**ЛАБОРАТОРНАЯ РАБОТА №7**

**СОЗДАНИЕ КЛАССОВ И РАБОТА С ОБЪЕКТАМИ**

**Цель работы**: получить основные понятия и навыки по созданию классов,

описанию свойств и работы с объектами.

Необходимая теоретическая подготовка:

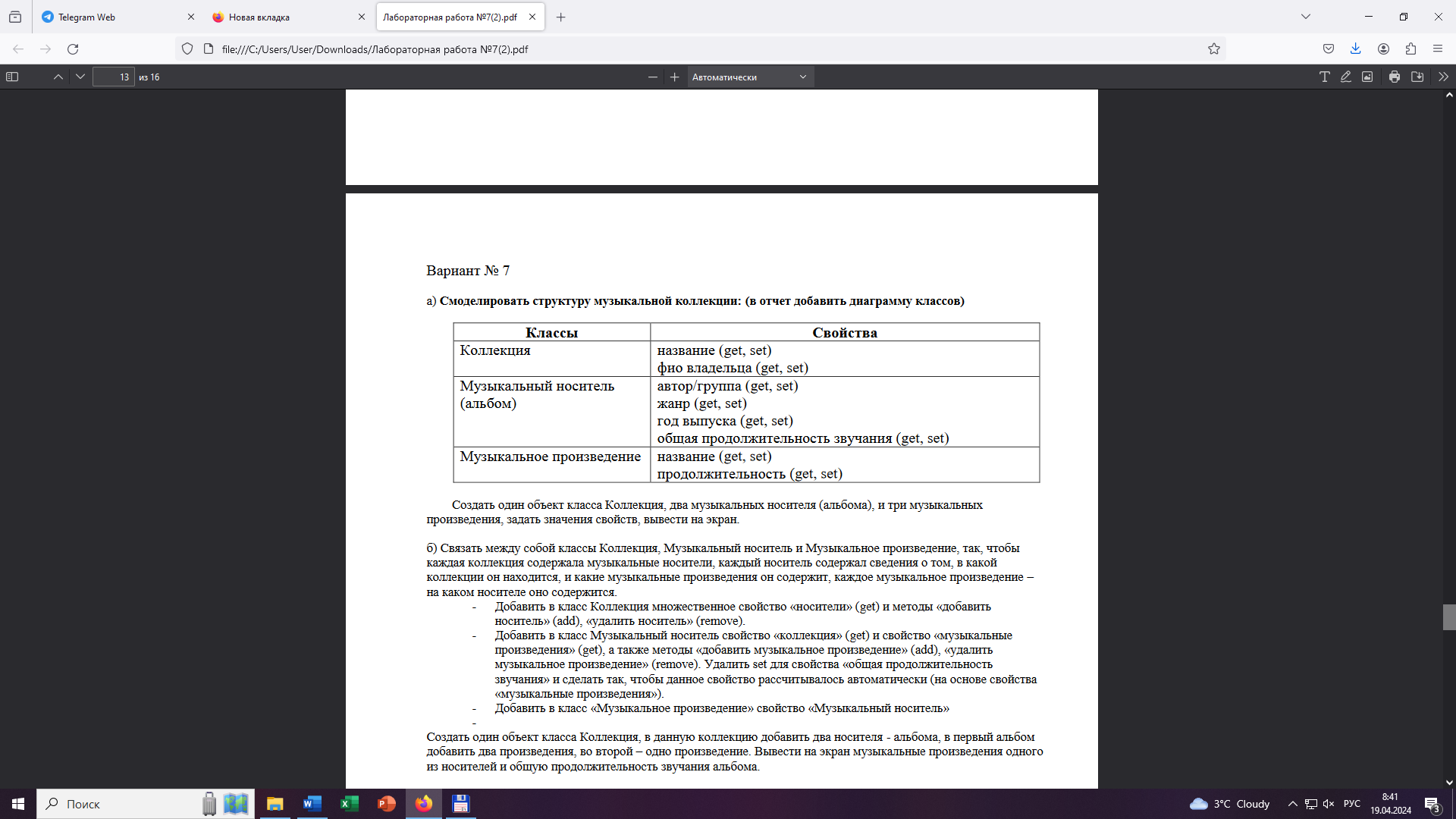
• основы UML;

• объектно-ориентированное программирование;

• основы Java.

