



Microsoft Partner
Silver Learning



TypeScript Fundamentals

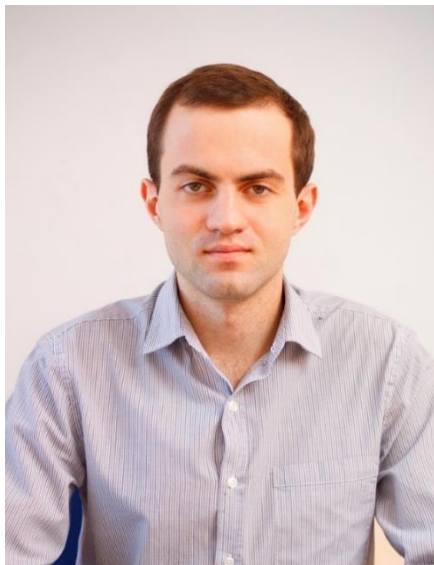
Классы и интерфейсы



ITVVDN
IT VIDEO DEVELOPERS NETWORK

TypeScript Fundamentals

Introduction



Охрименко Дмитрий
MCT

 _okhrimenko

 dmitriy.okhrimenko

 <https://goo.gl/eeTdMv>



MCID: 9210561

TypeScript Fundamentals

После урока обязательно



Повторите этот урок в видео формате на [ITVDN.com](http://itvdn.com)



Проверьте как Вы усвоили данный материал на TestProvider.com

TypeScript Fundamentals

Тема урока

Классы и интерфейсы



TypeScript Fundamentals

План урока

1. Классы в TypeScript

- Конструкторы
- Модификаторы доступа
- Accessors

2. Наследование

- base
- Абстрактные классы

3. Использование интерфейсов



TypeScript Fundamentals

Классы

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

```
class User {  
    firstName: string; // поле  
    lastName: string; // поле  
  
    print() : void { // метод  
        console.log(this.firstName + " " + this.lastName);  
    }  
}
```



