**МИНОБРНАУКИ РОССИИ**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**

**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

**Кафедра САПР**

**КУРСОВАЯ РАБОТА**

**по дисциплине «Объектно-ориентированное программирование»**

**Тема: Создание плагинов для игры Minecraft**

|  |  |  |
| --- | --- | --- |
| Студент гр. 3351 |  | Морозов А.А. |
| Преподаватель |  | Кулагин М.В. |

Санкт-Петербург

2024

**ЗАДАНИЕ**

**НА КУРСОВУЮ РАБОТУ**

|  |  |  |
| --- | --- | --- |
| Студент Морозов А.А. | | |
| Группа 3351 | | |
| Тема работы: создание плагинов для игры Minecraft | | |
| Исходные данные:  Java. JDK - Eclipse Temurin 20.0.2. Language level - default | | |
| Содержание пояснительной записки:  Аннотация, введение, плагин регистрации, плагин рангов, плагин комманд, описание программы, вывод, список используемых источников | | |
|  | | |
| Дата выдачи задания: 09.10.2024 | | |
| Дата сдачи реферата: 18.12.2024 | | |
| Дата защиты реферата: 18.12.2024 | | |
| Студент |  | Морозов А.А. |
| Преподаватель |  | Кулагин М.В. |

**АННОТАЦИЯ**

Основное содержание курсового проекта заключается в разработке плагинов для игры Minecraft на версии 1.12.2. Первый плагин реализует систему рангов для игроков, отображая их в табе и чате. Второй плагин добавляет команды для изменения погоды и времени суток, такие как /sun, /rain, /thunder, /day, /night, а также команды для смены игрового режима /gm 1 и /gm 3. Третий плагин обеспечивает регистрацию и авторизацию игроков на сервере. Методы исследования включают анализ существующих решений, проектирование архитектуры плагинов и тестирование их функциональности. Полученные результаты включают успешную интеграцию всех плагинов в игру, улучшение пользовательского опыта и повышение безопасности сервера.

**СОДЕРЖАНИЕ**

1. Введение
2. Плагин регистрации
3. Плагин рангов
4. Плагин комманд
5. Описание программы
6. Вывод
7. Список используемых источников

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# **ВВЕДЕНИЕ**

Цель работы заключается в разработке и интеграции плагинов для игры Minecraft на версии 1.12.2, улучшающих функциональность и безопасность сервера. Основные задачи включают создание системы рангов для игроков с отображением в табе и чате, реализацию команд для управления погодой и временем суток, а также внедрение системы регистрации и авторизации игроков. Методы решения включают анализ требований, проектирование архитектуры плагинов, программирование на Java с использованием API Bukkit, а также тестирование и отладку для обеспечения стабильной работы плагинов.

# **1. ПЛАГИН РЕГИСТРАЦИИ**

## 1.1. Регистрация новых игроков

Варианты использования системы:

1. Зарегистрироваться на сервере

- команда /registration <password> <password>

- команда /reg <password> <password>

В случае неверного написания игрок об этом уведомляется.

Зарегистрироваться на сервере второй и более раз нельзя. Программа проверяет наличие у игрока пароля и не даёт использовать команду более одного раза.

## 1.2. Авторизация уже зарегистрировавшихся игроков

Варианты использования системы:

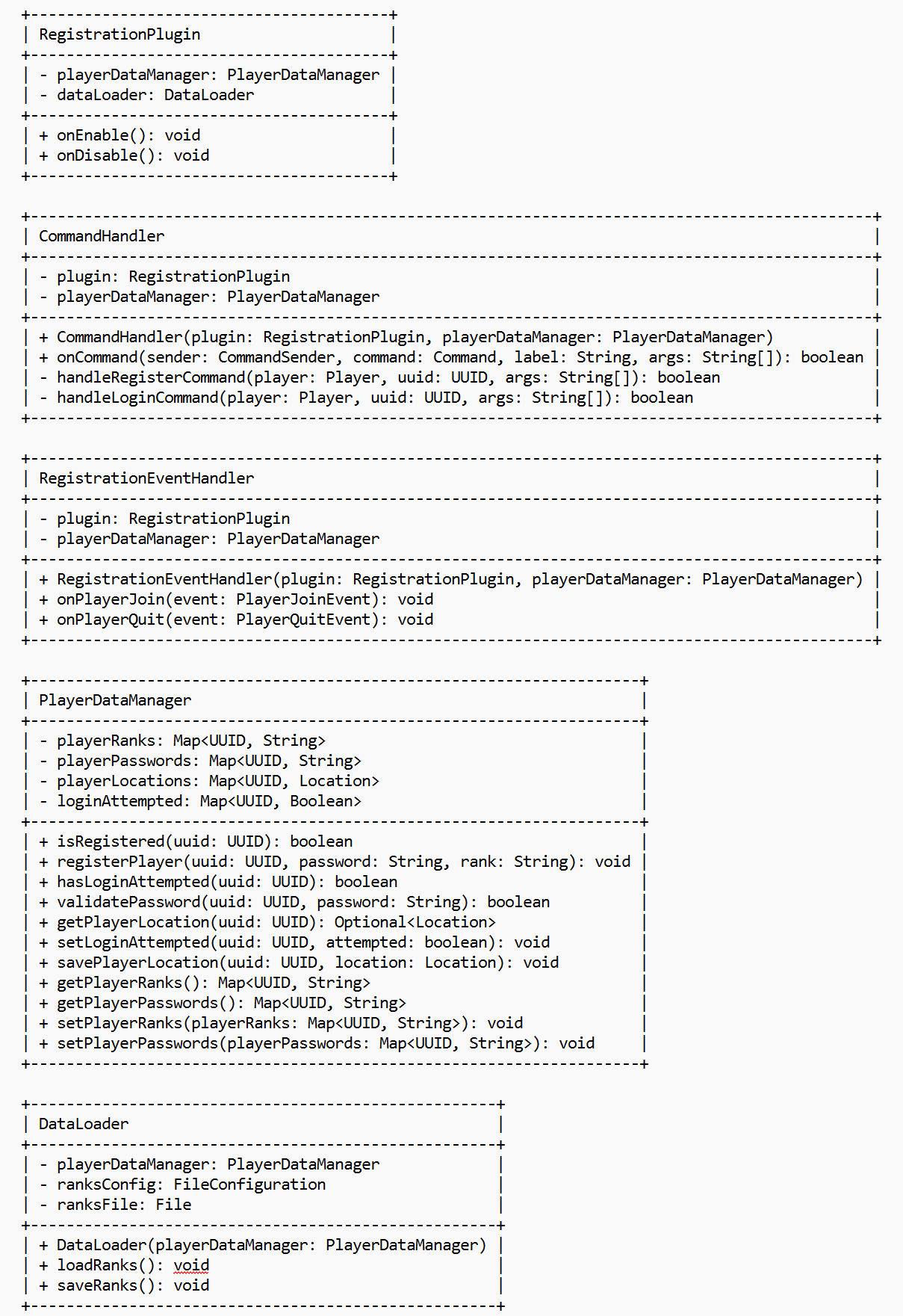
1. Авторизоваться на сервере

- команда /login <password>

- команда /log <password>

В случае неверного написания игрок об этом уведомляется.

Авторизоваться на сервере можно только 1 раз за каждый вход на сервер. Если выйти с сервера и зайти заново, то придётся авторизовываться вновь. Авторизоваться не зарегистрировавшись невозможно.



**2. ПЛАГИН РАНГОВ**

## 2.1 Отображение в табе

В табе (Tab) игроки отображаются в соответствии с их рангами. Более высшие ранги отображаются в табе выше.

Доступные ранги:

- Творец

- Мастер

- Техник

- Пионер

Оператор сервера может выдавать ранги с помощью команды /setrank <name> <rank>

У некоторых рангов в табе организованы анимации

## 2.2 Отображение в чате

В чате игроки отображаются аналогично в соответствии с их рангами. Ранги игроков хранятся в конфигурационном файле ranks.yml. Ранги сохраняются в файл после выхода игрока с сервера. Во время нахождения игрока на сервере, ранги хранятся во временной памяти.

