Санкт-Петербургский политехнический университет Институт компьютерных наук и технологий

**Кафедра «Компьютерные системы и программные технологии»**

**КУ Р С О В ОЙ П Р О Е К Т**

**Разработка игры "Battleship"**

по дисциплине «Технологии программирования»

Выполнил студент

гр. 3530901/20001 Арбузова Н. Е.

Преподаватель Степанов Д. С.

31 мая 2023 г.

Санкт-Петербург

2023

Санкт-Петербургский политехнический университет

**ЗАДАНИЕ**

**НА ВЫПОЛНЕНИЕ КУРСОВОГО ПРОЕКТА**

студенту группы 3530901/20001 Арбузовой Надежде Евгеньевне

1. Тема проекта: создание игры Battleship с графическим интерфейсом для игры двух человек.
2. Срок сдачи законченного проекта: 31 мая
3. Исходные данные к проекту: требования к реализовываемому проекту
4. Содержание пояснительной записки: введение с описанием правил игры, основная часть (технологии JavaFX, MVC и их применение в приложении), описание автоматических тестов для бизнес-логики, заключение, список используемых источников.

Дата получения задания: «21» апреля 2023 г.

Руководитель Степанов Д. С.

Задание принял к исполнению Арбузова Н. Е.

21 апреля 2023

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc136390979)

[ОПИСАНИЕ ПРЕДЛОЖЕННОГО РЕШЕНИЯ 4](#_Toc136390980)

[ТЕСТИРОВАНИЕ ПРОГРАММЫ 5](#_Toc136390981)

[ЗАКЛЮЧЕНИЕ 6](#_Toc136390982)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 8](#_Toc136390983)

# ВВЕДЕНИЕ

Цель работы: создать и протестировать игру Battleship с графическим интерфейсом для игры двух человек.

Правила игры:

Battleship - игра для двух человек. У каждого игрока есть игровое поле размером 10 на 10 клеток.  Горизонтали нумеруются сверху вниз, а вертикали помечаются буквами слева направо. При этом используются буквы латинского алфавита от «a» до «j».

1. В начале игроки размещают корабли на свое поле, не показывая противнику. Размещаются: один четырехпалубный, два трехпалубных, три двухпалубных и четыре однопалубных корабля. При размещении корабли не могут касаться друг друга сторонами и углами.
2. Первый ход делает игрок по договоренности.
3. Делая ход, игрок должен выбрать клетку, куда произведет выстрел.
4. Если выстрел пришёлся в клетку, не занятую ни одним кораблём противника, считается, что выстрел произведен мимо, право хода переходит к сопернику.
5. Если выстрел пришёлся в клетку, где находится многопалубный корабль (размером больше, чем 1 клетка), считается, что корабль ранен, кроме одного случая (см. пункт 6). Стрелявший игрок ставит на поле в эту клетку крестик, стрелявший игрок получает право на ещё один выстрел.
6. Если выстрел пришёлся в клетку, где находится однотрубный корабль, или последнюю непоражённую клетку многопалубного корабля, считается, что корабль убит, игрок отмечает потопленный корабль крестиками, а клетки вокруг него точками т. к. на них не может быть кораблей, стрелявший игрок получает право на ещё один выстрел.
7. Победителем считается тот, кто первым потопит все 10 кораблей противника.

# ОПИСАНИЕ ПРЕДЛОЖЕННОГО РЕШЕНИЯ

Для создания графического пользовательского интерфейса (GUI) использовалась библиотека JavaFX, которая обладает большим числом заготовок элементов интерфейса. В частности, были использованы следующие элементы: кнопка (Button), текст (Text), панель (AnchrolPane), выпадающий список (ComboBox), текстовое поле (TextField), панель компоновки (FlowPane), таблица (TableView), сетка (Gridpane), а также вертикальные и горизонтальные ряды (VBox и HBox). Взаимодействие пользователя с графическим элементом описывается в событиях для этого элемента.

