Movie Rating System

Array Methods - Extra

1. **Group Movies by Genre**

   - **Input:** An array of movie objects, where each object has at least a `title` and `genre` property.
   - **Task:** Write a function that groups the movies by their genres. The output should be an object where the keys are the unique genres, and the values are arrays of the movies belonging to that genre.
   - **Example Input:**
     ```
     [{title: "Inception", genre: "Action"}, {title: "The Dark Knight", genre: "Action"}, {title: "A Beautiful Mind", genre: "Drama"}]
     ```
   - **Example Output:**
     ```
     {Action: [{title: "Inception", genre: "Action"}, {title: "The Dark Knight", genre: "Action"}], Drama: [{title: "A Beautiful Mind", genre: "Drama"}]}
     ```

2. **Find Top Rated Movie for Each Genre**

   - **Input:** An array of movie objects, where each object has at least a `title`, `genre`, and `rating` property.
   - **Task:** Write a function that finds the top-rated movie for each genre. The output should be an object where keys are the genres and the values are objects representing the highest-rated movie within that genre.
   - **Example Input:**
     ```
     [{title: "Inception", genre: "Action", rating: 8.8}, {title: "The Dark Knight", genre: "Action", rating: 9.0}, {title: "A Beautiful Mind", genre: "Drama", rating: 8.2}]
     ```
   - **Example Output:**
     ```
     {Action: {title: "The Dark Knight", genre: "Action", rating: 9.0}, Drama: {title: "A Beautiful Mind", genre: "Drama", rating: 8.2}}
     ```

3. **Count Movies by Genre**

   - **Input:** An array of movie objects, where each object has at least a `title` and `genre` property.
   - **Task:** Write a function that counts the number of movies within each genre. The output should be an object where keys are the genres and values are the number of movies within that genre.
   - **Example Input:** `[{title: "Inception", genre: "Action"}, {title: "The Dark Knight", genre: "Action"}, {title: "A Beautiful Mind", genre: "Drama"}]`
   - **Example Output:** `{Action: 2, Drama: 1}`

4. **Genre Presence Across Years**

   - **Input:** An array of movie objects, where each object has at least a `title`, `genre`, and `year` property, and a genre string.
   - **Task:** Write a function that counts the number of movies of a specified genre released each year. The output should be an object where keys are the years and values are the number of movies of the specified genre released that year.
   - **Example Input:** `[{title: "Inception", genre: "Action", year: 2010}, {title: "The Dark Knight", genre: "Action", year: 2008}, {title: "A Beautiful Mind", genre: "Drama", year: 2001}], "Action"`
   - **Example Output:** `{2010: 1, 2008: 1}`

5. **Movies with Ratings within Range**

   - **Input:** An array of movie objects, where each object has at least a `title` and `rating` property, and two numbers representing the lower and upper bounds of a rating range.
   - **Task:** Write a function that returns an array of movie titles that have a rating within the provided range (inclusive).
   - **Example Input:** `[{title: "Inception", rating: 8.8}, {title: "The Dark Knight", rating: 9.0}, {title: "A Beautiful Mind", rating: 8.2}], 8.5, 9.0`
   - **Example Output:** `["Inception", "The Dark Knight"]`

6. **Find All Unique Genres**

   - **Input:** An array of movie objects, where each object has at least a `genre` property.

- **Task:** Write a function that returns a new array with all unique genres present in the input movies.
- **Example Input:**
  ```
  [{title: "Inception", genre: "Action"}, {title: "The Dark Knight", genre: "Action"}, {title: "A Beautiful Mind", genre: "Drama"}]
  ```
- **Example Output:** `["Action", "Drama"]`

7. **Deep Copy of Movies**

   - **Input:** An array of movie objects.
   - **Task:** Write a function that returns a new deep copied array of movie objects. Changing properties in the new array should not affect the original array.
   - **Example Input:**
     ```
     [{title: "Inception", genre: "Action"}, {title: "The Dark Knight", genre: "Action"}]
     ```
   - **Example Output:**
     ```
     [{title: "Inception", genre: "Action"}, {title: "The Dark Knight", genre: "Action"}]
     ```

8. **Sort Movies by Rating, Then Votes**

   - **Input:** An array of movie objects, where each object has at least a `title`, `rating`, and `votes` property.
   - **Task:** Write a function that returns a new array sorted first by rating (descending), then by votes (

descending) in case of a tie. - **Example Input:**

```
[{title: "Inception", rating: 8.8, votes: 2000}, {title: "The Dark Knight", rating: 9.0, votes: 1500}, {title: "A Beautiful Mind", rating: 8.8, votes: 3000}]
```

- **Example Output:**

```
[{title: "The Dark Knight", rating: 9.0, votes: 1500}, {title: "A Beautiful Mind", rating: 8.8, votes: 3000}, {title: "Inception", rating: 8.8, votes: 2000}]
```

9. **Normalize Ratings**

   - **Input:** An array of movie objects, where each object has at least a `title` and `rating` property.

   - **Task:** Write a function that returns a new array of movie objects where the rating of each movie is normalized to a 0-100 scale (0 being the lowest rating in the input and 100 being the highest).
   - **Example Input:** `[{title: "Inception", rating: 8.8}, {title: "The Dark Knight", rating: 9.0}, {title: "A Beautiful Mind", rating: 8.2}]`
   - **Example Output:** `[{title: "Inception", rating: 75.0}, {title: "The Dark Knight", rating: 100.0}, {title: "A Beautiful Mind", rating: 0.0}]`

10. **Top Director**

   - **Input:** An array of movie objects, where each object has at least a `title`, `rating`, and `director` property.
   - **Task:** Write a function that finds the director who has the highest average movie rating. The output should be an object containing the director's name and their average rating.
   - **Example Input:** `[{title: "Inception", director: "Christopher Nolan", rating: 8.8}, {title: "The Dark Knight", director: "Christopher Nolan", rating: 9.0}, {title: "A Beautiful Mind", director: "Ron Howard", rating: 8.2}]`
   - **Example Output:** `{director: "Christopher Nolan", averageRating: 8.9}`