Оператор сервера может выдавать ранги с помощью команды /setrank <name> <rank>

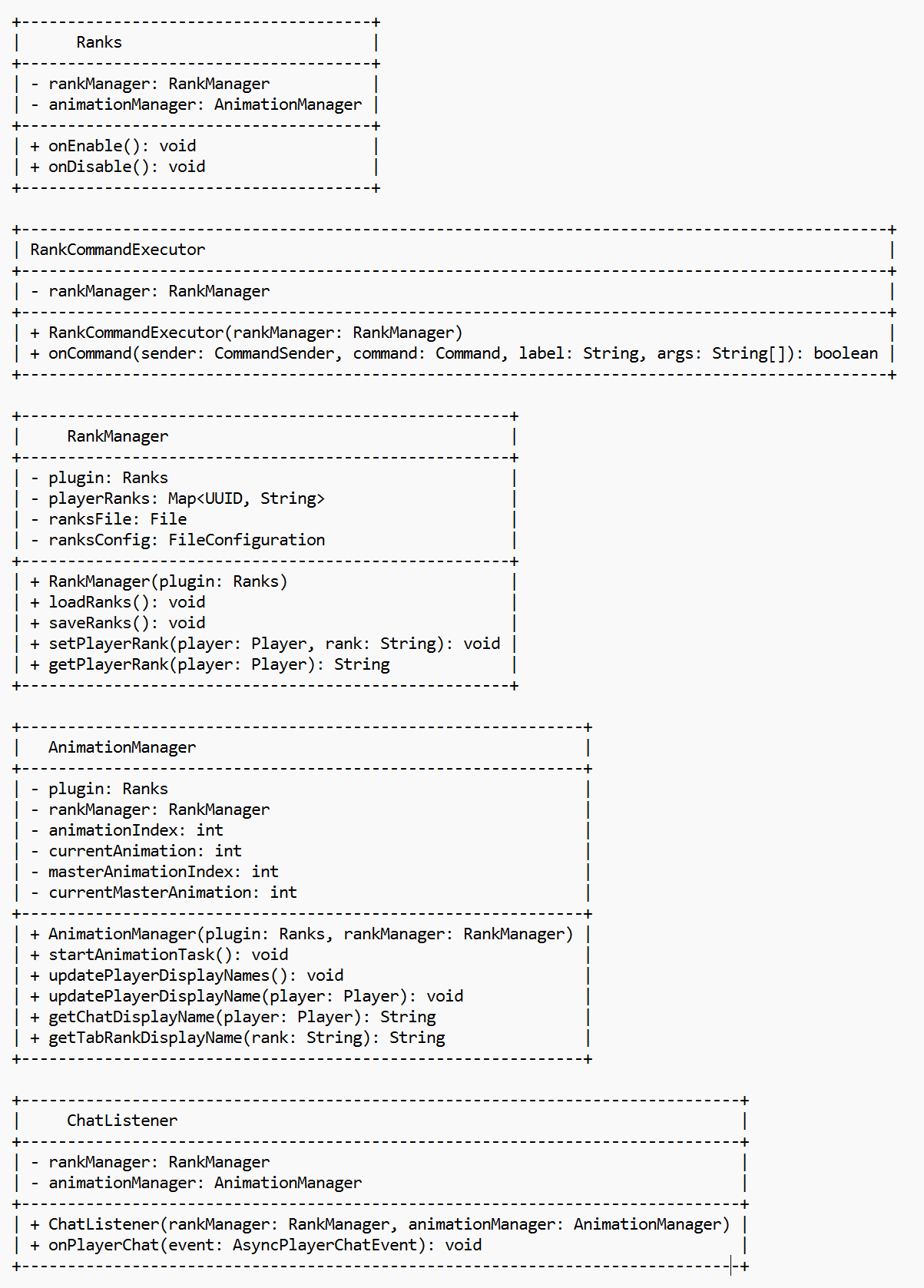
При неправильном написании команды или названия ранга появляется ошибка и пользователь уведомляется

## 2.3 Уровни доступа

Разные команды доступны для одних рангов, но недоступны для других. Например игрок с низшим рангом “Пионер” не может выдать себе креатив с помощью команды /gm 1 или не может поменять погоду на сервере с помощью команд /rain или /sun.

С другой стороны игрок-разработчик с высшим рангом может выполнять любые команды на сервере, например войти в режим наблюдателя с помощью команды /gm 3 или поставить на всём сервере ночь с помощью команды /night.

Игроки, имеющие доступ оператора на сервере, (это высшая мера управления сервером, позволяющая писать любые команды независимо от ранга) могут узнать информацию об игроке с помощью команды /seen <name>. Сюда входит местонахождения игрока в формате мира (обычный, ад, энд) и местоположение игрока в формате координат.



# **3. ПЛАГИН КОМАНД**

Реализует команды управления сервером:

/day

Ставит на сервере время: день

/night

Ставит на сервере время: ночь

/sun

Ставит на сервере погоду: солнце

/rain

Ставит на сервере погоду: дождь

/thunder

Ставит на сервере погоду: гроза

/gm 1

/gamemode 1

/gamemode creative

Ставит игроку режим игры: креатив

/gm 2

/gamemode 2

/gamemode adventure

Ставит игроку режим игры: приключение

/gm 3

/gamemode 3

/gamemode spectator

Ставит игроку режим игры: наблюдатель

/gm 0

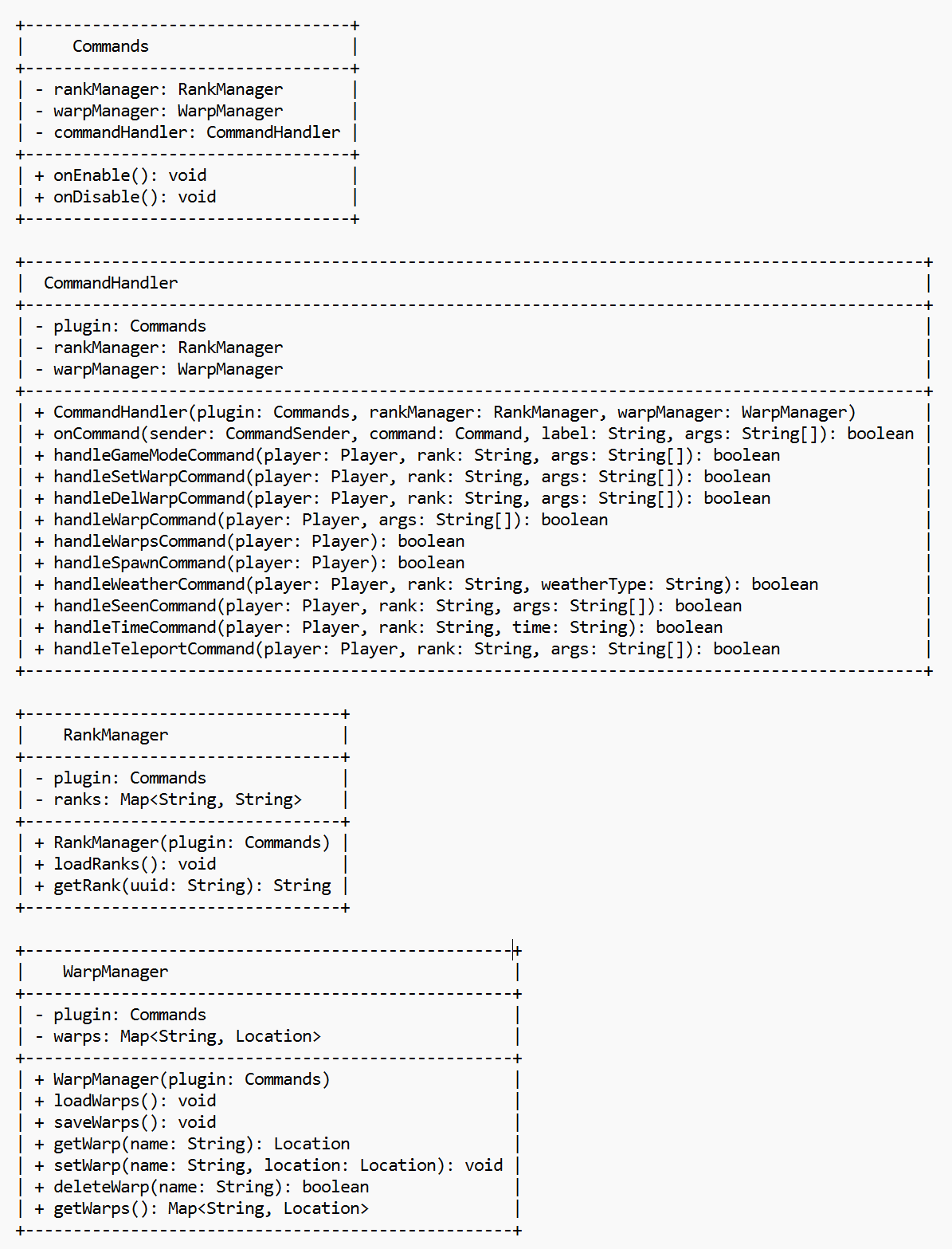
/gamemode 0

/gamemode survival

Ставит игроку режим игры: выживание

/seen

Выводит информацию о местоположении запрашиваемого игрока

****

# **Описание интерфейса пользователя программы**

При первом входе игроку необходимо зарегистрироваться. Это можно сделать с помощью команды /reg <password> <password>. Если игрок уже заходил на сервер, то ему будет необходимо прописать команду /log <password>. Если игрок не зарегистрирован, то он не может авторизоваться. Зарегистрироваться повторно невозможно. При каждом заходе на сервер игроку необходимо авторизовываться заново

В чате и в табе отображаются ранги игроков. Ранг можно установить с помощью команды /setrank <player> <rank>. Эту команду могут писать только операторы сервера. (Рис. 1-3)



Рис. 1 – Таб



Рис. 2 – Таб



Рис. 3 – Таб

В чате ранги отображаются аналогично. Только здесь анимации проигрыватель технически невозможно. (Рис. 4)

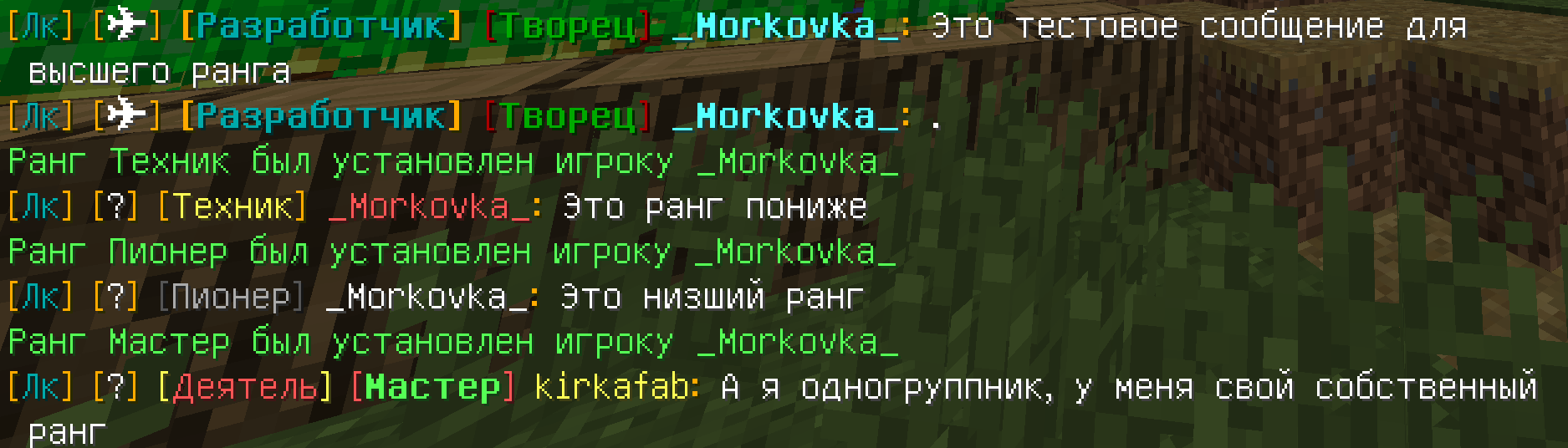


Рис. 4 – Чат сервера

С помощью вышеупомянутых команд можно изменять на сервере погоду или время. (Рис. 5)

