**JS Advanced: Exam 15 July 2018**

**Problem 3. Book Collection**

Write a **JavaScript class** **BookCollection** which holds a list containing shelf information **(shelfGenre, room, shelfCapacity).**

|  |
| --- |
| **class** BookCollection {  *//* ***TODO: implement this class*** } |

Each **BookCollection** is located in specific room, on a shelf with defined capacity and shelf name. Implement the following features:

* **Constructor** – It should contain the following properties – **room**(String), **shelfGenre**(String), **shelf**(an array), **shelfCapacity**(Number). If the room is: "**livingRoom**" or "**bedRoom**" or "**closet**", create the shelf’s genre, room and shelf capacity. If it **is** **not,** throw "Cannot have book shelf in {room's name}". Shelf capacity will always be a valid positive number.
* Method **addBook(bookName,** **bookAuthor, genre)** – adds book to the shelf only if there’s enough space in the shelf. If the shelf is full, remove the **first** book to make space for the **new** one. **The genre is optional**. In the end, **sort** our shelf **alphabetically** by **book author’s name**.
* Method **throwAwayBook(bookName) – removes** a book from the shelf by the given name.
* Method **showBooks(genre) –** returns all books by the given genre. You should return a string with the following information:

|  |
| --- |
| **“Results for search "{history}":”**  **“\uD83D\uDCD6 {bookAuthor} – "{bookName}"”**  **…** |

* Accessor property **shelfCondition** – returns the **count** of **free slots** left in the shelf.
* Method **toString()** – returns the **text** **representation** of the shelf in the following format:
* Empty shelf:

|  |
| --- |
| “**It's an empty shelf**” |

* Non-empty shelf:

|  |
| --- |
| **“"{shelfGenre}" shelf in {room} contains:”**  **“\uD83D\uDCD6 "{bookName}" – {bookAuthor}”**  **…** |

**Examples**

This is an example of how the **BookCollection** class is **intended to be used**:

|  |
| --- |
| **Sample code usage** |
| **let *livingRoom*** = **new** BookCollection(**"Programming"**, **"livingRoom"**, 5)  .addBook(**"Introduction to Programming with C#"**, **"Svetlin Nakov"**)  .addBook(**"Introduction to Programming with Java"**, **"Svetlin Nakov"**)  .addBook(**"Programming for .NET Framework"**, **"Svetlin Nakov"**); ***console***.log(***livingRoom***.toString()); |

|  |
| --- |
| **Corresponding output** |
| "Programming" shelf in livingRoom contains:  📖 "Introduction to Programming with C#" - Svetlin Nakov  📖 "Introduction to Programming with Java" - Svetlin Nakov  📖 "Programming for .NET Framework" - Svetlin Nakov |

|  |
| --- |
| **Sample code usage** |
| **let *garden*** = **new** BookCollection(**"Programming"**, **"garden"**); |
| **Corresponding output** |
| "Cannot have book shelf in garden" |

|  |
| --- |
| **Sample code usage** |
| **let *bedRoom*** = **new** BookCollection(**'Mixed'**, **'bedRoom'**, 5); ***bedRoom***.addBook(**"John Adams"**, **"David McCullough"**, **"history"**); ***bedRoom***.addBook(**"The Guns of August"**, **"Cuentos para pensar"**, **"history"**); ***bedRoom***.addBook(**"Atlas of Remote Islands"**, **"Judith Schalansky"**); ***bedRoom***.addBook(**"Paddle-to-the-Sea"**, **"Holling Clancy Holling"**); ***console***.log(**"Shelf's capacity: "** + ***bedRoom***.shelfCondition); ***console***.log(***bedRoom***.showBooks(**"history"**)); |
| **Corresponding output** |
| Shelf's capacity: 1  Results for search "history":  📖 Cuentos para pensar - "The Guns of August"  📖 David McCullough - "John Adams" |

**Submission**

Submit your class **BookCollection** as “**JavaScript code**”.

**Notes**

Use the following Unicode for visualizing the book icon: **"\uD83D\uDCD6".**

**No invalid input will be given.**