**МИНИСТЕРСТВО ТРАНСПОРТА И КОММУНИКАЦИЙ РЕСПУБЛИКИ БЕЛАРУСЬ**

**Учреждение образования**

**«Белорусский государственный университет транспорта»**

**Кафедра «Управление эксплуатационной работой и охрана труда**

**Отчёт  
по лабораторным работам**

по дисциплине «Средства и технологии анализа и разработки информационных систем»

Выполнил Проверил

Студент группы ГИ-31 м.т.н., ст.пр.

Долинова В. А. Козлов В. Г.

Гомель 2024

**ЛАБОРАТОРНАЯ РАБОТА №7  
ТЕМА: СОЗДАНИЕ КЛАССОВ И РАБОТА С ОБЪЕКТАМИ**

**Цель работы:** получить основные понятия и навыки по созданию классов, описанию свойств и работы с объектами.

**Задание**

Вариант 7.

а) Смоделировать структуру музыкальной коллекции: (в отчет добавить диаграмму классов).

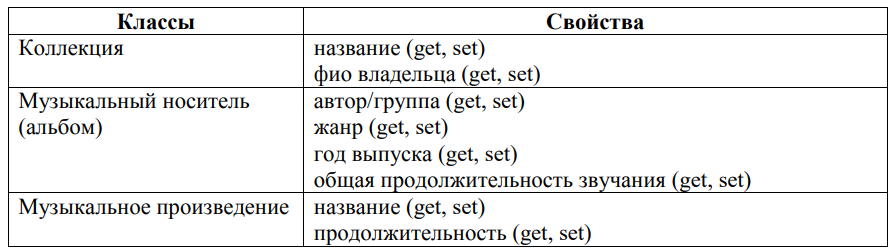


Рисунок 1 – **Структура музыкальной коллекции**

Создать один объект класса Коллекция, два музыкальных носителя (альбома), и три музыкальных произведения, задать значения свойств, вывести на экран.

б) Связать между собой классы Коллекция, Музыкальный носитель и Музыкальное произведение, так, чтобы каждая коллекция содержала музыкальные носители, каждый носитель содержал сведения о том, в какой коллекции он находится, и какие музыкальные произведения он содержит, каждое музыкальное произведение – на каком носителе оно содержится.

- Добавить в класс Коллекция множественное свойство «носители» (get) и методы «добавить носитель» (add), «удалить носитель» (remove).

- Добавить в класс Музыкальный носитель свойство «коллекция» (get) и свойство «музыкальные произведения» (get), а также методы «добавить музыкальное произведение» (add), «удалить музыкальное произведение» (remove). Удалить set для свойства «общая продолжительность звучания» и сделать так, чтобы данное свойство рассчитывалось автоматически (на основе свойства «музыкальные произведения»).

- Добавить в класс «Музыкальное произведение» свойство «Музыкальный носитель».

- Создать один объект класса Коллекция, в данную коллекцию добавить два носителя - альбома, в первый альбом добавить два произведения, во второй – одно произведение. Вывести на экран музыкальные произведения одного из носителей и общую продолжительность звучания альбома.

в) добавить в класс Коллекция метод, осуществляющий поиск музыкального произведения по названию. Метод содержит входной параметр Название (String) и возвращает значение типа Музыкальное произведение. Осуществить поиск в main(), вывести информацию по найденному произведению на экран, и отобразить на каком носителе оно содержится.

**Листинг**

//класс Main.java

public class Main {

public static void main(String[] args) {

MusicalCollection collection = new MusicalCollection("Популярные альбомы", "Долинова Виктория");

MusicCarrier album1 = new MusicCarrier("The Slim Shady LP", "Eminem", "Хип-хоп", 1999);

MusicCarrier album2 = new MusicCarrier("Starboy", "The Weeknd", "Поп", 2016);

MusicPiece song1 = new MusicPiece("My Name Is", 290);

MusicPiece song2 = new MusicPiece("Role Model", 205);

MusicPiece song3 = new MusicPiece("If I Had", 245);

collection.addCarrier(album1);

collection.addCarrier(album2);

album1.addMusicPiece(song1);

album1.addMusicPiece(song2);

album2.addMusicPiece(song3);

System.out.println("Название коллекции: " + collection.getName());

System.out.println("Владелец: " + collection.getOwner());

album1.displayMusicPieces();

int totalDuration = album1.getMusicPieces().stream().mapToInt(MusicPiece::getDuration).sum();

System.out.println("Общая продолжительность звучания первого альбома: " + totalDuration + "с");

album2.displayMusicPieces();

int totalDuration2 = album2.getMusicPieces().stream().mapToInt(MusicPiece::getDuration).sum();

System.out.println("Общая продолжительность звучания второго альбома: " + totalDuration2 + "с");

album1.findMusicPieceByName("My Name Is");