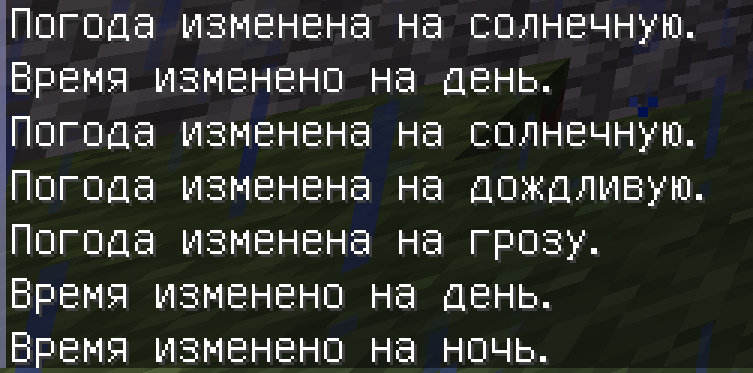


Рис. 5 – Изменение погоды и времени

Также можно узнать местонахождение игрока по нику. Если игрок не в сети, то в чате придёт оповещение. (Рис. 6)

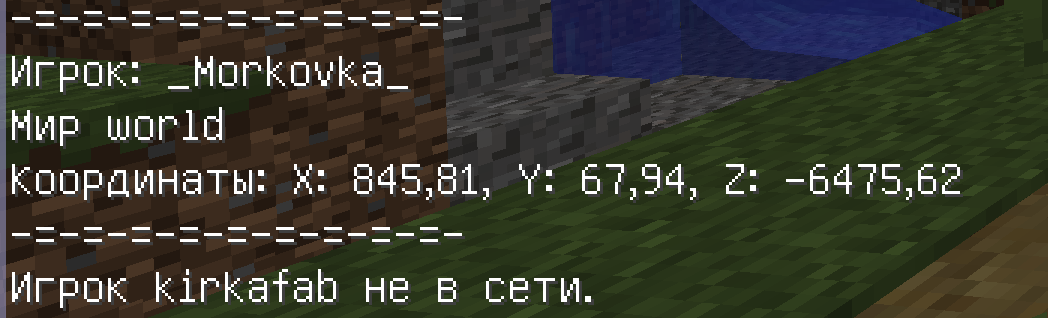


Рис. 6 – Местонахождение игроков

В случае, если ранг игрока не соответствует уровню приоритетов и у пользователя нет доступа к определённым командам, команда не выполнится. В чате игрок будет оповещён, что у него нет прав на эти команды. (Рис. 7)

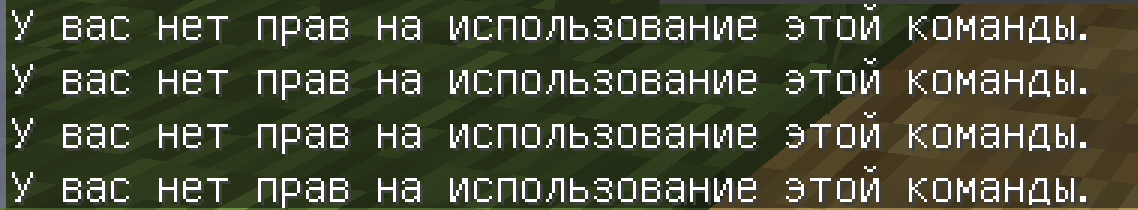


Рис. 7 – Нет прав

**ЗАКЛЮЧЕНИЕ**

В ходе выполнения курсовой работы по дисциплине «Объектно-ориентированное программирование» была успешно реализована система плагинов для игры Minecraft версии 1.12.2. Основные цели и задачи проекта были достигнуты, что подтверждается успешной интеграцией всех разработанных плагинов в игру.

Первый плагин, реализующий систему рангов для игроков, позволил отображать ранги в табе и чате, что значительно улучшило пользовательский опыт. Второй плагин добавил команды для управления погодой и временем суток, такие как **/sun**, **/rain**, **/thunder**, **/day**, **/night**, а также команды для смены игрового режима **/gm 1** и **/gm 3**. Третий плагин обеспечил регистрацию и авторизацию игроков на сервере, что повысило безопасность и контроль доступа.

Методы исследования включали анализ существующих решений, проектирование архитектуры плагинов и тестирование их функциональности. Все плагины были протестированы на различных сценариях использования, что подтвердило их стабильность и корректность работы.

Результаты работы включают:

1. Успешную интеграцию всех плагинов в игру Minecraft.
2. Улучшение пользовательского опыта за счет отображения рангов и добавления новых команд.
3. Повышение безопасности сервера благодаря системе регистрации и авторизации.

Код основных классов объектной модели представлен в приложении. Полный листинг кода можно найти по ссылке на GitHub: <https://github.com/millkun/OOP>

Таким образом, поставленная цель была достигнута, и полученные результаты соответствуют заявленным требованиям.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Документация Java Spigot Bukkit. For Minecraft plugins

**ПРИЛОЖЕНИЕ А**

Код программ:

package me.millkun.commands;  
  
import org.bukkit.plugin.java.JavaPlugin;  
  
public class Commands extends JavaPlugin {  
  
 private RankManager rankManager;  
 private WarpManager warpManager;  
 private CommandHandler commandHandler;  
  
 @Override  
 public void onEnable() {  
 rankManager = new RankManager(this);  
 warpManager = new WarpManager(this);  
 commandHandler = new CommandHandler(this, rankManager, warpManager);  
  
 // Регистрация команд  
 this.getCommand("gamemode").setExecutor(commandHandler);  
 this.getCommand("setwarp").setExecutor(commandHandler);  
 this.getCommand("delwarp").setExecutor(commandHandler);  
 this.getCommand("warp").setExecutor(commandHandler);  
 this.getCommand("warps").setExecutor(commandHandler);  
 this.getCommand("spawn").setExecutor(commandHandler);  
 this.getCommand("sun").setExecutor(commandHandler);  
 this.getCommand("rain").setExecutor(commandHandler);  
 this.getCommand("thunder").setExecutor(commandHandler);  
 this.getCommand("day").setExecutor(commandHandler);  
 this.getCommand("night").setExecutor(commandHandler);  
 this.getCommand("seen").setExecutor(commandHandler);  
 this.getCommand("tp").setExecutor(commandHandler);  
 }  
  
 @Override  
 public void onDisable() {  
 warpManager.saveWarps();  
 }  
}

package me.millkun.commands;  
  
import org.bukkit.Bukkit;  
import org.bukkit.Location;  
import org.bukkit.command.Command;  
import org.bukkit.command.CommandExecutor;  
import org.bukkit.command.CommandSender;  
import org.bukkit.entity.Player;  
  
public class CommandHandler implements CommandExecutor {  
  
 private Commands plugin;  
 private RankManager rankManager;  
 private WarpManager warpManager;  
  
 public CommandHandler(Commands plugin, RankManager rankManager, WarpManager warpManager) {  
 this.plugin = plugin;  
 this.rankManager = rankManager;  
 this.warpManager = warpManager;  
 }  
  
 @Override  
 public boolean onCommand(CommandSender sender, Command command, String label, String[] args) {  
 if (!(sender instanceof Player)) {  
 sender.sendMessage("Эта команда доступна только игрокам.");  
 return true;  
 }  
  
 Player player = (Player) sender;  
 String uuid = player.getUniqueId().toString();  
 String rank = rankManager.getRank(uuid);  
  
 switch (command.getName().toLowerCase()) {  
 case "gamemode":  
 case "gm":  
 return handleGameModeCommand(player, rank, args);  
 case "setwarp":  
 return handleSetWarpCommand(player, rank, args);  
 case "delwarp":  
 return handleDelWarpCommand(player, rank, args);  
 case "warp":  
 return handleWarpCommand(player, args);  
 case "warps":  
 return handleWarpsCommand(player);  
 case "spawn":  
 return handleSpawnCommand(player);  
 case "sun":  
 return handleWeatherCommand(player, rank, "sun");  
 case "rain":  
 return handleWeatherCommand(player, rank, "rain");  
 case "thunder":  
 return handleWeatherCommand(player, rank, "thunder");  
 case "day":  
 return handleTimeCommand(player, rank, "day");  
 case "night":  
 return handleTimeCommand(player, rank, "night");  
 case "seen":  
 return handleSeenCommand(player, rank, args);  
 case "tp":  
 return handleTeleportCommand(player, rank, args);  
 default:  
 return false;  
 }  
 }  
  
 private boolean handleGameModeCommand(Player player, String rank, String[] args) {  
 if (args.length != 1) {  
 player.sendMessage("Использование: /gamemode <mode>");  
 return true;  
 }  
  
