

МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

*Факультет Информационных технологий  
Кафедра Информатики и информационных технологий*

**направление подготовки  
09.03.02 «Информационные системы и технологии»**

## ЛАБОРАТОРНАЯ РАБОТА № 2

**Дисциплина:** «Теория информационных процессов и систем»

**Тема:** Разработка диаграмм классов и объектов для выбранной информационной системы.

**Выполнил:** студент группы 241-334

**Зотов Михаил Павлович**  
(Фамилия И.О.)

**Дата, подпись** 22.09.2025  
(Дата)

  
(Подпись)

**Проверил:** \_\_\_\_\_  
(Фамилия И.О., степень, звание) \_\_\_\_\_  
(Оценка)

**Дата, подпись** \_\_\_\_\_  
(Дата) \_\_\_\_\_  
(Подпись)

**Замечания:** \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

**Москва**

**2025**

**Целью** данной лабораторной работы является выделение сущностей и отношений между ними, а также в данной лабораторной работе, в соответствии с выбранной темой (которую вы выбрали в прошлой лабораторной работе), вы должны будете полностью описать и разработать диаграмму классов и объектов для своей системы, определите основные виды деятельности (назначение основных операций) рассматриваемого предмета исследования.

### **Описание классов**

#### **Атрибуты:**

userId : Int — уникальный идентификатор  
username : String — имя пользователя  
email : String — электронная почта  
passwordHash : String — хэш пароля  
registrationDate : Date — дата регистрации

#### **Методы:**

register() — регистрация нового пользователя  
login() — вход в систему  
createPlaylist(name) — создание плейлиста

### 2. Класс Track (Трек)

#### **Атрибуты:**

trackId : Int — уникальный ID трека  
title : String — название  
artist : String — исполнитель  
album : String — альбом  
duration : Int — длительность (сек.)  
genre : String — жанр  
filePath : String — путь к аудиофайлу

#### **Методы:**

play() — воспроизведение  
getInfo() — получение информации о треке

### 3. Класс Playlist (Плейлист)

#### **Атрибуты:**

playlistId : Int — уникальный ID  
name : String — название плейлиста  
description : String — описание  
creationDate : Date — дата создания

#### **Методы:**

addTrack(track) — добавление трека  
removeTrack(track) — удаление трека  
share() — поделиться плейлистом

### 4. Класс SearchEngine (Поисковый движок)

#### **Атрибуты:**

indexVersion : String — версия индекса  
lastUpdate : Date — дата последнего обновления

#### **Методы:**

search(query) — поиск по запросу  
filterByGenre(genre) — фильтрация по жанру  
sortByPopularity() — сортировка по популярности

### 5. Класс RecommendationSystem (Система рекомендаций)

#### **Атрибуты:**

algorithmType : String — тип алгоритма (collaborative, content-based)  
modelVersion : String — версия модели

**Методы:**

analyzeHistory(user) — анализ истории прослушиваний  
generateRecommendations(user) — формирование рекомендаций  
6. Класс MediaPlayer (Медиаплеер)

**Атрибуты:**

playerId : String — ID плеера  
volume : Int — громкость  
currentTrack : Track — текущий трек

**Методы:**

play(track) — воспроизведение трека  
pause() — пауза  
next() — следующий трек  
previous() — предыдущий трек  
7. Класс MusicDatabase (База данных музыки)

**Атрибуты:**

dbUrl : String — адрес БД  
lastBackup : Date — дата последнего бэкапа

**Методы:**

getTrack(id) — получить трек по ID  
addTrack(track) — добавить трек  
updateTrack(track) — обновить информацию о треке  
8. Класс FileStorage (Файловое хранилище)

**Атрибуты:**

storagePath : String — путь к хранилищу  
totalSpace : Long — общий объем

**Методы:**

uploadFile(file) — загрузка файла  
streamFile(track) — потоковая передача файла

## 2. Описание связей

**User — Playlist**

Тип: **ассоциация (composition)**

Почему: пользователь владеет плейлистами; при удалении пользователя удаляются его плейлисты.