}}

//класс MusicalCollection.java

import java.util.HashSet;

import java.util.Set;

public class MusicalCollection {

private String name;

private String owner;

private Set<MusicCarrier> carriers;

public MusicalCollection(String name, String owner) {

this.name = name;

this.owner = owner;

this.carriers = new HashSet<>();

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getOwner() {

return owner;

}

public void setOwner(String owner) {

this.owner = owner;

}

public void addCarrier(MusicCarrier carrier) {

carriers.add(carrier);

}

public void removeCarrier(MusicCarrier carrier) {

carriers.remove(carrier);

}

public void displayCarriers() {

System.out.println("Музыкальная коллекция: " + name);

System.out.println("Владелец: " + owner);

System.out.println("Название:");

for (MusicCarrier carrier : carriers) {

System.out.println(carrier.getName());

}

}

}

//класс MusicCarrier.java

import java.util.HashSet;

import java.util.Set;

public class MusicCarrier {

private String name;

private String author;

private String genre;

private int year;

private Set<MusicPiece> musicPieces;

private MusicalCollection collection;

public MusicCarrier(String name, String author, String genre, int year) {

this.name = name;

this.author = author;

this.genre = genre;

this.year = year;

this.musicPieces = new HashSet<>();

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getAuthor() {

return author;

}

public void setAuthor(String author) {

this.author = author;

}

public String getGenre() {

return genre;

}

public void setGenre(String genre) {

this.genre = genre;

}

public int getYear() {

return year;

}

public void setYear(Integer year){

this.year = year;

}

public Set<MusicPiece> getMusicPieces() {

return musicPieces;

}

public void addMusicPiece(MusicPiece musicPiece) {

musicPieces.add(musicPiece);

musicPiece.setCarrier(this);

}

public void removeMusicPiece(MusicPiece musicPiece) {

musicPieces.remove(musicPiece);

musicPiece.setCarrier(null);

}

public void setCollection(MusicalCollection collection) {

this.collection = collection;

}

public MusicalCollection getCollection() {

return collection;

}

public MusicPiece findMusicPieceByName(String musicname) {

for (MusicPiece musicPiece : musicPieces) {

if (musicPiece.getName().equals(musicname)) {

System.out.println("Найдено музыкальное произведение:");

System.out.println(musicPiece.getName());

System.out.println("Находится на носителе: " + musicPiece.getCarrier());

return musicPiece;

}

}

return null;

}

public void displayMusicPieces() {

System.out.println("Альбом: " + name);

System.out.println("Автор альбома: " + author);

System.out.println("Жанр: " + genre);

System.out.println("Год выпуска: " + year);

System.out.println("Музыкальное произведение:");

for (MusicPiece musicPiece : musicPieces) {

System.out.println(musicPiece.getName());

}

}

}

//класс MusicPiece.java

public class MusicPiece {

private String name;

private int duration;

private MusicCarrier carrier;

public MusicPiece(String name, int duration) {

this.name = name;

this.duration = duration;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public int getDuration() {

return duration;

}

public void setDuration(int duration) {

this.duration = duration;

}

public void setCarrier(MusicCarrier carrier) {

this.carrier = carrier;

}

public String getCarrier() {

return carrier.getName();

