Модулі та простір імен

№ уроку: 4 **Курс:** TypeScript

Засоби навчання: Visual Studio, Visual Studio Code, NotePad++

Огляд, мета та призначення уроку

Мета уроку – ознайомлення студентів із простором імен. Вивчити застосування модулів. Ознайомити учнів із базовими параметрами налаштування **Webpack**.

Вивчивши матеріал даного заняття, учень зможе:

- Використовувати роботу з модулями
- Застосовувати функціонал простору імен
- Використовувати різні збирачі модулів
- Застосовувати у своїх додатках збирач **Webpack**

Зміст уроку

1. Використання просторів імен
2. Робота із модулями
3. Налаштування **Webpack**

Резюме

- Для організації великих програм є простір імен. Він містить групу класів, інтерфейсів, функцій, інших просторів імен, які можуть використовуватися в деякому загальному контексті.
- Для визначення простору імен використовується ключове слово **namespace**. Щоб типи та об'єкти, визначені у просторі імен, були видні ззовні, вони визначаються ключовим словом **export**. Простори імен можуть містити і інтерфейси, і об'єкти, і функції.
- За допомогою **директиви /// <reference path="personnel.ts" />** підключаємо файл **personnel.ts**.
- **TypeScript** підтримує роботу з модулями. Модулі є концепцією, яка принесена стандартом **ES2015**, проте в сучасних браузерах нативна підтримка модулів ще не реалізована. Модулі певною мірою схожі на простори імен: вони можуть укладати різні класи, інтерфейси, функції, об'єкти. Також вони виділяються в окремі файли, що дозволяє зробити код програми більш ясним і чистішим, і разом з тим використовувати модулі в інших програмах. Водночас модулі підключаються до програми не за допомогою тега **<script>**, а за допомогою завантажувача модулів.
- Усі модулі мають певний формат і належать до певної системи. Усього є 5 різних систем модулів: **AMD** (**Asynchronys Module Defenition**), **CommonJS, UMD** (**Universal Module Defenition**), **System**, **ES 2015**.
- Для завантаження модулів застосовуються спеціальні завантажувачі:
    - **RequireJS**: використовує синтаксис, відомий як асинхронне визначення модуля або **asynchronous module definition(AMD);**
    - **Browserify**: використовує синтаксис **CommonJS**;
    - **SystemJS**: універсальний завантажувач, який може застосовуватися для модулів будь-якого типу.
- Щоб класи, інтерфейси, функції було видно ззовні, вони визначаються ключовим словом **export**. Щоб використовувати модуль у додатку, його потрібно імпортувати за допомогою оператора **import**.

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

- **Webpack** представляє популярний інструмент для збирання модулів в один файл. Для налаштування використовується файл **webpack.config.js**, який містить конфігурацію **Webpack** з основними параметрами:
    - **entry**: визначає вхідні файли для створення збірок;
    - **output**: визначає конфігурацію вихідних файлів;
    - **resolve**: визначає, як будуть оброблятися файли, якщо вони не мають розширень;
    - **module.rules**: визначає завантажувачі, які завантажують модулі;
    - **plugins**: визначає застосовувані плагіни.

Закріплення матеріалу

- Що таке **triple slash reference**?
- Як імпортувати модуль за замовчуванням?
- Як експортувати всі елементи файлу?
- Що таке **Webpack**?
- Як встановити **Webpack** локально у проєкт?

Додаткове завдання

Встановіть **Webpack** локально собі у проєкт. Налаштуйте всі параметри для компіляції готового файлу **main.js** до верхньої директорії **final**. Встановіть для **Webpack** завантажувальний сервер з портом **8800**.

Самостійна діяльність учня

Завдання 1
Вивчити основні поняття, розглянуті на уроці.

Завдання 2
Використовуючи імпорти **Typescript**, створіть наступну програму з 2 уроку (з абстрактними класами):

- Перший файл - визначені інтерфейси для опису.
- Другий файл – класи (батьківський та похідні).
- Третій файл - екземпляри класів, побудовані на шаблонах інтерфейсів.

Використовуйте будь-який завантажувач модулів.

Рекомендовані ресурси

https://www.typescriptlang.org/

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

https://github.com/Microsoft/TypeScript

https://webpack.js.org/

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