 String mode = args[0].toLowerCase();  
 if (rank.equals("Творец") || rank.equals("Мастер")) {  
 switch (mode) {  
 case "1":  
 case "creative":  
 player.setGameMode(org.bukkit.GameMode.*CREATIVE*);  
 player.sendMessage("Вы установили режим креатива.");  
 return true;  
 case "2":  
 case "adventure":  
 player.setGameMode(org.bukkit.GameMode.*ADVENTURE*);  
 player.sendMessage("Вы установили режим приключения.");  
 return true;  
 case "3":  
 case "spectator":  
 if (rank.equals("Творец")) {  
 player.setGameMode(org.bukkit.GameMode.*SPECTATOR*);  
 player.sendMessage("Вы установили режим наблюдателя.");  
 } else {  
 player.sendMessage("У вас нет доступа к этому режиму.");  
 }  
 return true;  
 default:  
 player.sendMessage("Неверный режим. Доступные режимы: creative, adventure, spectator.");  
 return true;  
 }  
 } else {  
 player.sendMessage("У вас нет доступа к этой команде.");  
 return true;  
 }  
 }  
  
 private boolean handleSetWarpCommand(Player player, String rank, String[] args) {  
 if (rank.equals("Творец") || rank.equals("Мастер") || rank.equals("Техник")) {  
 if (args.length != 1) {  
 player.sendMessage("Использование: /setwarp <название>");  
 return true;  
 }  
 String warpName = args[0];  
 Location location = player.getLocation();  
 warpManager.setWarp(warpName, location);  
 player.sendMessage("Варп " + warpName + " установлен.");  
 return true;  
 } else {  
 player.sendMessage("У вас нет доступа к этой команде.");  
 return true;  
 }  
 }  
  
 private boolean handleDelWarpCommand(Player player, String rank, String[] args) {  
 if (rank.equals("Творец") || rank.equals("Мастер") || rank.equals("Техник")) {  
 if (args.length != 1) {  
 player.sendMessage("Использование: /delwarp <название>");  
 return true;  
 }  
 String warpName = args[0];  
 if (warpManager.deleteWarp(warpName)) {  
 player.sendMessage("Варп " + warpName + " удален.");  
 } else {  
 player.sendMessage("Варп с таким названием не существует.");  
 }  
 return true;  
 } else {  
 player.sendMessage("У вас нет доступа к этой команде.");  
 return true;  
 }  
 }  
  
 private boolean handleWarpCommand(Player player, String[] args) {  
 if (args.length != 1) {  
 player.sendMessage("Использование: /warp <название>");  
 return true;  
 }  
 String warpName = args[0];  
 Location warpLocation = warpManager.getWarp(warpName);  
 if (warpLocation != null) {  
 player.teleport(warpLocation);  
 player.sendMessage("Вы телепортированы на варп " + warpName + ".");  
 } else {  
 player.sendMessage("Варп с таким названием не существует.");  
 }  
 return true;  
 }  
  
 private boolean handleWarpsCommand(Player player) {  
 if (warpManager.getWarps().isEmpty()) {  
 player.sendMessage("На сервере нет установленных варпов.");  
 } else {  
 player.sendMessage("Список варпов на сервере:");  
 for (String warpName : warpManager.getWarps().keySet()) {  
 player.sendMessage(warpName);  
 }  
 }  
 return true;  
 }  
  
 private boolean handleSpawnCommand(Player player) {  
 Location spawnLocation = new Location(plugin.getServer().getWorld("world"), 830.5, 68, -6479.5);  
 player.teleport(spawnLocation);  
 player.sendMessage("Вы телепортированы на спавн.");  
 return true;  
 }  
  
 private boolean handleWeatherCommand(Player player, String rank, String weatherType) {  
 if (rank.equals("Творец") || rank.equals("Мастер")) {  
 switch (weatherType) {  
 case "sun":  
 player.getWorld().setStorm(false);  
 player.sendMessage("Погода изменена на солнечную.");  
 break;  
 case "rain":  
 player.getWorld().setStorm(true);  
 player.sendMessage("Погода изменена на дождливую.");  
 break;  
 case "thunder":  
 player.getWorld().setStorm(true);  
 player.getWorld().setThundering(true);  
 player.sendMessage("Погода изменена на грозу.");  
 break;  
 }  
 return true;  
 } else {  
 player.sendMessage("У вас нет доступа к этой команде.");  
 return true;  
 }  
 }  
  
 private boolean handleSeenCommand(Player player, String rank, String[] args) {  
 if (rank.equals("Творец") || rank.equals("Мастер")) {  
 if (args.length != 1) {  
 player.sendMessage("Использование: /seen <имя игрока>");  
 return true;  
 }  
  
 String targetPlayerName = args[0];  
 Player targetPlayer = Bukkit.getPlayer(targetPlayerName);  
 if (targetPlayer == null || !targetPlayer.isOnline()) {  
 player.sendMessage("Игрок " + targetPlayerName + " не в сети.");  
 return true;  
 }  
  
 Location location = targetPlayer.getLocation();  
 String worldName = location.getWorld().getName();  
 String coordinates = String.format("X: %.2f, Y: %.2f, Z: %.2f", location.getX(), location.getY(), location.getZ());  
  
 player.sendMessage("-=-=-=-=-=-=-=-=-=-");  
 player.sendMessage("Игрок: " + targetPlayerName);  
 player.sendMessage("Мир " + worldName);  
 player.sendMessage("Координаты: " + coordinates);  
 player.sendMessage("-=-=-=-=-=-=-=-=-=-");  
 return true;  
 } else {  
 player.sendMessage("У вас нет доступа к этой команде.");  
 return true;  
 }  
 }  
  
 private boolean handleTimeCommand(Player player, String rank, String time) {  
 if (rank.equals("Творец") || rank.equals("Мастер")) {  
 switch (time) {  
 case "day":  
 player.getWorld().setTime(1000);  
 player.sendMessage("Время изменено на день.");  
 break;  
 case "night":  
 player.getWorld().setTime(13000);  
 player.sendMessage("Время изменено на ночь.");  
 break;  
 }  
 return true;  
 } else {  
 player.sendMessage("У вас нет доступа к этой команде.");  
 return true;  
 }  
 }  
  
 private boolean handleTeleportCommand(Player player, String rank, String[] args) {  
 if (rank.equals("Творец")) {  
 if (args.length != 1) {  
 player.sendMessage("Использование: /tp <имя игрока>");  
 return true;  
 }  
  
 String targetPlayerName = args[0];  
 Player targetPlayer = Bukkit.getPlayer(targetPlayerName);  
 if (targetPlayer == null || !targetPlayer.isOnline()) {  
 player.sendMessage("Игрок " + targetPlayerName + " не в сети.");  
 return true;  
 }  
  
 player.teleport(targetPlayer.getLocation());  
 player.sendMessage("Вы телепортированы к игроку " + targetPlayerName + ".");  
 return true;  
 } else {  
 player.sendMessage("У вас нет доступа к этой команде.");  
 return true;  
 }  
 }  
}

package me.millkun.commands;  
  
import org.bukkit.configuration.file.FileConfiguration;  
import org.bukkit.configuration.file.YamlConfiguration;  
  
import java.io.File;  
import java.util.HashMap;  
import java.util.Map;  
  
public class RankManager {  
  
 private Commands plugin;  
 private Map<String, String> ranks = new HashMap<>();  
  
 public RankManager(Commands plugin) {  
 this.plugin = plugin;  
 loadRanks();  
 }  
  
 private void loadRanks() {  
 // Укажите полный путь к файлу ranks.yml  
 File file = new File("D:/jeb32/Server-1.12.2-ETU/ranks.yml");  
  
 // Проверяем, существует ли файл  
 if (!file.exists()) {  
 plugin.getLogger().warning("Файл ranks.yml не найден по пути: " + file.getAbsolutePath());  
 return; // Выходим из метода, если файл не найден  
 }  
  
 // Загружаем конфигурацию из файла  
 FileConfiguration ranksConfig = YamlConfiguration.*loadConfiguration*(file);  
  
 // Загружаем ранги из конфигурации  
 ranksConfig.getConfigurationSection("ranks").getKeys(false).forEach(uuid -> {  
 ranks.put(uuid, ranksConfig.getString("ranks." + uuid));  
 });  
 }  
  
 public String getRank(String uuid) {  
 return ranks.get(uuid);  
 }  
}

package me.millkun.commands;  
  
import org.bukkit.Bukkit;  
import org.bukkit.Location;  
import org.bukkit.configuration.file.FileConfiguration;  
import org.bukkit.configuration.file.YamlConfiguration;  
  
import java.io.File;  
import java.io.IOException;  
import java.util.HashMap;  
import java.util.Map;  
  
public class WarpManager {  
  
 private Commands plugin;  
 private Map<String, Location> warps = new HashMap<>();  
  
 public WarpManager(Commands plugin) {  
 this.plugin = plugin;  
 loadWarps();  
 }  
  
 private void loadWarps() {  
 File file = new File(plugin.getDataFolder(), "warps.yml");  
 if (!file.exists()) {  
 plugin.saveResource("warps.yml", false);  
 }  
 FileConfiguration warpsConfig = YamlConfiguration.*loadConfiguration*(file);  
 for (String key : warpsConfig.getKeys(false)) {  
 double x = warpsConfig.getDouble(key + ".x");  
 double y = warpsConfig.getDouble(key + ".y");  
 double z = warpsConfig.getDouble(key + ".z");  
 String worldName = warpsConfig.getString(key + ".world");  
 Location location = new Location(Bukkit.*getWorld*(worldName), x, y, z);  
 warps.put(key, location);  
 }  
 }  
  
 public void saveWarps() {  
 File file = new File(plugin.getDataFolder(), "warps.yml");  
 FileConfiguration warpsConfig = YamlConfiguration.*loadConfiguration*(file);  
 for (Map.Entry<String, Location> entry : warps.entrySet()) {  
 String key = entry.getKey();  
 Location loc = entry.getValue();  
 warpsConfig.set(key + ".x", loc.getX());  
 warpsConfig.set(key + ".y", loc.getY());  
 warpsConfig.set(key + ".z", loc.getZ());  
 warpsConfig.set(key + ".world", loc.getWorld().getName());  
 }  
 try {  
 warpsConfig.save(file);  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
  
 public Location getWarp(String name) {  
 return warps.get(name);  
 }  
  
 public void setWarp(String name, Location location) {  
 warps.put(name, location);  
 saveWarps();  
 }  
  
 public boolean deleteWarp(String name) {  
 if (warps.containsKey(name)) {  
 warps.remove(name);  
 saveWarps();  
 return true;  
 }  
 return false;  
 }  
  
 public Map<String, Location> getWarps() {  
 return warps;  
 }  
}

package me.millkun.ranks;  
  
import org.bukkit.plugin.java.JavaPlugin;  
  
public final class Ranks extends JavaPlugin {  
  
 private RankManager rankManager;  
 private AnimationManager animationManager;  
  