**Вариант № 7**

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



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.

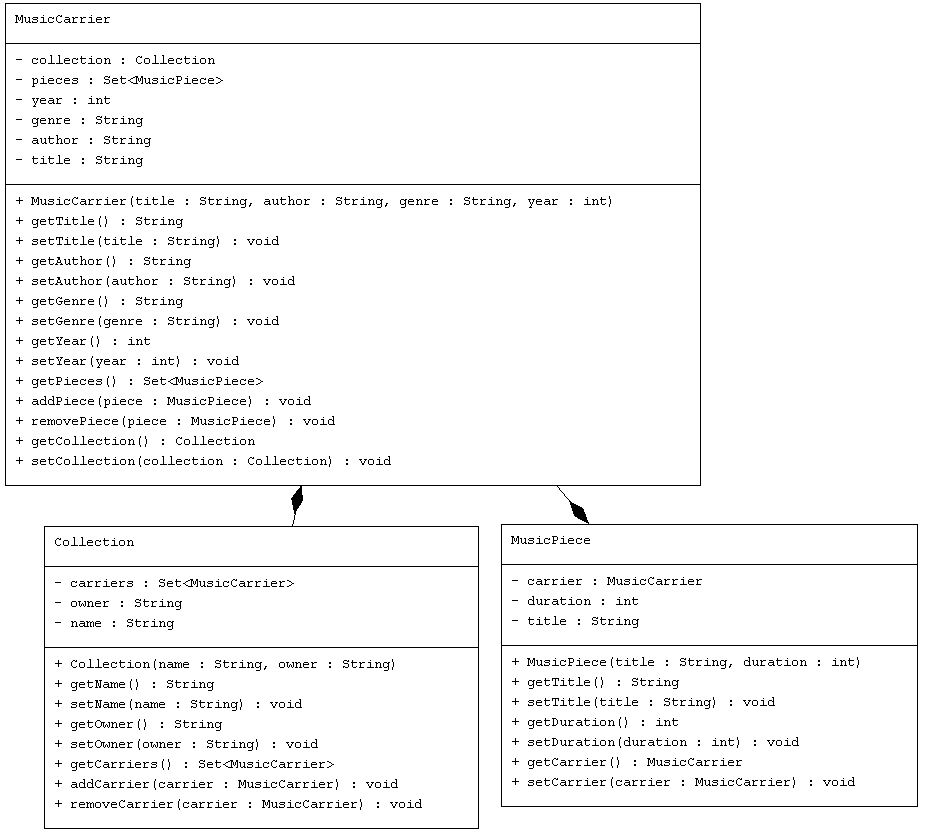


Рисунок 1-Диаграмма классов музыкальной коллекции

**Реализация:**

**Main.java:**

import java.util.Scanner;  
  
public class Main {  
 public static void main(String[] args) {  
 Collection collection = new Collection("My Collection", "John Doe");  
 MusicCarrier album1 = new MusicCarrier("Album 1", "Виктор Цой", "Genre 1", 2000);  
 MusicCarrier album2 = new MusicCarrier("Album 2", "Скриптонит", "Genre 2", 2010);  
  
 MusicPiece piece1 = new MusicPiece("Это любовь", 180);  
 MusicPiece piece2 = new MusicPiece("Батарейка", 200);  
 MusicPiece piece3 = new MusicPiece("Пачка сигарет", 220);  
  
 album1.addPiece(piece1);  
 album1.addPiece(piece2);  
 album2.addPiece(piece3);  
  
 collection.addCarrier(album1);  
 collection.addCarrier(album2);  
  