**User — Track**

Тип: **ассоциация (через прослушивания)**

Почему: пользователь слушает треки, система ведет историю прослушиваний.

**Playlist — Track**

Тип: **ассоциация (многие-ко-многим)**

Почему: плейлист содержит множество треков, трек может быть в нескольких плейлистиах.

**SearchEngine — MusicDatabase**

Тип: **зависимость (dependency)**

Почему: поисковый движок зависит от БД для получения данных.

## **RecommendationSystem — User, Track**

Тип: **ассоциация**

Почему: система анализирует поведение пользователя и треки для формирования рекомендаций.

## **MediaPlayer — FileStorage**

Тип: **ассоциации**

Почему: плеер получает аудиофайлы из хранилища.

Рисунок 1. Диаграмма классов

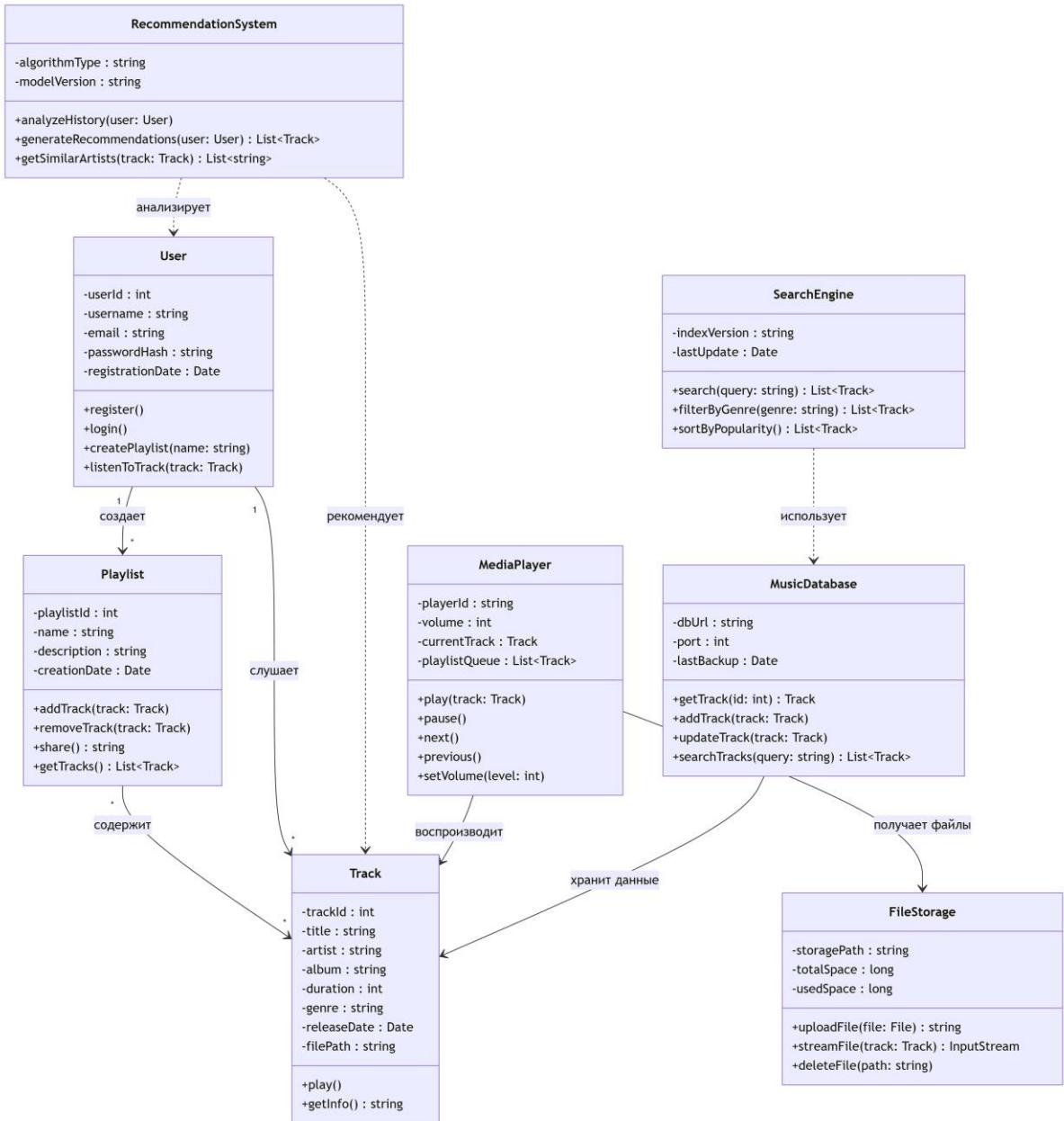
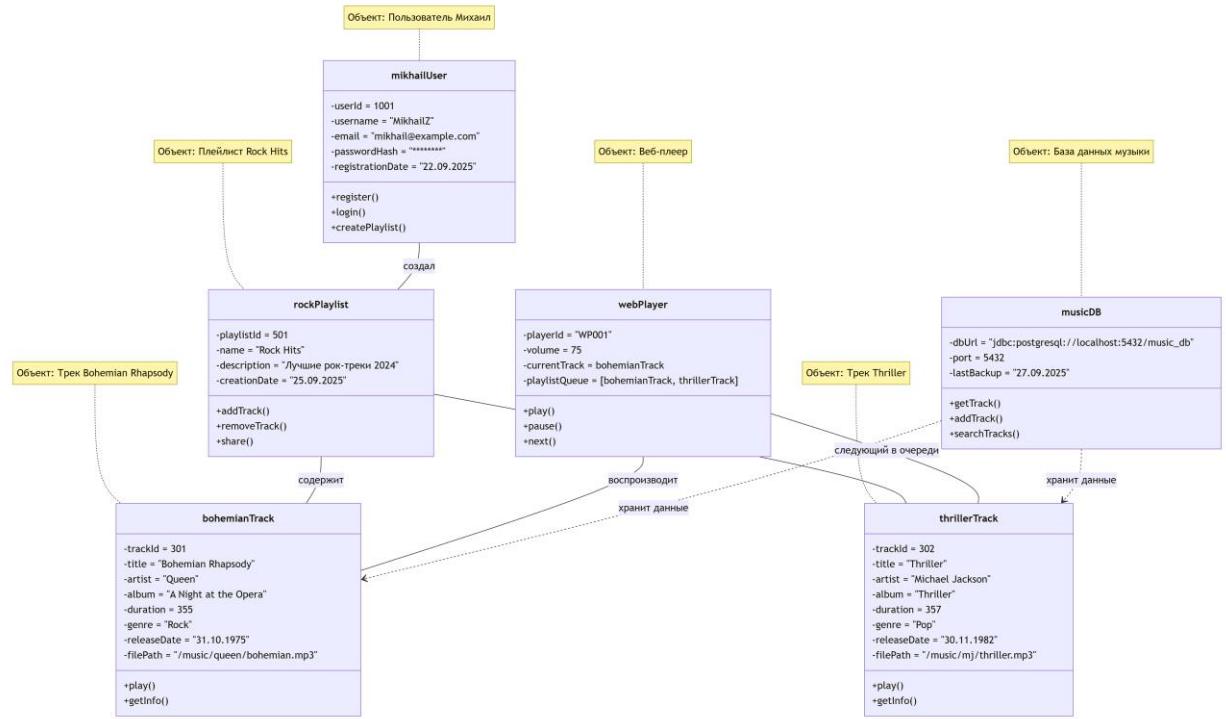


Рисунок 2. Диаграмма объектов



Построение диаграммы объектов. Эта диаграмма позволяет моделировать экземпляры сущностей, которые содержатся в диаграмме классов. На диаграмме объектов показаны множества объектов и отношений между ними в некоторый момент времени. В качестве класса для разработки диаграммы объектов выберем класс “`mikhailUser`”.