**Homework 4**

**Github:** <https://github.com/Dubs2310/CS5800-Homework-4.git>

**Flyweight**

* CharacterProperties.java

public class CharacterProperties {  
 private String font;  
 private String color;  
 private int size;  
  
 public CharacterProperties(String font, String color, int size) {  
 this.font = font;  
 this.color = color;  
 this.size = size;  
 }  
  
 public String getFont() { return font; }  
 public String getColor() { return color; }  
 public int getSize() { return size; }  
}

* CharacterPropertiesFactory.java

import java.util.HashMap;  
  
public class CharacterPropertiesFactory {  
 private static HashMap<String, CharacterProperties> *characterPropertiesMap* = new HashMap<>();  
  
 public static CharacterProperties getCharacterProperties(String font, String color, int size) {  
 String key = font + color + size;  
 CharacterProperties properties;  
 if (*characterPropertiesMap*.containsKey(key))  
 properties = *characterPropertiesMap*.get(key);  
 else {  
 properties = new CharacterProperties(font, color, size);  
 *characterPropertiesMap*.put(key, properties);  
 }  
 return properties;  
 }  
}

* Character.java

public class Character {  
 private char value;  
 private CharacterProperties properties;  
  
 public Character(char value, CharacterProperties properties) {  
 this.value = value;  
 this.properties = properties;  
 }  
  
 @Override  
 public String toString() {  
 return value + ", " + properties.getFont() + ", " + properties.getColor() + ", " + properties.getSize() + '\n';  
 }  
}

* Document.java

import java.io.File;  
import java.io.FileNotFoundException;  
import java.io.FileWriter;  
import java.io.IOException;  
import java.util.ArrayList;  
import java.util.Scanner;  
  
public class Document {  
 private ArrayList<Character> characters = new ArrayList<>();  
  
 public void addCharacter(char value, String font, String color, int size) {  
 CharacterProperties properties = CharacterPropertiesFactory.*getCharacterProperties*(font, color, size);  
 Character character = new Character(value, properties);  
 characters.add(character);  
 }  
  
 public ArrayList<Character> getCharacters() { return characters; }  
  
 public void saveToFile(String filename) {  
 try {  
 File file = new File(filename);  
 FileWriter writer = new FileWriter(file);  
 for (Character character: characters)  
 writer.write(character.toString());  
 writer.close();  
 } catch (IOException e) {  
 System.*out*.println("Error writing to " + filename);  
 e.printStackTrace();  
 }  
 }  
  
 public void loadFromFile(String filename) {  
 try {  
 File file = new File(filename);  
 Scanner scanner = new Scanner(file);  
 while (scanner.hasNextLine()) {  
 String[] characterProps = scanner.nextLine().split(", ");  
 char value = characterProps[0].charAt(0);  
 String font = characterProps[1];  
 String color = characterProps[2];  
 int size = Integer.*parseInt*(characterProps[3]);  
 addCharacter(value, font, color, size);  
 }  
 scanner.close();  
 } catch (FileNotFoundException e) {  
 System.*out*.println(filename + " not found");  
 e.printStackTrace();  
 }  
 }  
}

* Main.java

public class Main {  
 public static void main(String[] args) {  
 Document document = new Document();  
 document.addCharacter('H', "Arial", "Black", 12);  
 document.addCharacter('e', "Calibri", "Blue", 14);  
 document.addCharacter('l', "Verdana", "Red", 16);  
 document.addCharacter('l', "Calibri", "Red", 12);  
  
 document.addCharacter('o', "Arial", "Black", 12);  
 document.addCharacter('W', "Calibri", "Blue", 14);  
 document.addCharacter('o', "Verdana", "Red", 16);  
 document.addCharacter('r', "Calibri", "Red", 12);  
  
 document.addCharacter('l', "Arial", "Black", 12);  
 document.addCharacter('d', "Calibri", "Blue", 14);  
 document.addCharacter('C', "Verdana", "Red", 16);  
 document.addCharacter('S', "Calibri", "Red", 12);  
  
 document.addCharacter('5', "Arial", "Black", 12);  
 document.addCharacter('8', "Calibri", "Blue", 14);  
 document.addCharacter('0', "Verdana", "Red", 16);  
 document.addCharacter('0', "Calibri", "Red", 12);  
   
 document.saveToFile("HelloWorldCS5800.txt");  
   
