Класи та інтерфейси

№ уроку: 2 **Курс:** TypeScript

Засоби навчання: Visual Studio Code, Atom, Sublime Text, Vim, NotePad++

Огляд, мета та призначення уроку

Мета уроку – ознайомлення студентів із поняттям класу та його конструктором. Ознайомити учнів із модифікаторами доступу, конструкцією **get/set**. Вивчення поняття наслідування та використання абстрактних класів. Знайомство з інтерфейсами та їх застосуванням для функцій та класів.

Вивчивши матеріал даного заняття, учень зможе:

- Використовувати класи замість функцій конструкторів
- Розуміти визначення конструктора у класі
- Використовувати аксесори та модифікатори доступу
- Застосовувати наслідування на практиці
- Використовувати інтерфейси

Зміст уроку

1. Класи у **TypeScript**
2. Конструктори
3. Модифікатори доступу
4. Accessors
5. Наслідування
6. Абстрактні класи
7. Використання інтерфейсів

Резюме

- **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'єкта.
- **Методи** визначають поведінку майбутнього об'єкта.
- Окрім звичайних функцій, класи мають спеціальні функції - конструктори, які визначаються за допомогою ключового слова **constructor**. Конструктори виконують початкову ініціалізацію об'єкта.
- Окрім звичайних властивостей та функцій клас може мати статичні. Для використання статичних функцій та властивостей не треба створювати об'єкт класу. Статичні функції та властивості визначаються за допомогою ключового слова **static**.
- **public**, **private**, **protected** – ключові слова, модифікатори доступу. З їхньою допомогою визначається видимість членів 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-спадкоємцях (за межами класу та класів-спадкоємців доступ відсутній).
- Усі члени класів без модифікатора доступу за замовчанням використовують модифікатор доступу **public**.

CyberBionic systematics

CyberBionic Systematics ®
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine

Tel. 0 800 750 312
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

- Успадкування - механізм створення класу шляхом розширення вже класу, що вже існує.
- **extends** – ключове слово, яке визначає, який клас буде базовим (батьківським) для поточного. Клас спадкоємець отримує від батька властивості та методи.
- **Абстрактний клас -** це клас, який може виступати лише у ролі базового класу. Створити екземпляр абстрактного класу не вдасться. Абстрактний метод - це метод, який не має реалізації у поточному класі, але обов'язково має бути реалізований у похідному класі. Абстрактні методи можуть створюватися лише в абстрактних класах.
- Інтерфейс визначає властивості та методи, які об'єкт повинен реалізувати. **Інтерфейс** - це визначення абстрактного типу даних, але без реалізації. Інтерфейси визначаються за допомогою ключового слова **interface**.

Закріплення матеріалу

- Як у **TypeScript** реалізовано наслідування?
- Що таке абстрактний клас? Абстрактний метод?
- Навіщо використовується ключове слово **implements**?
- У чому різниця між **private** і **protected**?
- Навіщо використовується ключове слово **static**? **readonly**?
- Що таке опціональні параметри в інтерфейсі?
- Навіщо використовується конструктор у класах?

Додаткове завдання

Створіть інтерфейс, що описує поведінку тварини (властивості, способи пересування). Використовуйте цей інтерфейс до класу **Cat, Bird, Fish**. Подумайте, які властивості мають бути опціональними.

Самостійна діяльність учня

Завдання 1
Вивчити основні поняття, розглянуті на уроці.

Завдання 2
Створити поняття абстрактного батьківського класу **Car**. Від нього створити 3 похідні класи (марки автомобілів) із застосуванням методу **super().** У класах використовувати модифікатори як у батьківському класі, і у похідних. Створити від похідних класів мінімум по 2 екземпляри (моделі автомобілів). Методи у похідних класах повинні виводити на екран усі властивості (опис автомобіля). Подумайте, які властивості у похідних класах мають бути **public**, які – **private** та **protected**.

Рекомендовані ресурси

https://www.typescriptlang.org/

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

https://github.com/Microsoft/TypeScript

https://ua.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

CyberBionic systematics
CyberBionic Systematics ®
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine
Tel. 0 800 750 312
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com