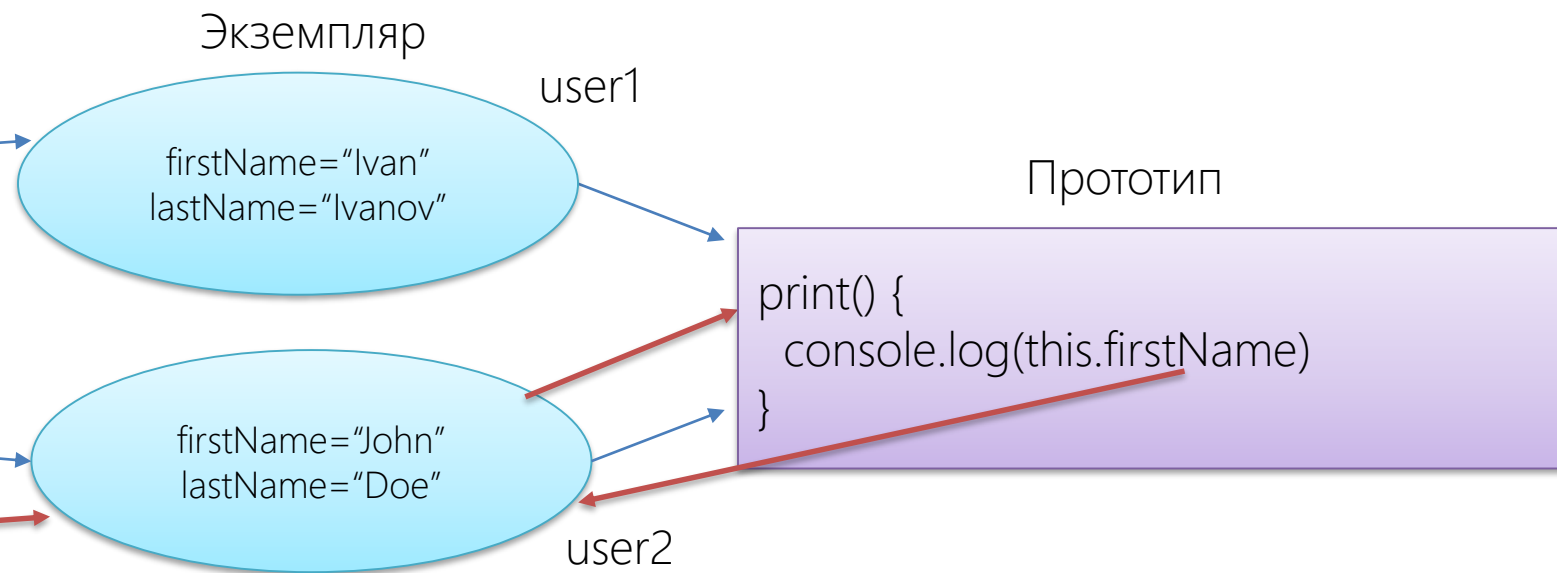
TypeScript Fundamentals

Экземпляры и прототипы

```
class User {  
  firstName: string; // СВОЙСТВО  
  lastName: string;  // СВОЙСТВО  
  
  print() : void {    // МЕТОД  
    console.log(this.firstName + " " + this.lastName);  
  }  
}
```

```
let user1: User = new User();  
user1.firstName = "Ivan";  
user1.lastName = "Ivanov";
```

```
let user2: User = new User();  
user2.firstName = "John";  
user2.lastName = "Doe";  
user2.print();
```



TypeScript Fundamentals

Конструктор

Конструктор – специальный метод, который используется для инициализации экземпляра.

В теле класса конструктор создается с помощью ключевого слова **constructor**.

При создании экземпляра класса для вызова конструктора необходимо использовать ключевое слово **new**.

```
constructor() {  
    this.value = "Hello world";  
}
```



TypeScript Fundamentals

Модификаторы доступа

`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`**



TypeScript Fundamentals

Наследование

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



TypeScript Fundamentals

Абстрактные классы

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.

```
abstract class Animal { // абстрактный класс
  constructor(public name: string) { }

  abstract makeSound(); // абстрактный метод

  public move(): void {
    console.log(this.name + " передвигается")
  }
}
```



TypeScript Fundamentals

Интерфейсы

Интерфейс — специальный тип данных определяющий контракт, которого должен придерживаться определенный объект.

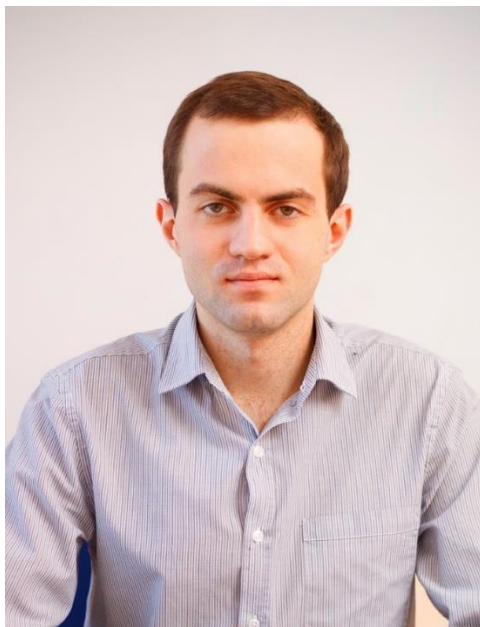
```
interface Report {  
  name: string;  
  build: () => string;  
}
```

```
class DailyReport implements Report {  
  name: string = "Daily Report";  
  
  build() : string {  
    return "some daili report data";  
  }  
}
```



TypeScript Fundamentals

Спасибо за внимание! До новых встреч!



