+Print(output : string) : void +GetPlayerInput(prompt : string) : string +fill(std::array<std::array<T, COL>, ROW>& arr, const T& value) : void

-first_player: Player -second_player: Player +StartGame(): void +Game()

Player

-board : shared_ptr<Board>

name : string

-IsInputCorrect(const string& s) : bool

+Player()

+SetName(name : string) : void +SetBoard(b : Board*) : void

+IsWin() : bool +GetName() : string +Play() : void

Board

-board : array<array<char, SIZE_OF_BOARD>,

SIZE_OF_BOARD>

-dies : int

-SIZE OF BOARD: static const int

-IsEmpty(row:int, col:int):bool -IsExist(row:int, col:int): bool

-Dfs(used: array<array<bool, SIZE OF BOARD>

, SIZE_OF_BOARD>&, row : int,

col: int, drow: array<int, 4>, dcol: array<int, 4>): int
-DfsForKill(used: array<array

SIZE_OF_BOARD>
, SIZE_OF_BOARD>&, row: int, col: int, drow: array<int,

4>, dcol : array<int, 4>,

CellsWithShips: vector<pair<int, int>>&): void

-KillShip(CellsWithShips : vector<pair<int, int>>&) : void -lsShipDie(CellsWithShips : vector<pair<int, int>>&) : void

-CheckKill(row: int, col:int): bool -IsNoContactsBetweenShips(): bool

-IsCorrectShips(): bool

-Verify(): bool

+Board()

+ClosePrint(): void +IsItWin(): bool

+Try(row: int, col: int): bool

+Input(): void