 @Override  
 public void onEnable() {  
 rankManager = new RankManager(this);  
 animationManager = new AnimationManager(this, rankManager);  
 getCommand("setrank").setExecutor(new RankCommandExecutor(rankManager));  
  
 // Регистрируем слушатель чата  
 getServer().getPluginManager().registerEvents(new ChatListener(rankManager, animationManager), this);  
 }  
  
 @Override  
 public void onDisable() {  
 rankManager.saveRanks();  
 }  
}

package me.millkun.ranks;  
  
import org.bukkit.ChatColor;  
import org.bukkit.command.Command;  
import org.bukkit.command.CommandSender;  
import org.bukkit.entity.Player;  
  
public class RankCommandExecutor implements org.bukkit.command.CommandExecutor {  
  
 private final RankManager rankManager;  
  
 public RankCommandExecutor(RankManager rankManager) {  
 this.rankManager = rankManager;  
 }  
  
 @Override  
 public boolean onCommand(CommandSender sender, Command command, String label, String[] args) {  
 if (command.getName().equalsIgnoreCase("setrank")) {  
 if (!sender.isOp()) {  
 sender.sendMessage(ChatColor.*RED* + "У Вас нет прав на использование этой команды");  
 return false;  
 }  
  
 if (args.length != 2) {  
 sender.sendMessage(ChatColor.*RED* + "/setrank <player> <rank>");  
 return false;  
 }  
  
 Player target = sender.getServer().getPlayer(args[0]);  
 if (target == null) {  
 sender.sendMessage(ChatColor.*RED* + "Игрок не найден");  
 return false;  
 }  
  
 String rank = args[1];  
 rankManager.setPlayerRank(target, rank);  
 sender.sendMessage(ChatColor.*GREEN* + "Ранг " + rank + " был установлен игроку " + target.getName());  
 return true;  
 }  
 return false;  
 }  
}

package me.millkun.ranks;  
  
import org.bukkit.configuration.file.FileConfiguration;  
import org.bukkit.configuration.file.YamlConfiguration;  
import org.bukkit.entity.Player;  
  
import java.io.File;  
import java.io.IOException;  
import java.util.HashMap;  
import java.util.Map;  
import java.util.UUID;  
  
public class RankManager {  
 private final Ranks plugin;  
 private final Map<UUID, String> playerRanks = new HashMap<>();  
 private File ranksFile;  
 private FileConfiguration ranksConfig;  
  
 public RankManager(Ranks plugin) {  
 this.plugin = plugin;  
 loadRanks();  
 }  
  
 public void loadRanks() {  
 // Указываем полный путь к файлу ranks.yml  
 ranksFile = new File("D:\\jeb32\\Server-1.12.2-ETU\\ranks.yml");  
 if (!ranksFile.exists()) {  
 try {  
 ranksFile.getParentFile().mkdirs();  
 ranksFile.createNewFile(); // Создаем файл, если он не существует  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
 ranksConfig = YamlConfiguration.*loadConfiguration*(ranksFile);  
  
 // Загружаем ранги из файла  
 for (String key : ranksConfig.getConfigurationSection("ranks").getKeys(false)) {  
 UUID uuid = UUID.*fromString*(key);  
 String rank = ranksConfig.getString("ranks." + key);  
 playerRanks.put(uuid, rank);  
 }  
 }  
  
 public void saveRanks() {  
 // Сохраняем ранги в файл  
 for (Map.Entry<UUID, String> entry : playerRanks.entrySet()) {  
 ranksConfig.set("ranks." + entry.getKey().toString(), entry.getValue());  
 }  
 try {  
 ranksConfig.save(ranksFile);  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
  
 public void setPlayerRank(Player player, String rank) {  
 playerRanks.put(player.getUniqueId(), rank);  
 // Update display name logic can be added here  
 }  
  
 public String getPlayerRank(Player player) {  
 return playerRanks.getOrDefault(player.getUniqueId(), "Пионер");  
 }  
}

package me.millkun.ranks;  
  
import org.bukkit.Bukkit;  
import org.bukkit.ChatColor;  
import org.bukkit.entity.Player;  
import org.bukkit.scheduler.BukkitRunnable;  
  
public class AnimationManager {  
  
 private final Ranks plugin; // Изменено на Ranks  
 private final RankManager rankManager;  
 private int animationIndex = 0;  
 private int currentAnimation = 0;  
 private int masterAnimationIndex = 0;  
 private int currentMasterAnimation = 0;  
  
