

Tic Tac Toe

Kółko i krzyżyk

Napisz program do gry w kółko i krzyżyk.

1. Napisz klasę `Board`, reprezentującą planszę do gry, służącą do przechowywania stanu gry. Klasa `Board` powinna posiadać:
 - metodę `get_state(self)` wypisującą na standardowym wyjściu stan planszy,
 - metodę `get_field(self, x, y)` przyjmującą jako parametry indeksy planszy i zwracającą stan pola określonego przez parametry,
 - metodę `set_field(self, move)` przyjmującą jako parametr obiekt klasy `Move` i aktualizującą stan planszy zgodnie z przyjętym parametrem.
2. Napisz klasę `Move` reprezentującą ruch gracza, służącą do przechowywania pojedynczych ruchów w grze.
3. Napisz klasę `Player` reprezentującą gracza, służącą do przechowywania nazwy gracza i wybranego znaku (kółko, krzyżyk). Klasa `Player` powinna posiadać:
 - metodę `get_move(self)`, która wczytuje ze standardowego wejścia współrzędne planszy, na których gracz chce wykonać swój ruch i zwraca obiekt klasy `Move`.
4. Napisz klasę `Game`, która posiada obiekt typu `Board` oraz metodę `play(self, player_one, player_two)`, która dostaje jako argumenty dwa obiekty typu `Player`, przeprowadza grę między nimi wywołując na zmianę metodę wczytującą ruch dla obu obiektów i aktualizując stan planszy, a na koniec zwraca 1, jeżeli wygrał gracz podany w pierwszym argumencie, -1 gdy wygrał gracz podany w drugim argumencie oraz 0 w przypadku remisu. Dla uproszczenia przyjmij, że gracze zawsze wykonują poprawne ruchy.

Przykładowa rozgrywka:

```
player_one = Player("Player One", "O")
player_two = Player("Player Two", "X")
```

```
game = Game()
game.play(player_one, player_two)
```

Klasy niezbędne do utworzenia programu:

```
class Move(object):
    def __init__(self, x, y, sign):

class Player(object):
    def __init__(self, name, sign):
    def get_move(self):

class Board(object):
    def __init__(self):
    def __str__(self):
    def get_state(self):
    def get_field(self, x, y):
    def set_field(self, move):

class Game(object):
    def __init__(self):
    def play(self, player_one, player_two):
    def game_over(self):
    def is_next_move_possible(self):
```