Microsoft Partner
Silver Learning

TypeScript

Класи та інтерфейси

ITVDN
IT VIDEO DEVELOPERS NETWORK

Introduction

Ігор Михайличенко
Frontend Engineer

iMykhailychenko

Ihor Mykhailychenko

Тема уроку

Класи та інтерфейси

План уроку

1. Класи в TypeScript
    - Конструктори
    - Модифікатори доступу
    - Accessors
2. Наслідування
    - base
    - Абстрактні класи
3. Використання інтерфейсів

Класи

Клас – це конструкція мови, що складається з ключового слова class, ідентифікатора (імені) та тіла.

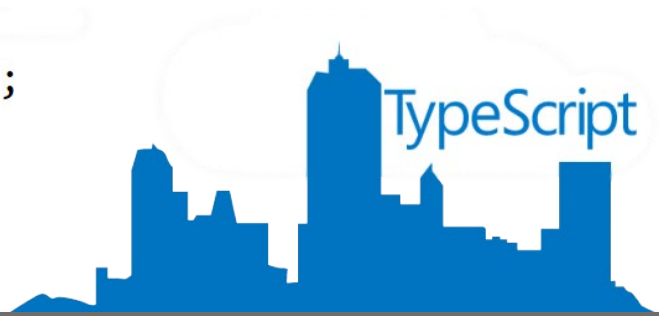
Клас може містити у своєму тілі: *поля, методи, властивості та конструктори*.

Поля – визначають стан майбутнього об'єкту.

Методи– визначають поведінку майбутнього об'єкту.

```
class User {
    firstName: string; // поле
    lastName: string;  // поле

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



Примірники та прототипи

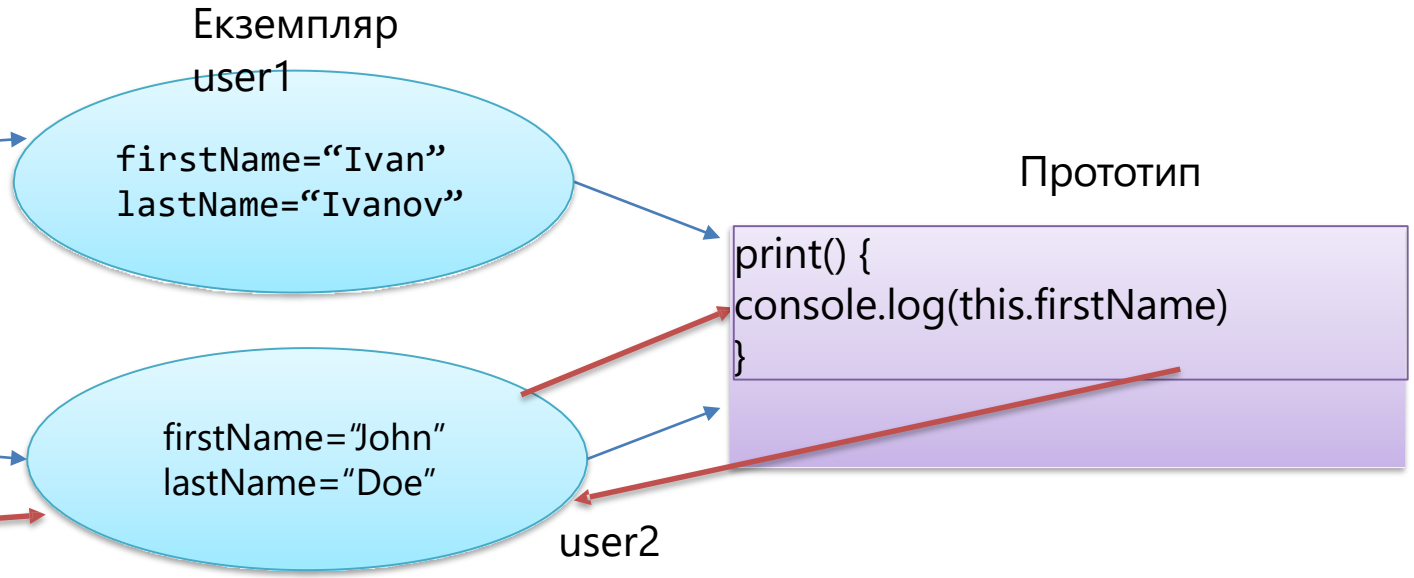
```
class User {
    firstName: string; // свойство
    lastName: string;  // свойство

    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();
```



Конструктор

Конструктор – спеціальний метод, який використовується для ініціалізації екземпляра.

У тілі класу конструктор створюється за допомогою ключового слова constructor.

Під час створення екземпляра класу для виклику конструктора необхідно використовувати ключове слово new.

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

Модифікатори доступа

public, private, protected – ключові слова, модифікатори доступу. З їхньою допомогою визначається видимість елементів класу.

public – видимий для всіх (у класі та за межами класу);
private – видимий тільки в межах класу (за межами класу доступ відсутній);
protected – видимий у межах класу та в класах спадкоємців (за межами класу та класів спадкоємців доступ відсутній).

Усі члени класів без модифікатора доступу, за замовчуванням використовують модифікатор доступу public

Наслідування

Спадкування - механізм створення класу за допомогою розширення вже існуючого класу.

Extends – ключове слово, яке визначає який клас буде базовим (батьківським) для поточного.

Клас-спадкоємець отримує від батьківського властивості та методи

Абстрактні класи

Абстрактний клас - це клас, який може виконувати ролі лише базового класу. Створити екземпляр абстрактного класу не вийде.

Абстрактний метод - це метод, який не має реалізації в поточному класі, але обов'язково має бути реалізований у похідному класі.

Абстрактні методи можуть створюватися лише в абстрактних класах.

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

    abstract makeSound(); // абстрактный метод

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

Интерфейс

Інтерфейс – спеціальний тип даних, що визначає контракт, якого має дотримуватися певний об'єкт.

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

```
class DailyReport implements Report {
    name: string = "Daily Report";

    build() : string {
        return "some daili report data";
    }
}
```

TestProvider Pearson CyberBionic systematics Microsoft Partner Windows Azure EBA

Перевірка знань

TestProvider.com

TestProvider – це online-сервіс перевірки знань з інформаційних технологій. За його допомогою ви можете оцінити свій рівень та виявити слабкі місця. Він буде корисним як у процесі вивчення технології, так і для загальної оцінки знань IT-спеціаліста.

Перевірте, як ви засвоїли даний матеріал на TestProvider.com

Успішне проходження фінального тестування дозволить вам отримати відповідний Сертифікат.

Дякую за увагу! До нової зустрічі!

Ігор Михайличенко
Frontend Engineer