

GameStore

Instruction:

- Develop a **GameStore** class that facilitates managing a video game inventory. This includes capabilities for adding games, searching within the inventory by game title or genre, and allowing customers to purchase games, which then removes them from the inventory.

Class Game::

Constructor

- **constructor(id, title, genre, price):**
 - **Parameters:** **id** type: number, **title** type: string, **genre** type: string, **price** type: number.
 - Initializes a new game instance with the specified id, title, genre, and price.

Properties

- **id** : A unique identifier for the game, type: Number.
- **title** : The title of the game, type: String.
- **genre** : The genre of the game (e.g., Action, Adventure, Puzzle) , type: String.
- **price** : The selling price of the game, type: Number.

Class GameStore::

Constructor

- **constructor(storeName):**
 - **Parameters:** **storeName** type: string
 - Initializes a new instance of GameStore with a specified name and an **empty inventory**.

Properties

- **storeName** : The name of the game store, type: string.
- **inventory** : A collection of games available for sale, type: Array of Game objects.

Methods

- **addGame(game: Game):**
 - **Parameters:** **game** type: game object
 - Adds a new **Game** object to the inventory. Ensure the game has valid attributes before adding.

- **searchByTitle(title):**
 - **Parameters:** **title** type: string
 - Searches the inventory for games matching a specific title and **returns** array of all matching games.
- **searchByGenre(genre):**
 - **Parameters:** **genre** type: string
 - Finds and **returns** array of all games within a specified genre in the inventory.
- **purchaseGame(id)**
 - **Parameters:** **id** type: number
 - Removes a game from the inventory by its ID and **returns** the game object. Validate the existence of the game before removal.
- **listInventory():**
 - **Parameters:** -
 - Lists all games currently in the inventory.
- **updateGamePrice(id, newPrice):**
 - **Parameters:** **id** type: number, **newPrice** type: number,
 - Updates the price of a game based on its id.
- **totalInventoryValue():**
 - **Parameters:** -
 - Calculates and **returns** the total value of all games in the inventory.
- **listGenres():**
 - **Parameters:** -
 - Lists all unique genres available in the inventory.

*For Module 1.1

- **sellGame(id, quantity)**
 - **Parameters:** **id** type: number, **quantity** type: number
 - Sells a game identified by its ID in the specified quantity. Decreases the game stock by the quantity sold. **Returns** game object if the game exists and there was enough stock to sell, undefined otherwise.

```
class Game {  
}  
  
class GameStore {  
}  
  
const store = new GameStore("Epic Game Store");  
  
// Adding games to the inventory  
store.addGame(new Game(1, "Game One", "Action", 59.99));  
store.addGame(new Game(2, "Game Two", "Adventure", 49.99));  
  
// Searching games by title  
console.log(store.searchByTitle("One"));  
  
// Searching games by genre  
console.log(store.searchByGenre("Adventure"));  
  
// Purchasing a game  
console.log(store.purchaseGame(1));  
  
// Listing the inventory  
console.log(store.listInventory());  
  
// Updating a game price  
store.updateGamePrice(2, 39.99);  
  
// Calculating total inventory value  
console.log(store.totalInventoryValue());  
  
// Listing all genres  
console.log(store.listGenres());  
  
// Selling a game (assuming the game with ID 2 had its stock increased appropriately before)  
console.log(store.sellGame(2, 1));
```