

Изучение Dart

№ урока 3 - месяц 2.

Тема: Миксины. Статические члены классов.

Средства обучения: Компьютер с установленной Visual Studio Code, Android Studio, Flutter SDK, JDK.

Обзор, цель и назначение урока:

Познакомить студентов с понятием миксин.

Объяснить ключевое слово `mixin`.

Дать общее понятие статического члена класса

Привести пример с использованием ключевого слова `static`.

Базово рассказать `Generic`.

Изучив материал данного занятия, учащийся сможет:

- Создавать зависимые, несамостоятельные классы и пользоваться ими на практике.
- Пользоваться ключевыми словами:
 - a. `with`
 - b. `mixin`
 - c. `static`
- Применять на практике статические члены классов.
- Создавать обобщенные типы данных.

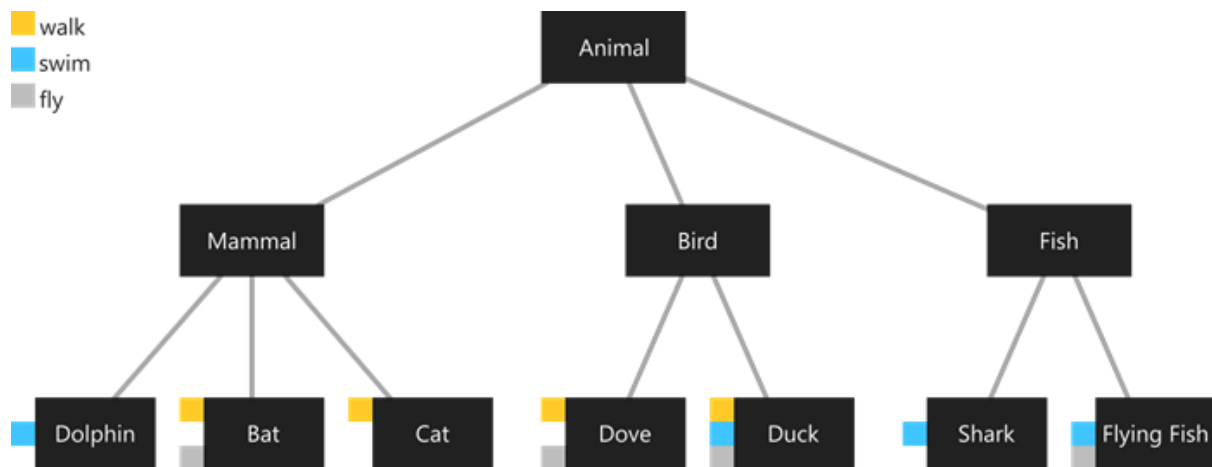
Содержание урока:

- Что такое миксины в целом?
- Где они(миксины) используется, и как правильно применять их

- Отличие наследования миксинов и ключевое слово **with**
- Ключевое слово **mixin**
- Понятие о статических членах класса
- Ключевое слово **static**
- Generic и где он применяется

Самостоятельная деятельность учащегося(ДЗ):

Задание 1:



У нас есть суперкласс под названием Animal, который имеет три подкласса (Mammal, Bird и Fish). Внизу у нас есть конкретные классы. Маленькие квадраты представляют поведение. Например, синий квадрат указывает, что экземпляр класса с таким поведением может плавать.

Задание 2.

Задание с водой. Практика понимая Тех задания.

Представьте, что у нас есть два класса существ, каждый из которых обладает собственным набором поведения. Самый распространенный вариант, который у нас есть, - это расширения, в которых мы можем взять свойства и методы одного класса и сделать их доступными в другом. Поскольку мы не будем использовать Reptile в отдельном экземпляре, мы можем установить его как абстрактный класс, чтобы его нельзя было инициализировать, а можно было просто расширить. Это хорошо для нашего текущего примера, но по мере того, как мы добавляем больше животных, быстро становилось очевидным, что многие из этих методов не только для рептилий. Если бы мы хотели создать класс Fish с методом плавания, а не просто расширять Reptile, что очень ограничивало бы нас, когда нам нужно больше функциональности от других классов, поскольку мы можем использовать только расширения по одному для каждого класса. Мы можем использовать миксины, чтобы разбить наше более универсальное поведение на более мелкие, более повторно используемые компоненты, которые мы можем добавить к любому классу, в котором они нам нужны.