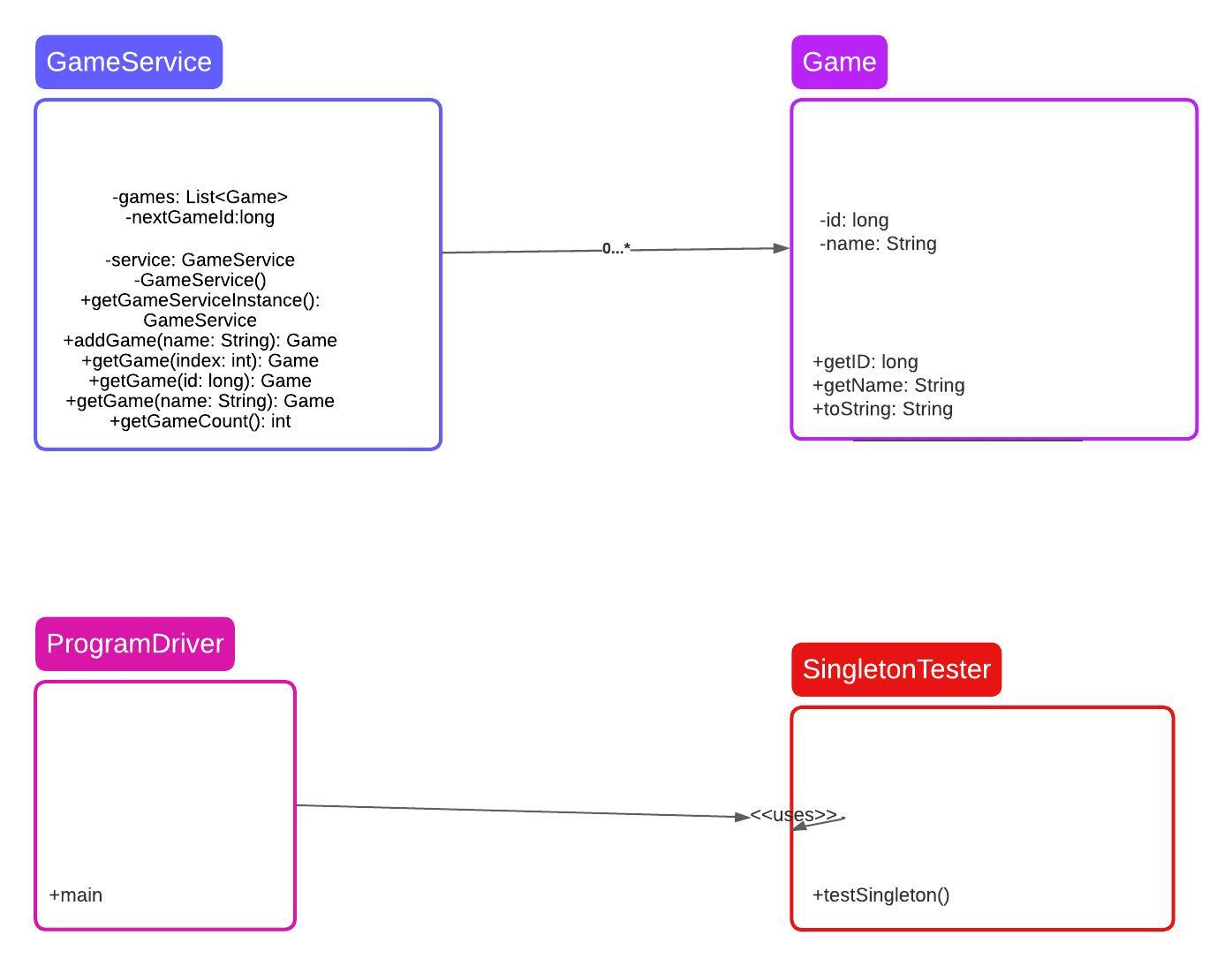
MK Fowlkes

[Paul.fowlkes@snhu.edu](mailto:Paul.fowlkes@snhu.edu)

CS 230 2-2



import java.util.ArrayList;

import java.util.List;

public class GameService {

private static GameService instance; // Singleton instance

private List<Game> games; // List of games

private long nextGameId; // Next game ID

private GameService() {

games = new ArrayList<>();

nextGameId = 1;

}

// Singleton pattern implementation

public static GameService getInstance() {

if (instance == null) {

instance = new GameService();

}

return instance;

}

// Add a new game

public Game addGame(String name) {

// Use an iterator pattern to check for duplicate names

for (Game game : games) {

if (game.getName().equals(name)) {

throw new IllegalArgumentException("Game with the same name already exists.");

}

}

Game newGame = new Game(nextGameId, name);

games.add(newGame);

nextGameId++;

return newGame;

}

// Get a game by index

public Game getGame(int index) {

if (index < 0 || index >= games.size()) {

throw new IndexOutOfBoundsException("Index is out of bounds.");

}

return games.get(index);

}

// Get a game by name using an iterator pattern

public Game getGame(String name) {

for (Game game : games) {

if (game.getName().equals(name)) {

return game;

}

}

throw new IllegalArgumentException("Game with the specified name not found.");

}

// Get the count of games

public int getGameCount() {

return games.size();

}

}

- The `GameService` class is designed as a Singleton to ensure only one instance of the class exists at any time. The `getInstance` method provides access to the Singleton instance.

- The `addGame` method uses the Iterator pattern to check for duplicate names. It iterates through the list of games and throws an exception if a game with the same name already exists.

- The `getGame(String name)` method allows you to retrieve a game by its unique name.

- The comments in the code explain the purpose and characteristics of the Singleton and Iterator patterns.