

Project (part A)

Тема: Відношення між класами:
асоціація, агрегація, композиція, реалізація

Завдання

1. Обрати і коротко описати **нову** предметну область.
2. Для обраної предметної області з використанням <https://www.lucidchart.com> за два етапи побудувати модель предметної області у вигляді діаграми класів, на якій класи/інтерфейси мають бути пов'язані різними типами відношень: асоціація, агрегація, композиція, реалізація.

Етап 1: Виявлення та визначення перелічень, класів, інтерфейсів та зав'язків між ними (див. приклад на рис. 1). Бажано використовувати всі перелічені у п.2 типи відношень.

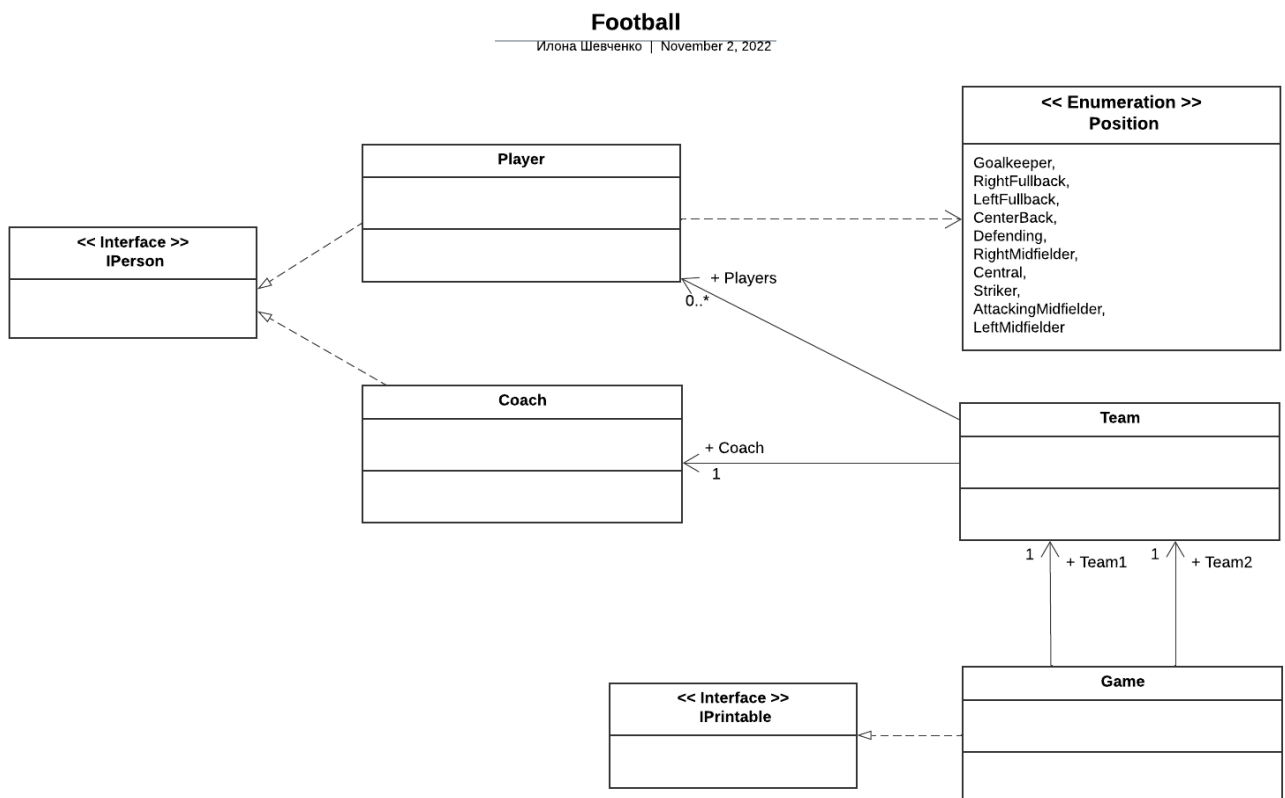


Рис. 1. Виявлення та визначення елементів предметної області та зв'язки між ними

Етап 2: Детальне проєктування класів та інтерфейсів: для полів/властивостей необхідно вказати модифікатор доступу, назву, тип; для методів – додатково визначити тип значення, що повертається, та список параметрів (див. приклад на рис. 2).