 // Вывод информации о коллекции, альбомах и произведениях  
  
 Scanner scanner = new Scanner(System.in);  
 System.out.print("Введите название музыкального произведения для поиска: ");  
 String searchTitle = scanner.nextLine();  
  
 MusicPiece foundPiece = collection.findMusicPieceByTitle(searchTitle);  
 if (foundPiece != null) {  
 System.out.println("Найдено музыкальное произведение:");  
 System.out.println("Название: " + foundPiece.getTitle());  
 System.out.println("Длительность: " + foundPiece.getDuration() + " секунд");  
 } else {  
 System.out.println("Музыкальное произведение с названием '" + searchTitle + "' не найдено.");  
 }  
 }

**MusicPiece.java**

public class MusicPiece {  
 private String title;  
 private int duration;  
 private MusicCarrier carrier;  
  
 public MusicPiece(String title, int duration) {  
 this.title = title;  
 this.duration = duration;  
 }  
  
 public String getTitle() {  
 return title;  
 }  
  
 public void setTitle(String title) {  
 this.title = title;  
 }  
  
 public int getDuration() {  
 return duration;  
 }  
  
 public void setDuration(int duration) {  
 this.duration = duration;  
 }  
  
 public MusicCarrier getCarrier() {  
 return carrier;  
 }  
 public void setCarrier(MusicCarrier carrier) {  
 this.carrier = carrier;  
 }  
}

**Collection.java:**

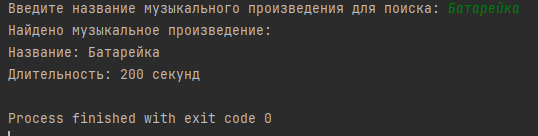
import java.util.Set;  
import java.util.HashSet;  
public class Collection {  
 private String name;  
 private String owner;  
 private Set<MusicCarrier> carriers;  
  
 public Collection(String name, String owner) {  
 this.name = name;  
 this.owner = owner;  
 this.carriers = new HashSet<>();  
 }  
  
 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 Set<MusicCarrier> getCarriers() {  
 return carriers;  
 }  
 public void addCarrier(MusicCarrier carrier) {  
 carriers.add(carrier);  
 }  
  
 public void removeCarrier(MusicCarrier carrier) {  
 carriers.remove(carrier);  
 }  
}

**MusicCarrier.java:**

import java.util.Set;  
import java.util.HashSet;  
public class MusicCarrier {  
 private String title;  
 private String author;  
 private String genre;  
 private int year;  
 private Set<MusicPiece> pieces;  
 private Collection collection;  
  
 public MusicCarrier(String title, String author, String genre, int year) {  
 this.title = title;  
 this.author = author;  
 this.genre = genre;  
 this.year = year;  
 this.pieces = new HashSet<>();  
 }  
  
 public String getTitle() {  
 return title;  
 }  
 public void setTitle(String title) {  
 this.title = title;  
 }  
 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(int year) {  
 this.year = year;  
 }  
 public Set<MusicPiece> getPieces() {  
 return pieces;  
 }  
 public void addPiece(MusicPiece piece) {  
 pieces.add(piece);  
 }  
 public void removePiece(MusicPiece piece) {  
 pieces.remove(piece);  
 }  
  
 public Collection getCollection() {  
 return collection;  
 }  
  
 public void setCollection(Collection collection) {  
 this.collection = collection;  
 }

public MusicPiece findMusicPieceByTitle(String title) {  
 for (MusicCarrier carrier : carriers) {  
 for (MusicPiece piece : carrier.getPieces()) {  
 if (piece.getTitle().equals(title)) {  
 return piece;  
 }  
 }  
 }  
 return null;   
 }  
}  
}

**Результат выполнения программы**



**Вывод:** получили основные понятия и навыки по созданию классов, описанию свойств и работы с объектами. Научились создавать диаграммы UML через плагины в intellij IDEA.