 Document documentCopy = new Document();  
 documentCopy.loadFromFile("HelloWorldCS5800.txt");  
   
 for (Character character : documentCopy.getCharacters()) {  
 System.*out*.print(character.toString());  
 }  
 }  
}

* Output (load from file)

**A screenshot of a computer

Description automatically generated**

* Output (save to file)

**A screenshot of a computer

Description automatically generated with medium confidence**

**Proxy**

* Song.java

public class Song {  
 private Integer id;  
 private String title;  
 private String artist;  
 private String album;  
 private int duration;  
  
 public Song(Integer id, String title, String artist, String album, int duration) {  
 this.id = id;  
 this.title = title;  
 this.artist = artist;  
 this.album = album;  
 this.duration = duration;  
 }  
  
 public Integer getId() { return id; }  
 public String getTitle() { return title; }  
 public String getArtist() { return artist; }  
 public String getAlbum() { return album; }  
 public int getDuration() { return duration; }  
}

* SongService.java

import java.util.List;  
  
public interface SongService {  
 public Song searchById(Integer songID);  
 public List<Song> searchByTitle(String title);  
 public List<Song> searchByAlbum(String album);  
}

* MusicStreamingService.java

import java.util.ArrayList;  
import java.util.List;  
  
public class MusicStreamingService implements SongService {  
 private List<Song> library;  
  
 public MusicStreamingService(List<Song> library) {  
 this.library = library;  
 }  
  
 private void timeDelay() {  
 try { Thread.*sleep*(3000); }  
 catch (Exception e) {}  
 }  
  
 @Override  
 public Song searchById(Integer songID) {  
 timeDelay();  
 Song searchResult = null;  
 for (Song song: library)  
 if (song.getId().equals(songID))  
 searchResult = song;  
 return searchResult;  
 }  
  
 @Override  
 public List<Song> searchByTitle(String title) {  
 timeDelay();  
 List<Song> searchResults = new ArrayList<>();  
 for (Song song: library)  
 if (song.getTitle().toLowerCase().contains(title.toLowerCase()))  
 searchResults.add(song);  
 return searchResults;  
 }  
  
 @Override  
 public List<Song> searchByAlbum(String album) {  
 timeDelay();  
 List<Song> searchResults = new ArrayList<>();  
 for (Song song: library)  
 if (song.getTitle().toLowerCase().contains(album.toLowerCase()))  
 searchResults.add(song);  
 return searchResults;  
 }  
}

* ProxyService.java

import java.util.HashMap;  
import java.util.List;  
import java.util.Map;  
  
public class ProxyService implements SongService {  
 private SongService streamingService;  
 private Map<Integer, Song> songCache = new HashMap<>();  
 private Map<String, List<Song>> titleSearchCache = new HashMap<>();  
 private Map<String, List<Song>> albumSearchCache = new HashMap<>();  
  
 public ProxyService(SongService streamingService) {  
 this.streamingService = streamingService;  
 }  
  
 // Check song cache if songID is present, otherwise retrieve song from the streaming service  
 @Override  
 public Song searchById(Integer songID) {  
 Song searchResult;  
 if (songCache.containsKey(songID))  
 searchResult = songCache.get(songID);  
 else {  
 searchResult = streamingService.searchById(songID);  
 songCache.put(songID, searchResult);  
 }  
 return searchResult;  
 }  
  
 private void putSongsInCache(List<Song> songs) {  
 if (songs.size() == 0) return;  
 for (Song song: songs) {  
 Integer songID = song.getId();  
 if (!songCache.containsKey(songID))  
 songCache.put(songID, song);  
 }  
 }  
  