 private static final String[][] *ANIMATIONS* = {  
 {  
 "&f✈ &3Разработчик &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &4&l&k?&3азработчик &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3Р&4&l&k?&3зработчик &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3Р&3&oа&4&l&k?&3работчик &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3Р&3&oа&3з&4&l&k?&3аботчик &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&4&l&k?&3ботчик &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&4&l&k?&3отчик &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&4&l&k?&3тчик &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&4&l&k?&3чик &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&4&l&k?&3ик &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&4&l&k?&3к &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&4&l&k?&r &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &4&l&k?&b&lMorkovka\_ &2&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_&4&k?&b&lorkovka\_ &2&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_М&4&l&k?&b&lrkovka\_ &2&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_Мо&4&l&k?&b&lkovka\_ &2&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_Мор&4&l&k?&b&lovka\_ &2&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_Морк&4&l&k?&b&lvka\_ &2&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_Морко&4&l&k?&b&lka\_ &2&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_Морков&4&l&k?&b&la\_ &2&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_Морковк&4&l&k?&b&l\_ &2&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_Морковка&4&l&k?&r &2&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_Морковка\_ &2&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_Морковка\_ &4&l&k?&2&lворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_Морковка\_ &d&lТ&4&l&k?&2&lорец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_Морковка\_ &d&lТв&4&l&k?&2&lрец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_Морковка\_ &d&lТво&4&l&k?&2&lец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_Морковка\_ &d&lТвор&4&l&k?&2&lц",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_Морковка\_ &d&lТворе&4&l&k?",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_Морковка\_ &d&lТворец",  
 },  
 {  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_Морковка\_ &d&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &c&l\_&b&lМорковка\_ &d&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &4&l\_&c&lМ&b&lорковка\_ &d&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &c&l\_&4&lМ&c&lо&b&lрковка\_ &d&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_&c&lМ&4&lо&c&lр&b&lковка\_ &d&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_М&c&lо&4&lр&c&lк&b&lовка\_ &d&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_Мо&c&lр&4&lк&c&lо&b&lвка\_ &d&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_Мор&c&lк&4&lо&c&lв&b&lка\_ &d&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_Морк&c&lо&4&lв&c&lк&b&lа\_ &d&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_Морко&c&lв&4&lк&c&lа&b&l\_ &d&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_Морков&c&lк&4&lа&c&l\_ &d&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_Морковк&c&lа&4&l\_ &d&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_Морковка&c&l\_ &d&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_Морковка\_ &d&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_Морковк&c&lа&4&l\_ &d&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_Морков&c&lк&4&lа&c&l\_ &d&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_Морко&c&lв&4&lк&c&lа&b&l\_ &d&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_Морк&c&lо&4&lв&c&lк&b&lа\_ &d&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_Мор&c&lк&4&lо&c&lв&b&lка\_ &d&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_Мо&c&lр&4&lк&c&lо&b&lвка\_ &d&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_М&c&lо&4&lр&c&lк&b&lовка\_ &d&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_&c&lМ&4&lо&c&lр&b&lковка\_ &d&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &c&l\_&4&lМ&c&lо&b&lрковка\_ &d&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &4&l\_&c&lМ&b&lорковка\_ &d&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &c&l\_&b&lМорковка\_ &d&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_Морковка\_ &d&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_Морковка\_ &d&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_Морковка\_ &d&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_Морковка\_ &d&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_Морковка\_ &d&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_Морковка\_ &d&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_Морковка\_ &d&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_Морковка\_ &d&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &c&l\_Морковка\_ &d&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_Морковка\_ &d&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &c&l\_Морковка\_ &d&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_Морковка\_ &d&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &c&l\_Морковка\_ &d&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_Морковка\_ &d&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &c&l\_Морковка\_ &d&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_Морковка\_ &d&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &c&l\_Морковка\_ &d&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_Морковка\_ &d&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &c&l\_Морковка\_ &d&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_Морковка\_ &d&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &c&l\_Морковка\_ &d&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_Морковка\_ &d&lТворец",  
 },  
 {  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_Морковка\_ &d&lТворец",  
 "&f✈ &3■&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3■ &b&l\_Морковка\_ &d&lТворец",  
 "&f✈ &3■&3■&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3■&3■ &b&l\_Морковка\_ &d&lТворец",  
 "&f✈ &3■&3■&3■&3&oр&3а&3&oб&3о&3&oт&3■&3■&3■ &b&l\_Морковка\_ &d&lТворец",  
 "&f✈ &3■&3■&3■&3■&3а&3&oб&3о&3&o■&3■&3■&3■ &b&l\_Морковка\_ &d&lТворец",  
 "&f✈ &3■&3■&3■&3■&3■&3&oб&3■&3■&3■&3■&3■ &b&l\_Морковка\_ &d&lТворец",  
 "&f✈ &3■&3■&3■&3■&3■&3▣&3■&3■&3■&3■&3■ &b&l\_Морковка\_ &d&lТворец",  
 "&f✈ &3■&3■&3■&3■&3▣&3▣&3▣&3■&3■&3■&3■ &b&l\_Морковка\_ &d&lТворец",  
 "&f✈ &3■&3■&3■&3▣&3▣&3▣&3▣&3▣&3■&3■&3■ &b&l\_Морковка\_ &d&lТворец",  
 "&f✈ &3■&3■&3▣&3▣&3▣&3▣&3▣&3▣&3▣&3■&3■ &b&l\_Морковка\_ &d&lТворец",  
 "&f✈ &3■&3▣&3▣&3▣&3▣&3▣&3▣&3▣&3▣&3▣&3■ &b&l\_Морковка\_ &d&lТворец",  
 "&f✈ &3▢&3▣&3▣&3▣&3▣&3▣&3▣&3▣&3▣&3▣&3▢ &b&l\_Морковка\_ &d&lТворец",  
 "&f✈ &3▢&3▢&3▣&3▣&3▣&3▣&3▣&3▣&3▣&3▢&3▢ &b&l\_Морковка\_ &d&lТворец",  
 "&f✈ &3▢&3▢&3▢&3▣&3▣&3▣&3▣&3▣&3▢&3▢&3▢ &b&l\_Морковка\_ &d&lТворец",  
 "&f✈ &3▢&3▢&3▢&3▢&3▣&3▣&3▣&3▢&3▢&3▢&3▢ &b&l\_Морковка\_ &d&lТворец",  
 "&f✈ &3▢&3▢&3▢&3▢&3▢&3&oб&3▢&3▢&3▢&3▢&3▢ &b&l\_Морковка\_ &d&lТворец",  
 "&f✈ &3▢&3▢&3▢&3▢&3а&3&oб&3о&3▢&3▢&3▢&3▢ &b&l\_Морковка\_ &d&lТворец",  
 "&f✈ &3▢&3▢&3▢&3&oр&3а&3&oб&3о&3&oт&3▢&3▢&3▢ &b&l\_Морковка\_ &d&lТворец",  
 "&f✈ &3▢&3▢&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3▢&3▢ &b&l\_Морковка\_ &d&lТворец",  
 "&f✈ &3▢&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3▢ &b&l\_Морковка\_ &d&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_Морковка\_ &d&lТворец",  
 },  
 {  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_&b&lМорковка\_ &d&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&21&3о&3&oт&3ч&3&oи&3к &b&l\_Морковка\_ &d&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&21&3о&3&oт&3ч&3&oи&3к &b&l\_Мор&20&b&lовка\_ &d&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&21&3о&3&oт&3ч&3&oи&3к &b&l\_&21&b&lор&20&bовка\_ &d&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&21&3о&3&oт&3ч&3&oи&3к &b&l\_&21&b&lор&20&bовка\_ &d&lТвор&20&d&lц",  
 "&f✈ &3Р&3&oа&21&3&oр&3а&21&3о&3&oт&3ч&3&oи&3к &b&l\_&21&b&lор&20&b&lовка\_ &d&lТвор&20&d&lц",  
 "&f✈ &3Р&3&oа&21&3&oр&3а&21&3о&3&oт&3ч&3&oи&3к &b&l\_&21&b&lор&20&b&lов&20&b&lа\_ &d&lТвор&20&d&lц",  
 "&f✈ &3Р&3&oа&21&3&oр&3а&21&3о&3&oт&3ч&3&oи&3к &b&l\_&21&b&lор&20&b&lов&20&b&lа\_ &d&lТ&21&d&lор&20&d&lц",  
 "&f✈ &3Р&3&oа&210&3а&21&3о&3&oт&3ч&3&oи&3к &b&l\_&21&b&lор&20&b&lов&20&b&lа\_ &d&lТ&21&d&lор&20&d&lц ",  
 "&f✈ &3Р&3&oа&210&3а&21&3о&3&oт&3ч&21&3к &b&l\_&21&b&lор&20&b&lов&20&b&lа\_ &d&lТ&21&d&lор&20&d&lц ",  
 "&f✈ &3Р&3&oа&210&3а&21&3о&3&oт&3ч&21&3к &b&l\_&21&b&lор&20&b&lов&20&b&lа\_ &d&lТ&210&d&lр&20&d&lц ",  
 "&f✈ &20&3&oа&210&3а&21&3о&3&oт&3ч&21&3к &b&l\_&21&b&lор&20&b&lо&210&b&lа\_ &d&lТ&210&d&lр&20&d&lц ",  
 "&f✈ &20&3&oа&210&3а&21&3о&3&oт&201&3к &b&l\_&21&b&lо&210&b&lо&210&b&lа\_ &d&lТ&210&d&lр&20&d&lц ",  
 "&f✈ &20&3&oа&210&3а&21&3о&3&oт&201&3к &201&b&lо&210110&b&lа\_ &d&lТ&210&d&lр&20&d&lц ",  
 "&f✈ &20&3&oа&21011&3о&3&oт&201&3к &201&b&lо&210110&b&lа\_ &d&lТ&21010&d&lц ",  
 "&f✈ &20&3&oа&21011&3о&3&oт&2011 &201&b&lо&210110&b&lа\_ &201010&d&lц ",  
 "&f✈ &20&3&oа&210110&3&oт&2011 &201&b&lо&2101100&b&l\_ &2010100 ",  
 "&f✈ &200101101011 &20101011001 &2010100 ",  
 "&f✈ &200101101011 &20101011001 &2010100 ",  
 "&f✈ &200101101011 &20101011001 &2010100 ",  
 "&f✈ &200101101011 &20101011001 &2010100 ",  
 "&f✈ &200101101011 &20101011001 &2010100 ",  
 "&f✈ &200101101011 &20101011001 &2010100 ",  
 "&f✈ &200101101011 &20101011001 &2010100 ",  
 "&f✈ &200101101011 &20101011001 &2010100 ",  
 "&f✈ &200101101011 &20101011001 &2010100 ",  
 "&f✈ &200101101011 &20101011001 &2010100 ",  
 "&f✈ &200101101011 &20101011001 &2010100 ",  
 "&f✈ &3Р&20010110101 &21010101100 &2101010 ",  
 "&f✈ &3Р&3&oа&2001011010 &21101010110 &2010101 ",  
 "&f✈ &3Р&3&oа&3з&200101101 &20110101011 &2001010 ",  
 "&f✈ &3Р&3&oа&3з&3&oр&20010110 &21011010101 &2100101 ",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&2001011 &20101101010 &2110010 ",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&200101 &21010110101 &2011001 ",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&20010 &21101011010 &2101100 ",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&2001 &20110101101 &2010110 ",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&200 &21011010110 &2101011 ",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&20 &20101101011 &2010101 ",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &20010110101 &2101010 ",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &20101101011 &201010&d&lТ ",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &21011010110 &21010&d&lТв ",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &20110101101 &2010&d&lТво ",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &21101011010 &210&d&lТвор ",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &21010110101 &20&d&lТворе ",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &20101101010 &d&lТворец ",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_&2010110101 &d&lТворец ",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_&b&l&k?&201011010 &d&lТворец ",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_&b&l&k??&20101101 &d&lТворец ",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_&b&l&k???&2010110 &d&lТворец ",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_&b&l&k????&201011 &d&lТворец ",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_&b&l&k?????&20101 &d&lТворец ",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_&b&l&k??????&2010 &d&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_&b&l&k???????&201 &d&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_&b&l&k????????&20 &d&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_&b&l&k????????&b&l\_ &d&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_&b&l&k????????&b&l\_ &d&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_&b&l&k????????&b&l\_ &d&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_&b&l&k???????&b&lа\_ &d&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_&b&l&k??????&b&lка\_ &d&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_&b&l&k?????&b&lвка\_ &d&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_&b&l&k????&b&lовка\_ &d&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_&b&l&k???&b&lковка\_ &d&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_&b&l&k??&b&lрковка\_ &d&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_&b&l&k?&b&lорковка\_ &d&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_&b&lМорковка\_ &d&lТворец",  
 },  
 {  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_Морковка\_ &d&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_Морковка\_ &d&lТворе&4&l&k?",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_Морковка\_ &d&lТвор&4&l&k?&2&lц",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_Морковка\_ &d&lТво&4&l&k?&2&lец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_Морковка\_ &d&lТв&4&l&k?&2&lрец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_Морковка\_ &d&lТ&4&l&k?&2&lорец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_Морковка\_ &4&l&k?&2&lворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_Морковка\_ &2&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_Морковка&4&l&k?&r &2&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_Морковк&4&l&k?&b&l\_ &2&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_Морков&4&l&k?&b&la\_ &2&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_Морко&4&l&k?&b&lka\_ &2&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_Морк&4&l&k?&b&lvka\_ &2&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_Мор&4&l&k?&b&lovka\_ &2&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_Мо&4&l&k?&b&lkovka\_ &2&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_М&4&l&k?&b&lrkovka\_ &2&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_&4&k?&b&lorkovka\_ &2&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &4&l&k?&b&lMorkovka\_ &2&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&3к &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&3&oи&4&l&k?&r &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&3ч&4&l&k?&3к &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&3&oт&4&l&k?&3ик &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&3о&4&l&k?&3чик &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&3&oб&4&l&k?&3тчик &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&3а&4&l&k?&3отчик &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3Р&3&oа&3з&3&oр&4&l&k?&3ботчик &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3Р&3&oа&3з&4&l&k?&3аботчик &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3Р&3&oа&4&l&k?&3работчик &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3Р&4&l&k?&3зработчик &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &4&l&k?&3азработчик &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3Разработчик &b&l\_Morkovka\_ &2&lТворец",  
 },  
 {  
 "&f✈ &3Разработчик &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3азработчик &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3зработчик К &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3работчик Ко &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3аботчик Кор &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3ботчик Корн &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3отчик Корне &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3тчик Корнеп &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3чик Корнепл &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3ик Корнепло &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3к Корнеплод &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3 Корнеплод, &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3Корнеплод, &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3орнеплод, ч &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3рнеплод, че &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3неплод, чем &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3еплод, чем &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3плод, чем д &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3лод, чем да &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3од, чем дал &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3д, чем даль &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3, чем дальш &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3 чем дальше &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3чем дальше &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3ем дальше в &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3м дальше в &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3 дальше в л &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3дальше в ле &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3альше в лес &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3льше в лес. &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3ьше в лес. &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3ше в лес. S &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3е в лес. Sk &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3 в лес. Ski &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3в лес. Skib &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3 лес. Skibi &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3лес. Skibid &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3ес. Skibidi &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3с. Skibidi &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3. Skibidi d &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3 Skibidi do &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3Skibidi dop &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3kibidi dop &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3ibidi dop d &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3bidi dop do &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3idi dop dop &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3di dop dop &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3i dop dop y &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3 dop dop ye &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3dop dop yes &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3op dop yes &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3p dop yes y &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3 dop yes ye &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3dop yes yes &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3op yes yes. &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3p yes yes. &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3 yes yes. Р &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3yes yes. Ра &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3es yes. Раз &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3s yes. Разр &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3 yes. Разра &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3yes. Разраб &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3es. Разрабо &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3s. Разработ &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3. Разработч &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3 Разработчи &b&l\_Morkovka\_ &2&lТворец",  
 "&f✈ &3Разработчик &b&l\_Morkovka\_ &2&lТворец",  
 },  
 };  
  
