Music Player

Array Methods - Extra

1. **Group Songs by Album**
    - **Input:** An array of song objects, where each object has at least a `title` and `album` property.
    - **Task:** Write a function that groups the songs by their albums. The output should be an object where the keys are the unique albums, and the values are arrays of the songs belonging to that album.
    - **Example Input:**
      ```
      [{title: "Yellow", album: "Parachutes"}, {title: "Clocks", album: "A Rush of Blood to the Head"}, {title: "The Scientist", album: "A Rush of Blood to the Head"}]
      ```
    - **Example Output:**
      ```
      {Parachutes: [{title: "Yellow", album: "Parachutes"}], "A Rush of Blood to the Head": [{title: "Clocks", album: "A Rush of Blood to the Head"}, {title: "The Scientist", album: "A Rush of Blood to the Head"}]}
      ```
2. **Filter Songs by Genre**
    - **Input:** An array of song objects, where each object has at least a `title` and `genre` property, and a genre string.
    - **Task:** Write a function that returns an array of songs that belong to a specific genre.
    - **Example Input:**
      ```
      [{title: "Yellow", genre: "Rock"}, {title: "Clocks", genre: "Rock"}, {title: "Smooth", genre: "Latin"}], "Rock"
      ```
    - **Example Output:**
      ```
      [{title: "Yellow", genre: "Rock"}, {title: "Clocks", genre: "Rock"}]
      ```
3. **Find the Longest Song**
    - **Input:** An array of song objects, where each object has at least a `title` and `duration` property.
    - **Task:** Write a function that returns the song object with the longest duration.

- **Example Input:**
  `[{title: "Yellow", duration: 4.6}, {title: "Clocks", duration: 5.1}, {title: "Smooth", duration: 4.9}]`
- **Example Output:** `{title: "Clocks", duration: 5.1}`

4. **Average Song Length**

- **Input:** An array of song objects, where each object has at least a `title` and `duration` property.
- **Task:** Write a function that returns the average duration of all songs.
- **Example Input:**
  `[{title: "Yellow", duration: 4.6}, {title: "Clocks", duration: 5.1}, {title: "Smooth", duration: 4.9}]`
- **Example Output:** `4.866666666666667`

5. **Sort Songs by Popularity**

- **Input:** An array of song objects, where each object has at least a `title` and `popularity` property.
- **Task:** Write a function that returns a new array of songs sorted by popularity in descending order.
- **Example Input:**
  `[{title: "Yellow", popularity: 80}, {title: "Clocks", popularity: 90}, {title: "Smooth", popularity: 85}]`
- **Example Output:**
  `[{title: "Clocks", popularity: 90}, {title: "Smooth", popularity: 85}, {title: "Yellow", popularity: 80}]`

6. **Count Songs by Artist**

- **Input:** An array of song objects, where each object has at least a `title` and `artist` property.
- **Task:** Write a function that returns an object where keys are the artists and values are the number of songs by that artist.
- **Example Input:**
  `[{title: "Yellow", artist: "Coldplay"}, {title: "Clocks", artist: "Coldplay"}, {title: "Smooth", artist: "Santana"}]`
- **Example Output:** `{"Coldplay": 2, "Santana": 1}`

7. **Find Songs Released in a Specific Year**

- **Input:** An array of song objects, where each object has at least a `title` and `year` property, and a year number.
- **Task:** Write a function that returns an array of song titles released in a specific year.
- **Example Input:**
  ```
  [{title: "Yellow", year: 2000}, {title: "Clocks", year: 2002}, {title: "Smooth", year: 2000}], 2000
  ```
- **Example Output:** `["Yellow", "Smooth"]`

8. **Find the Most Popular Artist**

   - **Input:** An array of song objects, where each object has at least a `title`, `artist`, and `popularity` property.
   - **Task:** Write a function that finds the artist with the highest average song popularity. The output should be an object containing the artist's name and their average popularity.
   - **Example Input:**
     ```
     [{title: "Yellow", artist: "Coldplay", popularity: 80}, {title: "Clocks", artist: "Coldplay", popularity: 90}, {title: "Smooth", artist: "Santana", popularity: 85}]
     ```
   - **Example Output:** `{artist: "Coldplay", averagePopularity: 85}`

9. **Filter Explicit Songs**

   - **Input:** An array of song objects, where each object has at least a `title` and `explicit` boolean property.
   - **Task:** Write a function that returns an array of non-explicit songs (where `explicit` is `false`).
   - **Example Input:**
     ```
     [{title: "Yellow", explicit: false}, {title: "Clocks", explicit: false}, {title: "Smooth", explicit: true}]
     ```
   - **Example Output:**
     ```
     [{title: "Yellow", explicit: false}, {title: "Clocks", explicit: false}]
     ```

10. **Deep Copy of Songs**

    - **Input:** An array of song objects.
    - **Task:** Write a function that returns a new deep copied array of song objects. Changing properties in the new array should not affect the original array.

- **Example Input:**

  ```
  [{title: "Yellow", artist: "Coldplay"}, {title: "Clocks", artist: "Coldplay"}]
  ```

- **Example Output:**

  ```
  [{title: "Yellow", artist: "Coldplay"}, {title: "Clocks", artist: "Coldplay"}]
  ```