

# Классы и интерфейсы

**№ урока:** 2    **Курс:** TypeScript Fundamentals

**Средства обучения:** Visual Studio, Visual Studio Code, NotePad++

## Обзор, цель и назначение урока

Цель урока – ознакомление студентов с понятием класса и его конструктором. Ознакомить учащихся с модификаторами доступа, парой **get/set**. Изучение понятия наследования и использования абстрактных классов. Знакомство с интерфейсами и их применением для функций и классов.

## Изучив материал данного занятия, учащийся сможет:

- Использовать классы вместо функций конструкторов;
- Понимать определение конструктора в классе;
- Использовать аксессоры и модификаторы доступа;
- Применять на практике наследование;
- Использовать интерфейсы.

## Содержание урока

1. Классы в **TypeScript**;
2. Конструкторы;
3. Модификаторы доступа;
4. **Accessors**;
5. Наследование;
6. Абстрактные классы;
7. Использование интерфейсов.

## Резюме

- **Класс** – это конструкция языка, состоящая из ключевого слова **class**, идентификатора (имени) и тела. Класс может содержать в своем теле: поля, методы, свойства и конструкторы.
- **Поля** определяют состояние будущего объекта.
- **Методы** определяют поведение будущего объекта.
- Кроме обычных функций классы имеют специальные функции - конструкторы, которые определяются с помощью ключевого слова **constructor**. Конструкторы выполняют начальную инициализацию объекта.
- Кроме обычных свойств и функций класс может иметь статические. Для использования статических функций и свойств не надо создавать объект класса. Статические функции и свойства определяются с помощью ключевого слова **static**.
- **public, private, protected** – ключевые слова, модификаторы доступа. С их помощью определяется видимость членов класса.
- **public** – видимый для всех (в классе и за пределами класса).
- **private** – видимый только в пределах класса (за пределами класса доступ отсутствует).
- **protected** – видимый в пределах класса и в классах-наследниках (за пределами класса и классов-наследников доступ отсутствует).
- Все члены классов без модификатора доступа по умолчанию используют модификатор доступа **public**.
- Наследование - механизм создания класса посредством расширения уже существующего класса.
- **extends** – ключевое слово, которое определяет, какой класс будет базовым (родительским) для текущего. Класс наследник получает от родителя свойства и методы.

- **Абстрактный класс** - это класс, который может выступать только в роли базового класса. Создать экземпляр абстрактного класса не получится. Абстрактный метод - это метод, который не имеет реализации в текущем классе, но обязательно должен быть реализован в производном классе. Абстрактные методы могут создаваться только в абстрактных классах.
- Интерфейс определяет свойства и методы, которые объект должен реализовать. **Интерфейс** - это определение абстрактного типа данных, но без реализации. Интерфейсы определяются с помощью ключевого слова **interface**.

### Закрепление материала

- Как в **TypeScript** реализовано наследование?
- Что такое абстрактный класс? Абстрактный метод?
- Для чего используется ключевое слово **implements**?
- В чем разница между **private** и **protected**?
- Для чего используется ключевое слово **static**? **Readonly**?
- Что такое опциональные параметры в интерфейсе?
- Для чего используется конструктор в классах?

### Дополнительное задание

Создайте интерфейс, описывающий поведение животного (свойства, методы передвижения). Примените данный интерфейс к классу **Cat**, **Bird**, **Fish**. Подумайте, какие свойства должны быть опциональными.

### Самостоятельная деятельность учащегося

#### Задание 1

Выучить основные понятия, рассмотренные на уроке.

#### Задание 2

Создать понятие абстрактного родительского класса **Car**. От него создать 3 производных класса (марки автомобилей) с применением метода **super()**. В классах использовать модификаторы как в родительском классе, так и в производных. Создать от производных классов минимум по 2 экземпляра (модели автомобилей). Методы в производных классах должны выводить на экран все свойства (описание автомобиля). Подумайте, какие свойства в производных классах должны быть **public**, какие – **private** и **protected**.

### Рекомендуемые ресурсы

<https://www.typescriptlang.org/>

<https://www.typescriptlang.org/play/index.html>

<https://github.com/Microsoft/TypeScript>

[https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5\\_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)