MODULE TWO– Project One Milestone

KEYNEISHA D. MCNEALEY

PROFESSOR HECKER

07/09/2024

GameService Class

+ GameService

- games: List<Game>

- nextGameId: long

- instance: GameService (static)

+ getInstance(): GameService (static)

+ addGame(name: String): Game

+ getGame(index: int): Game

+ getGameCount(): int

+ private GameService() (private constructor)

Game Class

+ Game

- id: long

- name: String

+ getId(): long

+ getName(): String

+ toString(): String

ProgramDriver Class

+ ProgramDriver

+ main()

SingletonTester Class

+ SingletonTester

+ testSingleton()

Relationships

A straight horizontal line connects the GameService class to the Game class, denoted with 0...*.

A straight horizontal line with a closed arrow points from the ProgramDriver class to the SingletonTester class, denoted as <<uses>>.

Explanation

The GameService class is implemented as a Singleton, ensuring that only one instance of the class is created. The instance attribute is a static variable that holds the single instance of the class. The getInstance() method is a static method that returns the single instance of the class, creating it if it does not exist. The private constructor private GameService() ensures that the class cannot be instantiated from outside. The Game class remains unchanged, representing a game with an id and a name.

The ProgramDriver class and SingletonTester class are also unchanged, with the ProgramDriver class having a main() method and the SingletonTester class having a testSingleton() method.

The necessary static and instance attributes and methods have been implemented to adhere to the Singleton pattern.