}

}

**Результат**

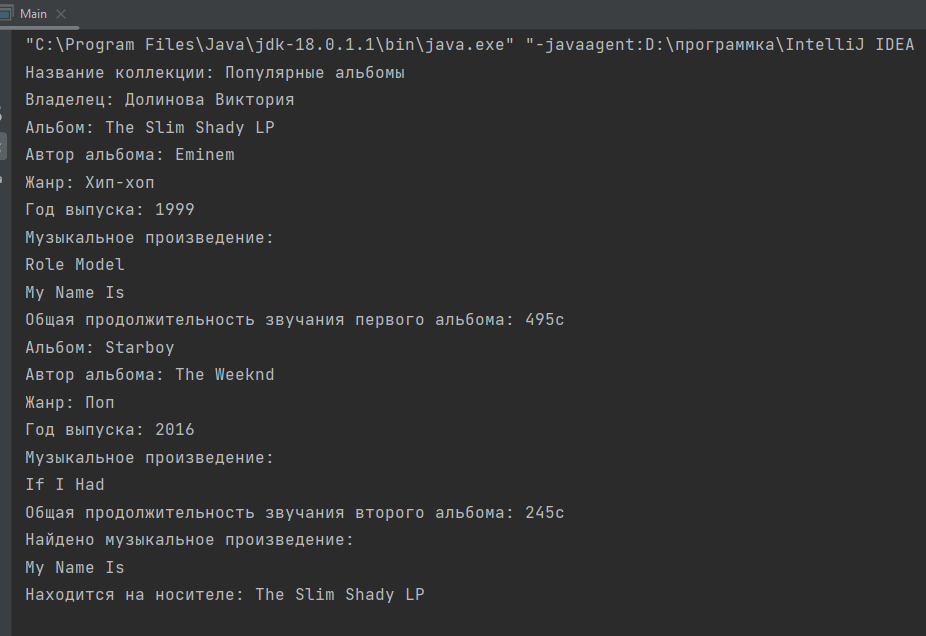
****

Рисунок 2 – **Результат работы программы «Музыкальные коллекции»**

На рисунке 2 видно, что были созданы следующие объекты: один объект класса Коллекция, два музыкальных носителя (альбома) и три музыкальных произведения, в первый альбом добавлено два произведения, во второй – одно произведение. Кроме того, заданы значения свойств каждого из объектов, а результат выведен на экран, а также приведен итог автоматического расчета (на основе свойства «музыкальные произведения») свойства «общая продолжительность звучания альбома».

Также необходимо отметить, что на рисунке 2 представлен результат работы метода, который осуществляет поиск музыкального произведения по названию. Информация по найденному произведению отображена на экран, в том числе о носителе, на котором оно содержится.

Диаграмма классов разработанной программы представлена на рисунке 3.

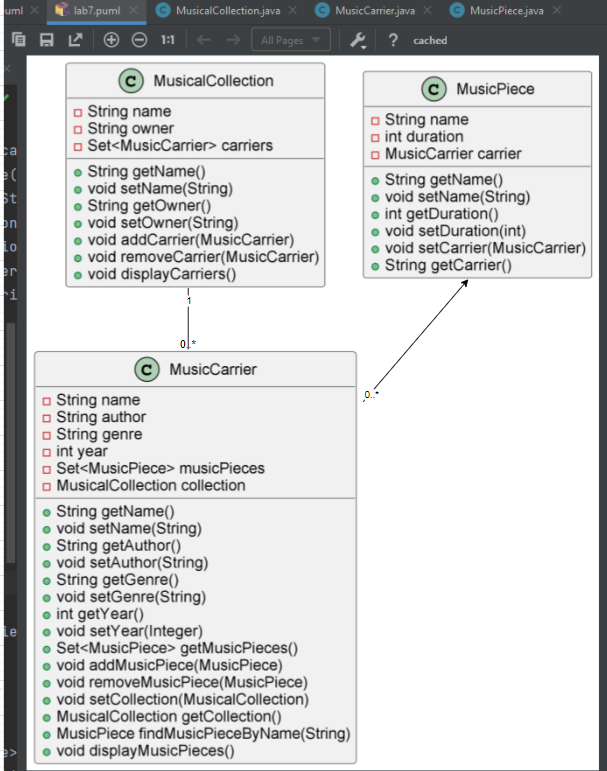


Рисунок 3 – **Диаграмма классов программы «Музыкальные коллекции»**

Как видно из рисунка 3, созданы классы «MusicalCollection», «MusicalCarrier», «MusicalPiece», описаны поля классов со спецификаторами доступа private для реализации принципа инкапсуляции, а методы (свойства) – с public для получения доступа к приватным полям посредством чтения get() и записи set(). Кроме того, для установления значения полей в каждом из классов представлен конструктор класса.

В объектно-ориентированном проектировании взаимосвязи (ассоциации) между классами описываются с помощью отношения композиции. Участниками отношения являются два взаимосвязанных класса и ассоциация. Ассоциация может быть однонаправленной или двунаправленной. Для каждого окончания ассоциации определяется «арность», т.е. ограничение на количество объектов данного класса, которые могут взаимодействовать с указанным количеством объектов другого класса.

Так, между классами «MusicalCollection» и «MusicalCarrier» установлена двунаправленная ассоциация, которая означает двустороннюю связь между классами, когда первый класс содержит ссылки на объекты второго класса, а второй класс содержит ссылки на объекты первого, а между классами «MusicalCarrier», «MusicalPiece» – однонаправленная, которая означает одностороннюю связь между двумя классами, когда один класс содержит одну или много ссылок на объекты второго класса. Взаимосвязь посредством ссылок в данном случае представлена созданием векторов Set<MusicCarrier>, Set<MusicPiece>, которые относятся к набору коллекций и не могут включать повторяющиеся элементы, содержат реализацию в классах HashSet, использующих хэш-таблицу для хранения с механизмом хеширования.

**Вывод:** получены основные понятия и навыки по созданию классов, описанию свойств и работы с объектами на языке программирования Java в среде программирования IntelliJ IDEA Community Edition 2022.2.2. В результате в программе «Музыкальные коллекции» разработаны классы «Коллекция», «Музыкальный носитель», «Музыкальное произведение», описаны их поля и методы, настроены связи и реализованы такие функции, как «Поиск музыкального произведения по названию» и другие, а также созданы объекты каждого из классов и выведены на экран для визуального представления работы программы.