

Стандартна бібліотека

Необхідно розробити бібліотеку, що реалізує стандартні структури даних. Реалізувати функції-конструктори, заготовки яких задані в файлі `collection_lib.js`. Тести для завдання в файлі `tests.js`

Завдання 1. Колекція (Collection)

Колекція елементів. Підтримує операції вставки і вилучення першого і останнього елемента, зберігає посилання на перший і останній елемент. Методи та властивості колекції:

- `first` - перший елемент колекції
- `last` - останній елемент колекції
- `length` - кількість елементів в колекції
- `isEmpty` - `true`, якщо колекція порожня
- `pickFirst()` - витягує перший елемент з колекції і повертає його
- `pickLast()` - витягує останній елемент з колекції і повертає його
- `insertFirst()` - вставляє елемент в початок колекції
- `insertLast()` - вставляє елемент в кінець колекції
- `empty()` - очищає колекцію.

Завдання 2. Черга (Queue)

Список елементів, організованих за принципом "першим прийшов - першим вийшов". Елементи додаються і витягуються з одного кінця списку. Методи та властивості:

- `length` - кількість елементів в черзі
- `enqueue(item)` - додає елемент в чергу
- `dequeue()` - витягує елемент з черги
- `empty()` - очищає чергу.

Завдання 3. Масив фіксованої довжини (Fixed array)

Масив фіксованого розміру. Спроба додати або отримати елемент за межами зазначеного діапазону повинна викликати помилку `RangeError`. Методи та властивості:

- `length` - кількість елементів в масиві
- `insertAt(index, item)` - записує елемент в масив по заданому індексу
- `getAt(index)` - повертає елемент за вказаною індексу.

Завдання 4. Множина (Set)

Список, який зберігає унікальні елементи. Методи та властивості:

- `length` - кількість елементів у множині
- `insert(item)` - додає елемент до множини
- `remove(item)` - видаляє елемент з множини
- `has(item)` - перевіряє, чи входить елемент до множини
- `intersect(set)` - повертає множину елементів, які входять у вихідну множину та в передану множину (в обидві відразу, [перетин множин](#))
- `union(set)` - повертає множину елементів, які входять у вихідну множину або в передану множину (в будь-яку з двох, [об'єднання множин](#))
- `empty()` - очищає множину

Додаткове завдання. Не використовувати масив.

Реалізувати базову колекцію на основі [двозв'язаного списку](#).