Программа была написана с использованием концепция MVC (model- view-controller) для отделения бизнес-логики от визуализации.

В классе Battleship содержится main функция программы, задание параметров окна и расположения внутри него всех элементов интерфейса, а также прописаны события на действия мыши.

Классы Controller, AddPlayerController, PlacementOfShips, GameController содержат объявление всех объектов графического интерфейса, а также методы для работы с ними.

Классы FileSystemPlayer, Round содержат переменные и методы, предназначенные для работы бизнес-логики. Все переменные здесь объявлены как приватные, а для возможности обращения к ним из классов контролеров имеются геттеры.

В соответствии с выбранным шаблоном разработки, пользователь взаимодействует с view, все команды от пользователя обрабатывает controller, который в свою очередь обращается к model и, если это необходимо, перерисовывает view.

# ТЕСТИРОВАНИЕ ПРОГРАММЫ

Для тестирования бизнес-логики было написано 2 автоматических теста, использующих возможности библиотеки JUnit.

Первый тест проверяет правильность работы с информацией об игроках: добавление игрока, изменение и хранение информации о количестве раундов и побед. Второй тест проверяет правильность работы с информацией об расположении кораблей каждого игрока: проверяет, что нельзя разместить корабль так, чтобы часть этого корабля выходила за рамки поля, проверяет, что нельзя разместить два корабля рядом друг с другом.

# ЗАКЛЮЧЕНИЕ

Было создано приложение с графическим интерфейсом, предназначенное для игры двух человек в Battleship. Также были разработаны автоматические тесты для проверки работоспособности кода. В ходе выполнения этого задания мною были изучены библиотека JavaFX и шаблон MVC.

Исходные файлы приложения лежат в репозитории на GitHub: https://github.com/NadyaArbuzova/Battleship.git

Изображение выглядит как текст, снимок экрана, диаграмма, прямоугольный

Автоматически созданное описаниеИзображение выглядит как текст, снимок экрана, диаграмма, прямоугольный

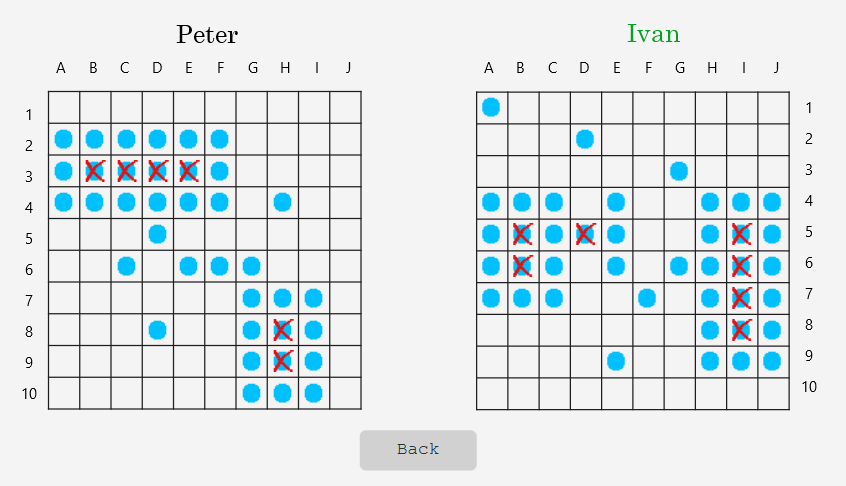
Автоматически созданное описание

Рис 1. Скриншоты приложения

# СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

* 1. https://metanit.com/java/tutorial/ – описание языка Java
  2. https://metanit.com/java/javafx/ – описание JavaFX
  3. https://stackoverflow.com/ - решения технических вопросов
  4. <https://docs.oracle.com/javase/8/> - документация по Java
  5. <https://github.com/twobomb/battleship/blob/master/src/sample/Ship.java> - пример игры