Частина № 2

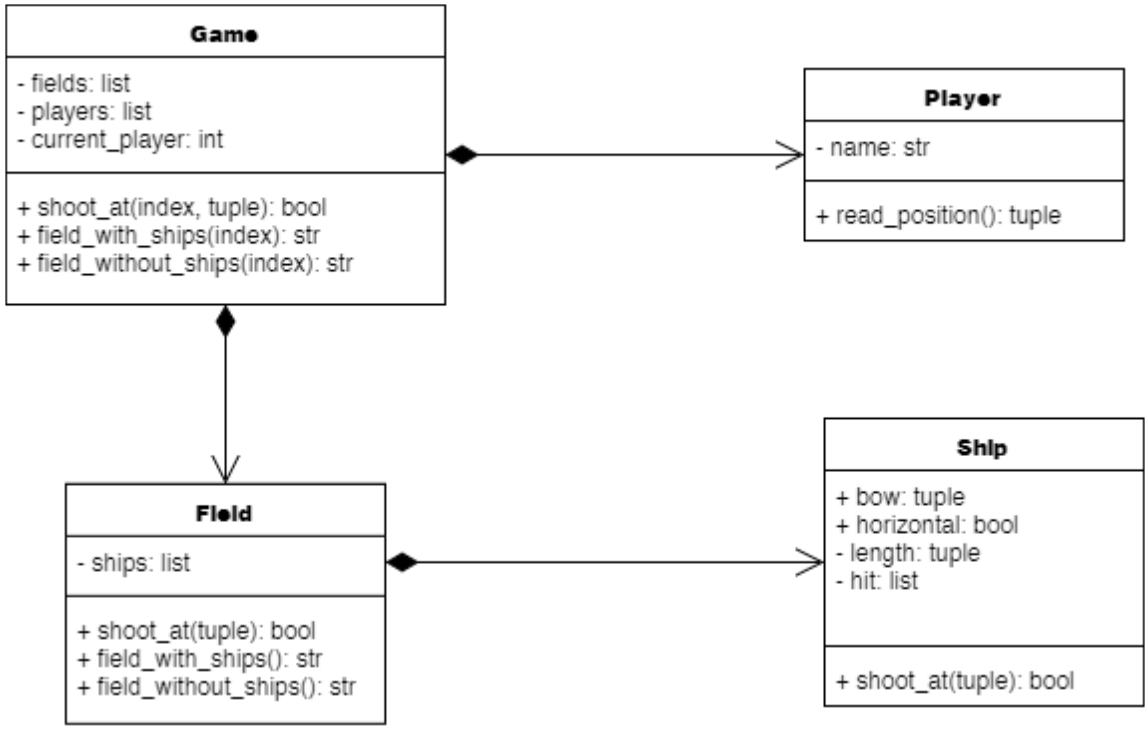
«Морський бій» є грою у якій беруть участь два учасника, які ходять по черзі називаючи координати на невідомому полі суперник. Якщо за координатами за названими суперником знаходить корабель, то корабель або його частина «тоне», а той, хто влучив, здобуває право зробити ще один хід. Перемагає у грі той хто першим потопить усі кораблі суперника. Класично гравці розміщують свої кораблі (чотири кораблі розміром 1 × 1, три кораблі розміром 1 × 2, два кораблі розміром 1 × 3, один корабель розміром 1 × 4) на полі розміром 10 × 10, стовпці цього поля позначають літерами англійського алфавіту від A до J, а рядки числами від 1 до 10.

Реалізуйте морський бій у вигляді консольної гри, у якій гравці по черзі вводять координати пострілу, на дошці суперника. Дошки гравці генеруються випадково. Інтерфейси програми може виглядати, наприклад так

```
Game Battleship
Field of second player:



Player 1, enter move: A1
Field of first player:



Player 2, enter move: B2
Field of second player:
X
```

```
Player 1, enter move: C10
```
і т. д.

Для реалізації проекту використайте наступний опис класів та UML діаграму зображену нижче.



Клас Ship

Атрибути:

- bow — tuple з координатами типу int, що означають розміщення лівого верхнього кута кораблю (корми);
- horizontal — змінна типу bool, що означає напрямок розміщення напрямку;
- length — розмір корабля, наприклад лінкор має розмір (1, 4); атрибут повинен бути приватним;
- hit — список bool, що відповідає тому чи суперник влучив у відповідну частину корабля; атрибут повинен бути приватним

Методи:

- __init__(length) — у ініціалізаторі атрибуту length повинно присвоюватися передане значення, а також повинні створюватися атрибути із значеннями за замовчування;
- shoot_at(tuple) — виконує операцію, яка відображає те, що у класі Ship суперник влучив у частину відповідну частину корабля.

Клас Field

Атрибути:

- ships — двохвимірний список кораблів, елементи якого посиланнями на об’єкти класу Ship; атрибут повинен бути приватним.

Методи:

- __init__() — генерує двохвимірний список, елементами якого є посилання на об’єкти класу Ship. Порожні клітинки можна представити об’єкт класу Ship з особливими параметрами або як None. Для генерації поля ви можете використати раніше реалізовані функції.
- shoot_at(tuple) — виконує операцію, яка означає, що суперник влучив у клітинку ігрового поля Field;.
- field_without_ships() — повертає стрічку, що зображає лише ті клітинки ігрового поля, у які стріляв суперник;

- field_with_ships() — повертає стрічку, що зображає ігрове поле з кораблями, а також усі клітинки у які стріляв суперник.

Клас Player

Атрибути:

- name — ім’я гравця; атрибут повинен бути приватним.

Методи:

- __init__(name) — створює гравця за його ім’ям;
- read_position() — зчитує координати пострілу гравця і конвертує їх у потрібний тип.

Клас Game

Атрибути:

- field — список ігрових полів (об’єкти класу Field); атрибут повинен бути приватним;
- players — список гравців (об’єкти класу Player); атрибут повинен бути приватним;
- current_player — індекс поточного гравця; атрибут повинен бути приватним.

Методи:

- __init__() — ініціалізує нову гру, список гравців, список полів та індекс початкового гравця;
- read_position() — зчитує координати пострілу гравця і конвертує їх у потрібний тип.
- field_without_ships(index) — повертає стрічку, що зображає лише ті клітинки ігрового поля Field з індексом index, у які стріляв суперник;
- field_with_ships() — повертає стрічку, що зображає ігрове поле Field з індексом index з кораблями, а також усі клітинки у які стріляв суперник.

При потребі доповніть класи своїми методами, а також при потребі змініть інтерфейс гри. Наприклад, вам може знадобитися метод, що перевіряє чи у грі наявний переможець.