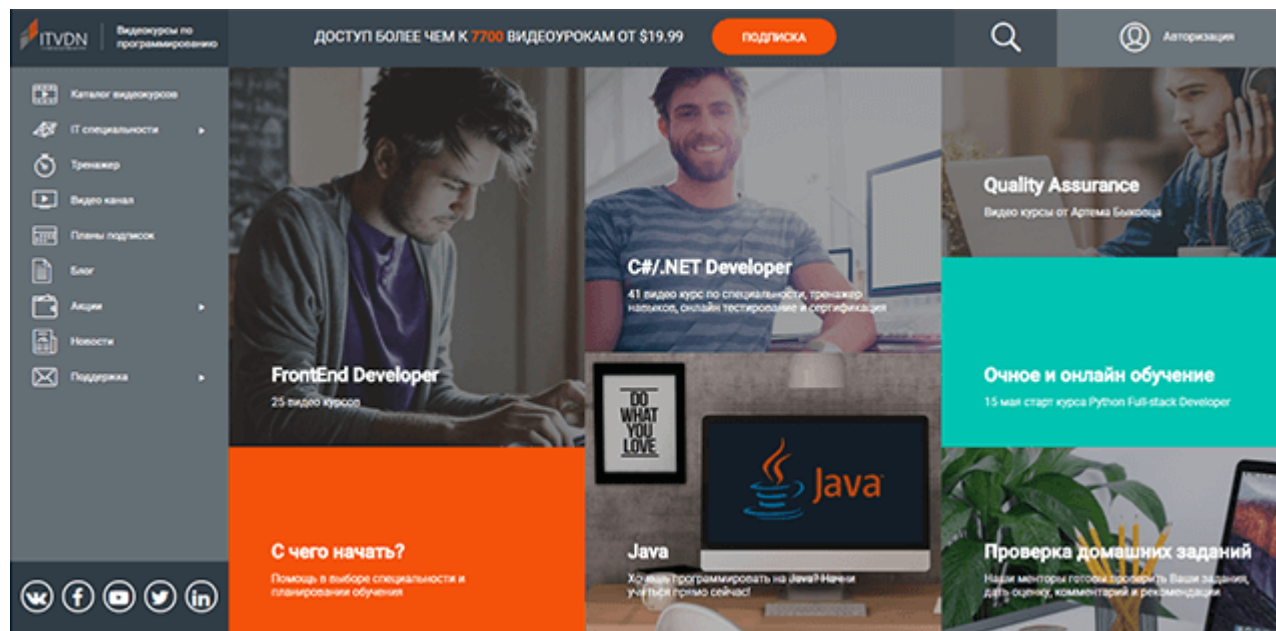
Охрименко Дмитрий
МСТ



MCID: 9210561

Смотрите наши уроки в видео формате

ITVDN.com



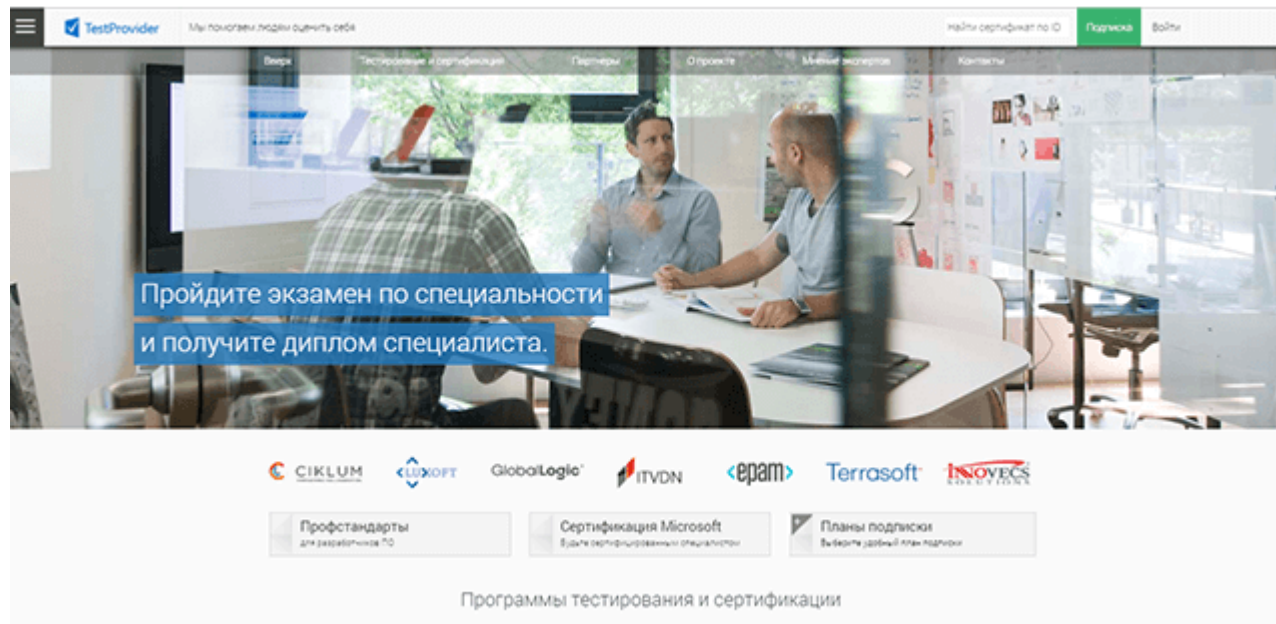
Посмотрите этот урок в видео формате на образовательном портале ITVDN.com для закрепления пройденного материала.

Курсы записаны сертифицированными тренерами, которые работают в учебном центре CyberBionic Systematics и другими высококвалифицированными разработчиками.



Проверка знаний

TestProvider.com



TestProvider – это online сервис проверки знаний по информационным технологиям. С его помощью Вы можете оценить Ваш уровень и выявить слабые места. Он будет полезен как в процессе изучения технологии, так и для общей оценки знаний IT специалиста.

После каждого урока проходите тестирование для проверки знаний на [TestProvider.com](https://testprovider.com)

Успешное прохождение финального тестирования позволит Вам получить соответствующий Сертификат.



TypeScript Fundamentals

Q&A

Информационный видеосервис для разработчиков программного обеспечения

