

## Лабораторна робота №2 Класи та інтерфейси

**Мета:** ознайомлення студентів із поняттям класу та його конструктором. Ознайомити студентів із модифікаторами доступу, конструкцією get/set. Вивчення поняття наслідування та використання абстрактних класів. Знайомство з інтерфейсами та їх застосуванням для функцій та класів. Розвинути навички роботи з класами та інтерфейсами в TypeScript, зрозуміти, як взаємодіють об'єкти та типи, а також навчитися створювати складні об'єктні структури.

### Вимоги та самостійна робота:

- JavaScript Classes - <https://uk.javascript.info/classes>
- Classes - <https://www.typescriptlang.org/docs/handbook/2/classes.html>
- Object Types - <https://www.typescriptlang.org/docs/handbook/2/objects.html>
- Cheat sheets - <https://github.com/Computer-Science-Department-ChNU/javascript-frameworks/tree/main/typescript/00-cheat-sheets>

### Практичні завдання:

#### Завдання 1.

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

#### Завдання 2.

1. Створіть інтерфейс Shape, який міститиме методи для обчислення площі (getArea(): number) та периметру (getPerimeter(): number).
2. Реалізуйте класи Circle, Rectangle, та Triangle, які імплементуватимуть інтерфейс Shape. У кожному класі повинні бути відповідні властивості для збереження розмірів фігури.
3. Додайте до кожного класу метод для масштабування фігури (scale(factor: number)), який буде змінювати розміри фігури пропорційно до заданого коефіцієнта.
4. Напишіть код, який створює масив об'єктів типу Shape, додаючи в нього різні фігури, та обчислює загальну площу та периметр усіх фігур у масив

#### Завдання 3.

1. Створити поняття абстрактного батьківського класу Car. Від нього створити 3 похідні класи (марки автомобілів) із застосуванням методу super().
2. У класах використовувати модифікатори як у батьківському класі, і у похідних.

3. Створити від похідних класів мінімум по 2 екземпляри (моделі автомобілів). Методи у похідних класах повинні виводити на екран усі властивості (опис автомобіля).
4. Подумайте, які властивості у похідних класах мають бути public, які – private та protected.

#### **Завдання 4.**

1. Створіть абстрактний клас Employee, який міститиме поля для імені, віку та заробітної плати. Додайте абстрактний метод getAnnualBonus(): number, який повертатиме річний бонус співробітника.
2. Реалізуйте класи Developer та Manager, що наслідуватимуть клас Employee. Клас Developer має повертати бонус у розмірі 10% від зарплати, а клас Manager – 20%.
3. Додайте інтерфейс Payable, який міститиме метод pay(): void. Обидва класи Developer та Manager повинні імплементувати цей інтерфейс.
4. Напишіть код для створення масиву об'єктів типу Employee, який міститиме розробників і менеджерів, та підраховує загальну річну суму бонусів для всіх співробітників.

#### **Завдання 5.**

1. Створіть інтерфейс Course, який міститиме поля для назви курсу, тривалості (в годинах), та масиву студентів.
2. Реалізуйте клас OnlineCourse, який імплементуватиме інтерфейс Course. Додайте метод для реєстрації студентів на курс (registerStudent(student: string): void) та метод для перевірки, чи студент вже зареєстрований на курс (isStudentRegistered(student: string): boolean).
3. Створіть клас CourseManager, який буде керувати масивом курсів. Додайте методи для додавання курсу (addCourse(course: Course): void), видалення курсу (removeCourse(courseName: string): void), та пошуку курсу за назвою (findCourse(courseName: string): Course | undefined).
4. Напишіть код, який створить кілька курсів, додасть їх до CourseManager, зареєструє студентів на деякі курси, та виведе список курсів разом із зареєстрованими студентами.

#### **Завдання 6.**

1. Створіть інтерфейс LibraryItem, який міститиме властивості для назви та автора. Додайте метод borrow(): void, який позначатиме елемент як позичений.
2. Реалізуйте класи Book, Magazine, та DVD, які будуть імплементувати інтерфейс LibraryItem. Кожен клас повинен мати специфічні для нього властивості (наприклад, для книг – кількість сторінок, для журналів – номер випуску, для DVD – тривалість).
3. Створіть клас Library, який буде містити масив бібліотечних елементів. Додайте методи для додавання елементів до бібліотеки (addItem(item:

LibraryItem): void), пошуку елементів за назвою (findItemByName(name: string): LibraryItem | undefined), та виведення списку доступних елементів.

4. Напишіть код, який демонструє роботу бібліотеки, додаючи до неї різні елементи та виконуючи операції з ними.

#### **Контрольні запитання:**

1. Як у TypeScript реалізовано наслідування?
2. Що таке абстрактний клас? Абстрактний метод?
3. Навіщо використовується ключове слово implements?
4. У чому різниця між private і protected?
5. Навіщо використовується ключове слово static? readonly?
6. Що таке опціональні параметри в інтерфейсі?
7. Навіщо використовується конструктор у класах?