**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**ГОСУДАРСТВЕННОЕ УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**«ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ БЕЛОРУССКОГО ГОСУДАРСТВЕННОГО УНИВЕРСИТЕТА»**

КАФЕДРА ИНФОРМАЦИОННЫХ СИСТЕМ УПРАВЛЕНИЯ

**ОТЧЕТ**

ПО УЧЕБНОЙ ПРАКТИКЕ

(ВАРИАНТ 12, ЛАБОРАТОРНАЯ РАБОТА 12)

Петрова Андрея Александрович

студента 2 курса, группа 14

специальность «Прикладная информатика»

**Руководитель**:

старший преподаватель Орешко И.Г.

Минск, 2021

## **ЗАДАНИЕ №12.**

**Цель работы:** Проанализируйте ваш вариант задания. Можно ли его реализовать как часть MUD системы (например, в одной из комнат MudPlace), требуется ли для этого внести изменения в парадигму MUD? Какие изменения потребует реализация клиента MUD, другие классы примера? Оформите эти размышления в вашем отчёте в качестве анализа предметной области. При реализации, по возможности, используйте парадигму MUD и классы примера 2 при реализации вашего варианта задания.

Создайте на основе технологии RMI клиент/серверное приложение:

Игра по сети в “Морской бой”. Игра между двумя клиентами через сервер. Клиент предлагает поиграть и ждёт, пока другой клиент согласится на игру. Сервер организует связь между играющими клиентами.

**Ход работы:**

На стороне сервера реализован  класс *Worker,* который выполняет всю логику приложения. В частности, он умеет отслеживаться количество подключённых игроков, отслеживать игровой процесс, выводить его логи и отправлять данные, необходимые для игры клиенту.

Непосредственное создание сервера:

Код:

BattleshipInterface obj = new Worker();  
Registry rgsty = LocateRegistry.*createRegistry*(5000);  
rgsty.rebind("BattleshipImplementationobj", obj);

Клиент может запросить у сервера все новости, либо новости за указанный период. Подключение клиента:

Код:

try{  
 obj = (BattleshipInterface)Naming.*lookup*("//localhost:5000/BattleshipImplementationobj");  
}  
catch(Exception e){  
 System.*out*.println("Exception in lookup"+e.getMessage());  
 e.printStackTrace();  
}

Сначала запускается Server, лишь после этого можно запустить Client(ограничение стоит на 2 человека, ибо игроков может быть только 2) для трансляции игрового процесса, а также запросов клиента.

Также в этом задании предлагалось провести рассуждение, подходит ли концепция MUD для реализации данного приложения.

Если говорить коротко, парадигма MUD не подходит для реализации данного задания. К этому есть следующие причины:

MUD само по себе расшифровывается как Multi user domain. Ключевое здесь то, что MUD ориентировано на наличие большего числа взаимодействующих между собой пользователей. В данном задании взаимодействие осуществляется только между сервером и клиентом.

Парадигма MUD хоть и представляет большую гибкость при создании виртуального мира, однако эта гибкость несёт с собой дополнительные расходы. Кажется неразумным поддерживать интерфейс класс MudPlace, если пользоваться будут только методами получения новостей

Единственный вариант внедрения приложения в парадигму MUD – создание отдельной комнат, где можно будет осуществлять несколько игр между разными игроками одновременно. Однако для этого необходимо немного менять интерфейс MUD-парадигмы, а именно теперь описание комнаты должно быть динамически изменяемым (иметь разные id комнаты).

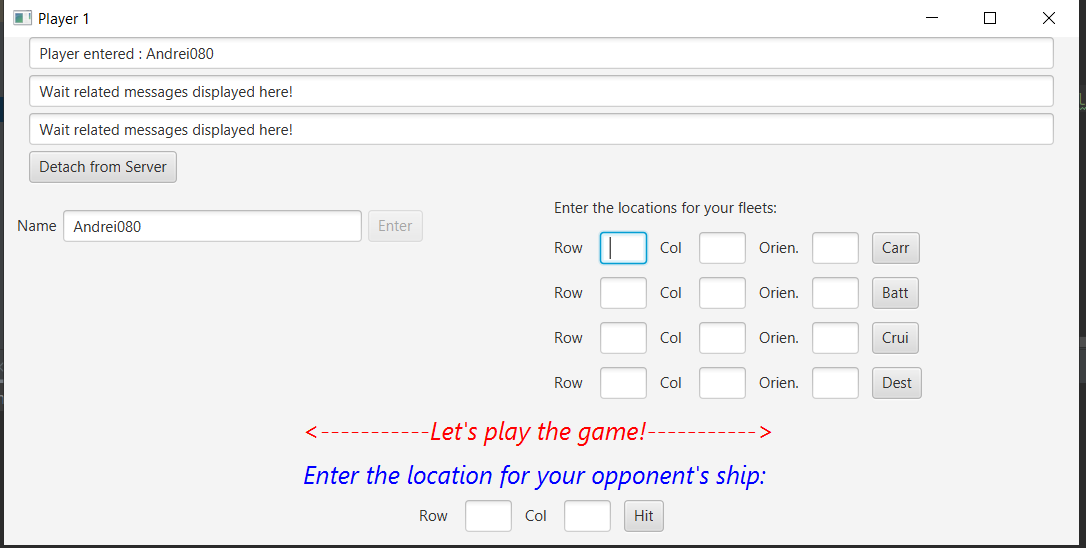


Рисунок 1. Результат выполнения задания №12