

Вам дан протокол и три сущности, которые его реализуют. Добавьте в протокол новый метод. Реализуйте этот метод с помощью расширения протокола. Создайте массив из этих сущностей и вызовите в цикле новый метод у каждого элемента массива.

**Код:**

protocol Dog {  
    var name: String { get set }  
    var color: String { get set }  
}  
  
struct Haski: Dog {  
    var name: String  
    var color: String  
}  
  
class Corgi: Dog {  
    var name: String  
    var color: String  
  
    init(name: String, color: String) {  
        self.name = name  
        self.color = color  
    }  
}  
  
struct Hound: Dog {  
    var name: String  
    var color: String  
}

Должны выполняться следующие условия:

1. В протокол добавлен метод func speak() -> String.
2. Реализация метода из пункта 1 сделана с помощью расширения.
3. Создан массив, содержащий как минимум один экземпляр каждой сущности.
4. У каждого элемента массива вызван новый метод.

Советы и рекомендации

Для выполнения заданий ознакомьтесь с официальной документацией [языка Swift](https://swift.org/).

Что оценивается

* Все задания выполнены в соответствии с описанными условиями.
* Учтены рекомендации и подсказки.
* Код работает корректно и возвращает описанный в задании результат.
* Модификаторы доступа private должны быть там, где нужно скрыть внутреннюю реализацию.

Как отправить задание на проверку

Пришлите файл Playground с выполненными заданиями через форму ниже.