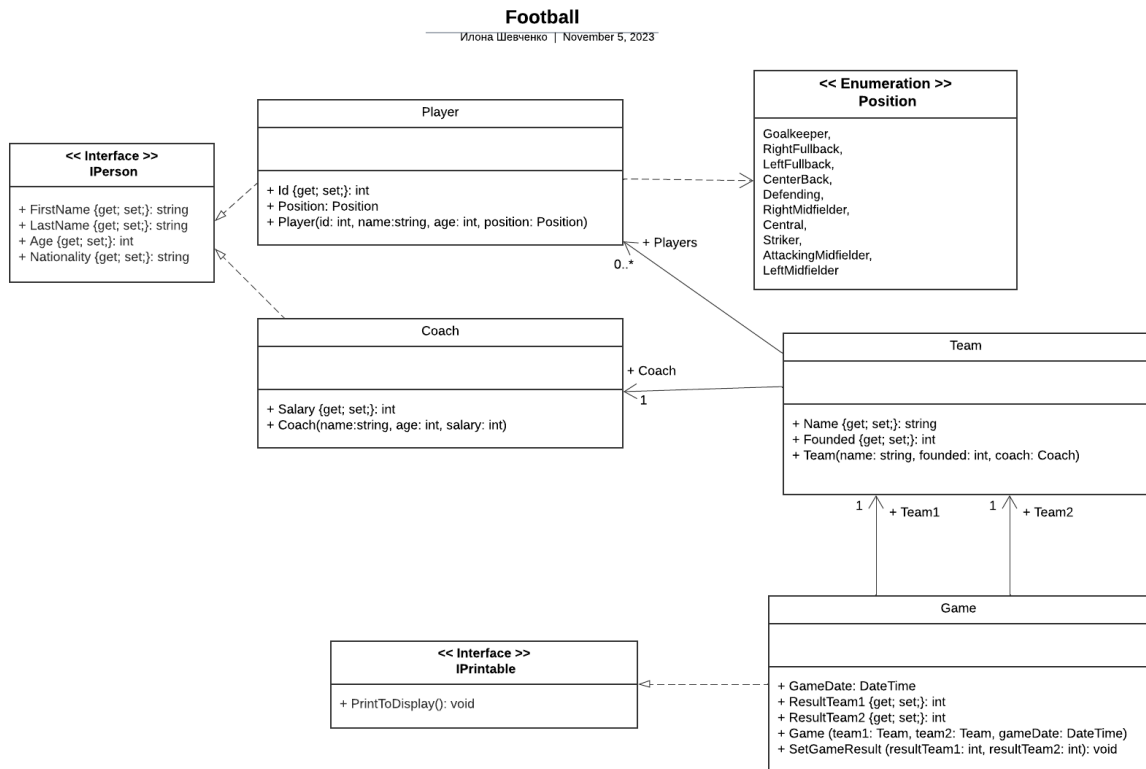


Рис. 2 Детальне проєктування елементів моделі предметної області

3. Засобами середи розробки (IDE) створити каркас проєкту, для цього необхідно визначити в окремих файлах *.cs всі спроектовані у п.2 інтерфейси, класи, перелічення. **Коректно реалізувати відношення між класами!**

Увага! Реалізовувати властивості, конструктори і методи у класах не потрібно! Замість реального коду вони мають містити заглушки ***throw new NotImplementedException()***.

4. Для тестування спроектованих класів додати проєкт з тест-класами, які повинні містити достатній для повноцінного тестування набір **реалізованих** unit-тестів.

Увага! Методи класів предметної області коду ще не мають, але для них вже пишемо повноцінні unit-тести (TDD-принцип розробки програми). Після запуску розроблених unit-тестів їх статус має бути failed.

5. Використання GitHub-репозиторію з коммітами є обов'язковим! Репозиторій повинен мати Readme.md файл, в який необхідно:
 - додати красивий опис предметної області, для цього використовуйте заголовки, списки, рисунки та ін.);
 - вставити розроблену діаграму класів.

6. Оформити звіт:

- Титульний аркуш
- Завдання
- Опис предметної області
- Проектування діаграми класів (аналог рис. 1 і рис. 2)
- Реалізація класів
- Реалізація тест-класів
- Результати запуску unit-тестів.
- Посилання на GitHub-репозиторій

Терміни виконання роботи:

Під час заняття студент має продемонструвати хоча б половину програми.

Студент закінчує розробку програми та оформлює звіт за рахунок годин на самостійну роботу.

Крайній термін здачі програми – 10-й тиждень (наступна лабораторна пара за розкладом).

Крайній термін здачі звіту – 11-й тиждень.

Нарахування балів:

№ п/п	Пункти нарахування балів	Бали
1.	Опис предметної області (п.п. 1)	0..0,5
2.	Побудова діаграми класів (п.п. 2)	0..2
3.	Реалізація каркасу проекту (п.п.3)	0..0,5
4.	Розробка unit-тестів (п.п.4)	0..1
5.	Використання GitHub-репозиторію, комміти, Readme.md-файл	0..1