 // Check titleSearchCache if search was done before, else get list from streaming service  
 // Add songs to songCache for future use  
 @Override  
 public List<Song> searchByTitle(String title) {  
 List<Song> searchResults;  
 if (titleSearchCache.containsKey(title))  
 searchResults = titleSearchCache.get(title);  
 else {  
 searchResults = streamingService.searchByTitle(title);  
 titleSearchCache.put(title, searchResults);  
 }  
 putSongsInCache(searchResults);  
 return searchResults;  
 }  
  
 // Check albumSearchCache if search was done before, else get list from streaming service  
 // Add songs to songCache for future use  
 @Override  
 public List<Song> searchByAlbum(String album) {  
 List<Song> searchResults;  
 if (albumSearchCache.containsKey(album))  
 searchResults = albumSearchCache.get(album);  
 else {  
 searchResults = streamingService.searchByAlbum(album);  
 albumSearchCache.put(album, searchResults);  
 }  
 putSongsInCache(searchResults);  
 return searchResults;  
 }  
}

* Main.java

import java.util.ArrayList;  
import java.util.List;  
  
public class Main {  
 public static void printSearchResult(Song searchResult) {  
 System.*out*.println("Title: " + searchResult.getTitle());  
 System.*out*.println("Artist: " + searchResult.getArtist());  
 System.*out*.println("Album: " + searchResult.getAlbum());  
 System.*out*.println("Duration: " + searchResult.getDuration());  
 System.*out*.println();  
 }  
  
 public static void printSearchResults(List<Song> searchResults) {  
 for (Song searchResult: searchResults)  
 *printSearchResult*(searchResult);  
 }  
  
 public static void main(String[] args) {  
 List<Song> songLibrary = new ArrayList<>();  
 songLibrary.add(new Song(0, "In the End", "Linkin Park", "Hybrid Theory", 218));  
 songLibrary.add(new Song(1, "Numb", "Linkin Park", "Meteora", 186));  
 songLibrary.add(new Song(2, "Lose Yourself", "Eminem", "8 Mile Soundtrack", 326));  
 songLibrary.add(new Song(3, "Without Me", "Eminem", "The Eminem Show", 289));  
 songLibrary.add(new Song(4, "Poker Face", "Lady Gaga", "The Fame", 216));  
 songLibrary.add(new Song(5, "Bad Romance", "Lady Gaga", "The Fame Monster", 295));  
  
 SongService streamingService = new MusicStreamingService(songLibrary);  
 streamingService = new ProxyService(streamingService);  
  
 Song searchResult = streamingService.searchById(4);  
 System.*out*.println("------------------------------");  
 System.*out*.println("Search By ID: (4)");  
 *printSearchResult*(searchResult);  
  
 searchResult = streamingService.searchById(4);  
 System.*out*.println("------------------------------");  
 System.*out*.println("Search By ID: (4)");  
 *printSearchResult*(searchResult);  
  
 List<Song> searchResults = streamingService.searchByTitle("th");  
 System.*out*.println("------------------------------");  
 System.*out*.println("Search By Title: (th)");  
 *printSearchResults*(searchResults);  
  
 searchResults = streamingService.searchByTitle("th");  
 System.*out*.println("------------------------------");  
 System.*out*.println("Search By Title: (th)");  
 *printSearchResults*(searchResults);  
   
 searchResults = streamingService.searchByAlbum("i");  
 System.*out*.println("------------------------------");  
 System.*out*.println("Search By Album: (i)");  
 *printSearchResults*(searchResults);  
  
 searchResults = streamingService.searchByAlbum("i");  
 System.*out*.println("------------------------------");  
 System.*out*.println("Search By Album: (i)");  
 *printSearchResults*(searchResults);  
 }  
}

* Output

A screenshot of a computer

Description automatically generated with medium confidence A screenshot of a computer

Description automatically generated A screenshot of a computer

Description automatically generated with medium confidence