 private static final int[] *ANIMATION\_TICKS* = {3, 2, 2, 2, 3, 2};  
 private static final long[] *ANIMATION\_PAUSES* = {440L, 100L, 100L, 100L, 100L, 200L};  
  
 private static final String[][] *ANIMATIONS2* = {  
 {  
 "&cДеятель &ekirkafab &a&lМастер",  
 "&cтея&fД&cель &ekirkafab &a&lМастер",  
 "&cтеяД&fь&cле &ekirkafab &a&lМастер",  
 "&cте&fь&cДяле &ekirkafab &a&lМастер",  
 "&cтеьДяле &ekirkafab &a&lМастер",  
 "&cтеьДяле &ekirkafab &a&lМастер",  
 "&cтеьДяле &ekirkafab &a&lМастер",  
 "&c&mт&r&cеьДяле &ekirkafab &a&lМастер",  
 "&c&mте&r&cьДяле &ekirkafab &a&lМастер",  
 "&c&mтеь&r&cДяле &ekirkafab &a&lМастер",  
 "&c&mтеьД&r&cяле &ekirkafab &a&lМастер",  
 "&c&mтеьДя&r&cле &ekirkafab &a&lМастер",  
 "&c&mтеьДял&r&cе &ekirkafab &a&lМастер",  
 "&c&mтеьДяле&r &ekirkafab &a&lМастер",  
 "&c&k???????&r &ekirkafab &a&lМастер",  
 "&c&k???????&r &ekirkafab &a&lМастер",  
 "&cДеятель &ekirkafab &a&lМастер",  
 "&cДеятель &ekirkafab &a&lМастер",  
 "&cДеятель &ekirkafab &a&lМастер",  
 "&4>&cеятел&4< &ekirkafab &a&lМастер",  
 "&4 >&cяте&4< &ekirkafab &a&lМастер",  
 "&4 >&cт&4< &ekirkafab &a&lМастер",  
 " &4∧ &ekirkafab &a&lМастер",  
 " &ekirkafab &a&lМастер",  
 " &ekirkafab &a&lМастер",  
 " &4∨ &ekirkafab &a&lМастер",  
 "&4 <&cт&4> &ekirkafab &a&lМастер",  
 "&4 <&cетя&4> &ekirkafab &a&lМастер",  
 "&4<&cлетяе&4> &ekirkafab &a&lМастер",  
 "&cьлетяеД &ekirkafab &a&lМастер",  
 "&cьлетяеД &ekirkafab &a&lМастер",  
 "&cлетяеДе &ekirkafab &a&lМастер",  
 "&cетяеДея &ekirkafab &a&lМастер",  
 "&cтяеДеят &ekirkafab &a&lМастер",  
 "&cяеДеяте &ekirkafab &a&lМастер",  
 "&cеДеятел &ekirkafab &a&lМастер",  
 "&cДеятель &ekirkafab &a&lМастер",  
 },  
 };  
  
 private static final int[] ANIMATION2\_TICKS = {3};  
 private static final long[] ANIMATION2\_PAUSES = {200L};  
  
 public AnimationManager(Ranks plugin, RankManager rankManager) {  
 this.plugin = plugin; // Сохраняем ссылку на экземпляр Ranks  
 this.rankManager = rankManager;  
 startAnimationTask();  
 }  
  
 private void startAnimationTask() {  
 // Таймер для анимации "Творец"  
 new BukkitRunnable() {  
 @Override  
 public void run() {  
 animationIndex++;  
 if (animationIndex >= ANIMATIONS[currentAnimation].length) {  
 animationIndex = 0;  
 currentAnimation = (currentAnimation + 1) % ANIMATIONS.length;  
 // Пауза перед следующей анимацией  
 this.cancel();  
 Bukkit.getScheduler().runTaskLater(plugin, this, ANIMATION\_PAUSES[currentAnimation]);  
 } else {  
 updatePlayerDisplayNames();  
 // Обновляем каждые ANIMATION\_TICKS[currentAnimation] тиков  
 this.cancel();  
 Bukkit.getScheduler().runTaskLater(plugin, this, ANIMATION\_TICKS[currentAnimation]);  
 }  
 }  
 }.runTaskTimer(plugin, 0L, 20L);  
  
 // Таймер для анимации "Мастер"  
 new BukkitRunnable() {  
 @Override  
 public void run() {  
 masterAnimationIndex++;  
 if (masterAnimationIndex >= ANIMATIONS2[currentMasterAnimation].length) {  
 masterAnimationIndex = 0;  
 currentMasterAnimation = (currentMasterAnimation + 1) % ANIMATIONS2.length;  
 // Пауза перед следующей анимацией  
 this.cancel();  
 Bukkit.getScheduler().runTaskLater(plugin, this, ANIMATION2\_PAUSES[currentMasterAnimation]);  
 } else {  
 updatePlayerDisplayNames();  
 // Обновляем каждые ANIMATION2\_TICKS[currentMasterAnimation] тиков  
 this.cancel();  
 Bukkit.getScheduler().runTaskLater(plugin, this, ANIMATION2\_TICKS[currentMasterAnimation]);  
 }  
 }  
 }.runTaskTimer(plugin, 0L, 20L);  
 }  
  
  
 private void updatePlayerDisplayNames() {  
 for (Player player : Bukkit.getOnlinePlayers()) {  
 updatePlayerDisplayName(player);  
 }  
 }  
  
 private void updatePlayerDisplayName(Player player) {  
 String rank = rankManager.getPlayerRank(player);  
 String displayName = rank.equals("Творец") || rank.equals("Мастер") ? getTabRankDisplayName(rank) : player.getName() + " " + getTabRankDisplayName(rank);  
 displayName = displayName.replaceAll("\\[", "").replaceAll("\\]", ""); // Удаляем белые скобки  
 player.setDisplayName(displayName);  
 player.setPlayerListName(displayName);  
 }  
  
 public String getChatDisplayName(Player player) {  
 String rank = rankManager.getPlayerRank(player);  
 String playerName = player.getName();  
 String rankDisplay = getTabRankDisplayName(rank);  
  
 switch (rank) {  
 case "Творец":  
 return ChatColor.translateAlternateColorCodes('&', "&6[&3Лк&6] [&f✈&6] &6&l[&3&lРазработчик&6&l] &4[&2&lТворец&4] &b&l\_Morkovka\_&6: &f");  
 case "Мастер":  
 return ChatColor.translateAlternateColorCodes('&', "&6[&3Лк&6] [&f?&6] &e[&cДеятель&e] &c[&a&lМастер&c] &ekirkafab&6: &f");  
 case "Техник":  
 return ChatColor.translateAlternateColorCodes('&', "&6[&3Лк&6] [&f?&6] &6[&eТехник&6] &c" + playerName + "&6: &f");  
 case "Пионер":  
 return ChatColor.translateAlternateColorCodes('&', "&6[&3Лк&6] [&f?&6] &8[&7Пионер&8] &f" + playerName + "&6: &f");  
 default:  
 return ChatColor.translateAlternateColorCodes('&', "&6[&3Лк&6] [&f?&6] &8[&7Новичок&8] &f" + playerName + "&6: &f");  
 }  
 }  
  
 private String getTabRankDisplayName(String rank) {  
 switch (rank) {  
 case "Творец":  
 return ChatColor.translateAlternateColorCodes('&', ANIMATIONS[currentAnimation][animationIndex]);  
 case "Мастер":  
 return ChatColor.translateAlternateColorCodes('&', ANIMATIONS2[currentMasterAnimation][masterAnimationIndex]);  
 case "Техник":  
 return ChatColor.YELLOW + "Техник ";  
 case "Пионер":  
 return ChatColor.GRAY + "Пионер ";  
 default:  
 return ChatColor.GRAY + "Новичок ";  
 }  
 }  
}

package me.millkun.ranks;  
  
import org.bukkit.ChatColor;  
import org.bukkit.entity.Player;  
import org.bukkit.event.EventHandler;  
import org.bukkit.event.Listener;  
import org.bukkit.event.player.AsyncPlayerChatEvent;  
import org.bukkit.plugin.java.JavaPlugin;  
  
public class ChatListener implements Listener {  
  
 private final RankManager rankManager;  
 private final AnimationManager animationManager;  
  
 public ChatListener(RankManager rankManager, AnimationManager animationManager) {  
 this.rankManager = rankManager;  
 this.animationManager = animationManager;  
 }  
  
 @EventHandler  
 public void onPlayerChat(AsyncPlayerChatEvent event) {  
 Player player = event.getPlayer();  
 String chatDisplayName = animationManager.getChatDisplayName(player);  
 String message = event.getMessage();  
  
 // Отключаем стандартное форматирование чата  
 event.setCancelled(true);  
  
 // Отправляем собственное форматированное сообщение  
 for (Player onlinePlayer : event.getRecipients()) {  
 onlinePlayer.sendMessage(chatDisplayName + message);  
 }  
 }  
}

package me.millkun.registration;  
  
import org.bukkit.plugin.java.JavaPlugin;  
  
public final class RegistrationPlugin extends JavaPlugin {  
  
 private PlayerDataManager playerDataManager;  
 private DataLoader dataLoader;  
  
 @Override  
 public void onEnable() {  
 playerDataManager = new PlayerDataManager();  
 dataLoader = new DataLoader(playerDataManager);  
 dataLoader.loadRanks();  
  
 getCommand("register").setExecutor(new CommandHandler(this, playerDataManager));  
 getCommand("login").setExecutor(new CommandHandler(this, playerDataManager));  
 getServer().getPluginManager().registerEvents(new RegistrationEventHandler(this, playerDataManager), this);  
 }  
  
