**Документация к игре "Морской бой"**

1. **Основные элементы поля:**

* Size: Размер поля (10x10)
* Ship: Символ обозначающий корабль(‘К’)
* Water: Символ, обозначающий воду ('М')
* Hit: Символ, обозначающий попадание ('Р')
* Miss: Символ, обозначающий промах ('П')
* Dead: Символ, обозначающий уничтоженный корабль ('У')
* Unknown: Символ, обозначающий неизвестную клетку ('.')

1. **Class Player:**

* Этот класс представляет игрока и его игровое поле
* field: Двумерный массив, представляющий игровое поле игрока
* visibleField: Двумерный массив, представляющий видимое поле игрока (для отображения игры непосредственно игрокам)

1. **Основные функции:**

**1) void createField(String filepath, Player player):**

Загружает игровое поле из файла

- filepath: Имя файла с игровым полем

- player: Игрок, для которого загружается поле

**2)boolean isValidField(Player player):**

- Проверка корректности данных: Метод проверяет, что каждая строка в файле имеет длину 10 символов и содержит только допустимые символы ('К' и 'М'). Если данные некорректны, выводится сообщение об ошибке

- field: поле игрока

-Возвращает true, если поле корректное, иначе false

**3)boolean isValidShip(Player player):**

-Проверяет, что корабли не соприкасаются (по его периметру только M)

-Проверка на количество и размер кораблей:

* 1 корабль — ряд из 4 клеток
* 2 корабля — ряд из 3 клеток
* 3 корабля — ряд из 2 клеток
* 4 корабля — 1 клетка

- field: поле игрока

-Возвращает true, если корабли и их размещение корректны, иначе false

**4) void printField(Player player):**

Печатает игровое поле. Метод выводит заголовки колонок (буквы A-J) и номера строк (1-10). Печатает содержимое каждой клетки поля, включая символы кораблей, воды, попаданий и промахов.

- visibleField: Игровое поле для печати

**5) boolean isValidPosition(int x, int y):**

Проверяет, находятся ли координаты введённые пользователем в пределах допустимого диапазона (1..10, ‘A’..’J’ включительно)

- Координата X

- Координата Y

- Возвращает true, если координаты допустимы, иначе false

**6)** **boolean takeShot(Player opposer, int x, int y):**

Выполняет выстрел по полю соперника:

Метод проверяет, попадают ли координаты на корабль.

Если выстрел попадает в корабль, обновляет состояние клетки и возвращает true.

Если выстрел не попадает, обновляет состояние клетки на "Промах" и возвращает false.

- opposer: Соперник, по полю которого производится выстрел

- Координата X

- Координата Y

- Возвращает true, если выстрел попал в корабль, иначе false

**7)** **boolean isGameOver(Player player):**

Проверяет, закончилась ли игра

Метод проходит по всем клеткам поля игрока.

Если находит хотя бы один не потопленный корабль, возвращает false.

Если все корабли потоплены, возвращает true.

- player: Игрок, чье поле проверяется

- Возвращает true, если все корабли игрока уничтожены, иначе false

**8)** **void playGame(Player player1, Player player2):**

Основной игровой цикл.

Метод поочередно выполняет ходы игроков до тех пор, пока один из них не победит.

Запрашивает координаты у игрока, проверяет их корректность и выполняет выстрел.

Сообщает результат выстрела (попадание или промах) и передает ход следующему игроку.

- player1: Первый игрок

- player2: Второй игрок

**9)** **void main(String[] args):**

Основной метод программы. Создает двух игроков, загружает их поля и запускает игру