**Урок 3. Миксины.**

**Статические члены**

**классов.**

**План урока:**

1. **Опрос по материалам предыдущего урока и проверка д.з.**
2. **Миксины, их особенности и ключевое слово with**
3. **Ключевое слово mixin**
4. **Общее понятие о статических членах класса**
5. **Ключевое слово static**
6. **Упомянуть про Generic**

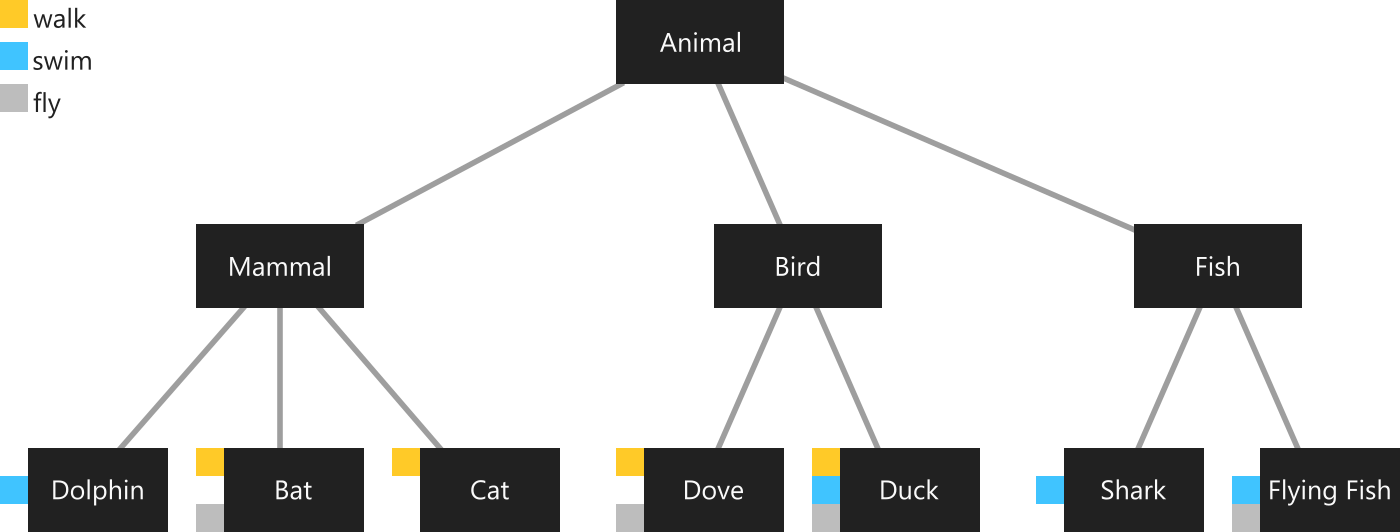
**Начало урока:**

1. Опросить студентов по материалу предыдущего урока:
   1. Что такое наследование и как оно реализуется?
   2. Что такое переопределение классов? Приведите примеры
   3. Для чего нужны абстрактные классы?
   4. В чем разница между интерфейсом и классом?
2. Рассказать студентам что такое миксины в целом. Привести примеры.
3. Объяснить студентам разницу наследования и миксинов.
4. Ключевое слово **mixin** и привести пример реализации миксина.
5. Пояснить за статические поля и методы. Привести примеры.
6. Ключевое слово **static** и его применение.
7. Рассказать про обобщенные типы данных.
8. Ключевое слово **Generic** и его применение.

**Домашнее задание:**

**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.