 @Override  
 public void onDisable() {  
 dataLoader.saveRanks();  
 }  
}

package me.millkun.registration;  
  
import org.bukkit.ChatColor;  
import org.bukkit.GameMode;  
import org.bukkit.Location;  
import org.bukkit.command.Command;  
import org.bukkit.command.CommandExecutor;  
import org.bukkit.command.CommandSender;  
import org.bukkit.entity.Player;  
  
import java.util.UUID;  
  
public class CommandHandler implements CommandExecutor {  
  
 private final RegistrationPlugin plugin;  
 private final PlayerDataManager playerDataManager;  
  
 public CommandHandler(RegistrationPlugin plugin, PlayerDataManager playerDataManager) {  
 this.plugin = plugin;  
 this.playerDataManager = playerDataManager;  
 }  
  
 @Override  
 public boolean onCommand(CommandSender sender, Command command, String label, String[] args) {  
 if (!(sender instanceof Player)) {  
 sender.sendMessage(ChatColor.*RED* + "Эту команду может использовать только игрок.");  
 return false;  
 }  
  
 Player player = (Player) sender;  
 UUID uuid = player.getUniqueId();  
  
 if (command.getName().equalsIgnoreCase("register")) {  
 return handleRegisterCommand(player, uuid, args);  
 } else if (command.getName().equalsIgnoreCase("login")) {  
 return handleLoginCommand(player, uuid, args);  
 }  
 return false;  
 }  
  
 private boolean handleRegisterCommand(Player player, UUID uuid, String[] args) {  
 if (playerDataManager.isRegistered(uuid)) {  
 player.sendMessage(ChatColor.*RED* + "Вы уже зарегистрированы.");  
 return false;  
 }  
  
 if (args.length != 2) {  
 player.sendMessage(ChatColor.*RED* + "/register <password> <password>");  
 return false;  
 }  
  
 String password1 = args[0];  
 String password2 = args[1];  
  
 if (!password1.equals(password2)) {  
 player.sendMessage(ChatColor.*RED* + "Пароли не совпадают.");  
 return false;  
 }  
  
 playerDataManager.registerPlayer(uuid, password1, "Пионер");  
 player.sendMessage(ChatColor.*GREEN* + "Вы успешно зарегистрировались!");  
 player.setGameMode(GameMode.*SURVIVAL*);  
 player.teleport(new Location(player.getWorld(), 830.5, 68, -6479.5));  
 return true;  
 }  
  
 private boolean handleLoginCommand(Player player, UUID uuid, String[] args) {  
 if (!playerDataManager.isRegistered(uuid)) {  
 player.sendMessage(ChatColor.*RED* + "Сначала зарегистрируйтесь.");  
 return false;  
 }  
  
 if (playerDataManager.hasLoginAttempted(uuid)) {  
 player.sendMessage(ChatColor.*RED* + "Вы уже вошли в систему.");  
 return false;  
 }  
  
 if (args.length != 1) {  
 player.sendMessage(ChatColor.*RED* + "/login <password>");  
 return false;  
 }  
  
 String password = args[0];  
 if (playerDataManager.validatePassword(uuid, password)) {  
 player.sendMessage(ChatColor.*GREEN* + "Вы успешно вошли в систему!");  
 player.setGameMode(GameMode.*SURVIVAL*);  
 player.teleport(playerDataManager.getPlayerLocation(uuid).orElse(new Location(player.getWorld(), 830.5, 68, -6479.5)));  
 playerDataManager.setLoginAttempted(uuid, true);  
 return true;  
 } else {  
 player.sendMessage(ChatColor.*RED* + "Неверный пароль.");  
 return false;  
 }  
 }  
}

package me.millkun.registration;  
  
import org.bukkit.ChatColor;  
import org.bukkit.GameMode;  
import org.bukkit.Location;  
import org.bukkit.entity.Player;  
import org.bukkit.event.EventHandler;  
import org.bukkit.event.Listener;  
import org.bukkit.event.player.PlayerJoinEvent;  
import org.bukkit.event.player.PlayerQuitEvent;  
  
import java.util.UUID;  
  
public class RegistrationEventHandler implements Listener {  
  
 private final RegistrationPlugin plugin;  
 private final PlayerDataManager playerDataManager;  
  
 public RegistrationEventHandler(RegistrationPlugin plugin, PlayerDataManager playerDataManager) {  
 this.plugin = plugin;  
 this.playerDataManager = playerDataManager;  
 }  
  
 @EventHandler  
 public void onPlayerJoin(PlayerJoinEvent event) {  
 Player player = event.getPlayer();  
 UUID uuid = player.getUniqueId();  
  
 if (!playerDataManager.isRegistered(uuid)) {  
 player.teleport(new Location(player.getWorld(), 939.5, 40, -6484.5));  
 player.setGameMode(GameMode.*ADVENTURE*);  
 player.sendMessage(ChatColor.*YELLOW* + "Пожалуйста, зарегистрируйтесь с помощью команды /register <пароль> <пароль>");  
 } else {  
 player.teleport(new Location(player.getWorld(), 939.5, 40, -6484.5));  
 player.setGameMode(GameMode.*ADVENTURE*);  
 player.sendMessage(ChatColor.*YELLOW* + "Пожалуйста, войдите в систему с помощью команды /login <пароль>");  
 player.sendMessage(ChatColor.*YELLOW* + "Если не понимаете какой пароль от Вас требуют, возможно ник, с которого Вы зашли, уже занят. Попробуйте изменить его.");  
 playerDataManager.setLoginAttempted(uuid, false);  
 }  
 }  
  
 @EventHandler  
 public void onPlayerQuit(PlayerQuitEvent event) {  
 Player player = event.getPlayer();  
 playerDataManager.savePlayerLocation(player.getUniqueId(), player.getLocation());  
 }  
}

package me.millkun.registration;  
  
import org.bukkit.Location;  
  
import java.util.HashMap;  
import java.util.Map;  
import java.util.Optional;  
import java.util.UUID;  
  
public class PlayerDataManager {  
  
 private final Map<UUID, String> playerRanks = new HashMap<>();  
 private final Map<UUID, String> playerPasswords = new HashMap<>();  
 private final Map<UUID, Location> playerLocations = new HashMap<>();  
 private final Map<UUID, Boolean> loginAttempted = new HashMap<>();  
  
 public boolean isRegistered(UUID uuid) {  
 return playerRanks.containsKey(uuid);  
 }  
  
 public void registerPlayer(UUID uuid, String password, String rank) {  
 playerPasswords.put(uuid, password);  
 playerRanks.put(uuid, rank);  
 }  
  
 public boolean hasLoginAttempted(UUID uuid) {  
 return loginAttempted.getOrDefault(uuid, false);  
 }  
  
 public boolean validatePassword(UUID uuid, String password) {  
 return playerPasswords.containsKey(uuid) && playerPasswords.get(uuid).equals(password);  
 }  
  
 public Optional<Location> getPlayerLocation(UUID uuid) {  
 return Optional.*ofNullable*(playerLocations.get(uuid));  
 }  
  
 public void setLoginAttempted(UUID uuid, boolean attempted) {  
 loginAttempted.put(uuid, attempted);  
 }  
  
 public void savePlayerLocation(UUID uuid, Location location) {  
 playerLocations.put(uuid, location);  
 }  
  
 public Map<UUID, String> getPlayerRanks() {  
 return playerRanks;  
 }  
  
 public Map<UUID, String> getPlayerPasswords() {  
 return playerPasswords;  
 }  
  
 public void setPlayerRanks(Map<UUID, String> playerRanks) {  
 this.playerRanks.clear();  
 this.playerRanks.putAll(playerRanks);  
 }  
  
 public void setPlayerPasswords(Map<UUID, String> playerPasswords) {  
 this.playerPasswords.clear();  
 this.playerPasswords.putAll(playerPasswords);  
 }  
}

package me.millkun.registration;  
  
import org.bukkit.configuration.file.FileConfiguration;  
import org.bukkit.configuration.file.YamlConfiguration;  
  
import java.io.File;  
import java.io.IOException;  
import java.util.Map;  
import java.util.UUID;  
  
public class DataLoader {  
  
 private final PlayerDataManager playerDataManager;  
 private FileConfiguration ranksConfig;  
 private File ranksFile;  
  
 public DataLoader(PlayerDataManager playerDataManager) {  
 this.playerDataManager = playerDataManager;  
 this.ranksFile = new File("D:/jeb32/Server-1.12.2-ETU/ranks.yml");  
 }  
  
 public void loadRanks() {  
 if (!ranksFile.exists()) {  
 ranksFile.getParentFile().mkdirs();  
 try {  
 ranksFile.createNewFile();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
 ranksConfig = YamlConfiguration.*loadConfiguration*(ranksFile);  
  
 for (String key : ranksConfig.getConfigurationSection("ranks").getKeys(false)) {  
 UUID uuid = UUID.*fromString*(key);  
 String rank = ranksConfig.getString("ranks." + key);  
 playerDataManager.getPlayerRanks().put(uuid, rank);  
 }  
  
 for (String key : ranksConfig.getConfigurationSection("passwords").getKeys(false)) {  
 UUID uuid = UUID.*fromString*(key);  
 String password = ranksConfig.getString("passwords." + key);  
 playerDataManager.getPlayerPasswords().put(uuid, password);  
 }  
 }  
  
 public void saveRanks() {  
 for (Map.Entry<UUID, String> entry : playerDataManager.getPlayerRanks().entrySet()) {  
 ranksConfig.set("ranks." + entry.getKey().toString(), entry.getValue());  
 }  
  
 for (Map.Entry<UUID, String> entry : playerDataManager.getPlayerPasswords().entrySet()) {  
 ranksConfig.set("passwords." + entry.getKey().toString(), entry.getValue());  
 }  
  
 try {  
 ranksConfig.save(ranksFile);  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
}