Collections

Gesamtpunktzahl 27/27

✓ Какой из следующих интерфейсов наиболее подходящий для представления упорядоченного списка объектов? 1/1

- (●) A) List ✓
- ( ) B) Set
- ( ) C) Map
- ( ) D) Queue

Feedback

List представляет упорядоченную коллекцию, где элементы могут дублироваться.

✓ В какой коллекции порядок элементов гарантированно сохраняется? 1/1

- ( ) A) HashSet
- (●) B) LinkedHashSet ✓
- ( ) C) TreeSet
- ( ) D) HashMap

Feedback

В LinkedHashSet порядок элементов сохраняется в порядке их вставки.

✓ Какая коллекция автоматически сортирует элементы? 1/1

- A) HashSet
- B) LinkedHashSet
- C) TreeSet ✓
- D) LinkedList

Feedback

TreeSet автоматически сортирует элементы по естественному порядку или с использованием компаратора.

✓ В каком из классов допускаются ключи с null? 1/1

- A) TreeMap
- B) HashMap ✓
- C) LinkedHashMap
- D) HashSet

✓ Какая структура представляет последовательность элементов в порядке "первым пришел, первым ушел" (FIFO)? 1/1

- A) Stack
- B) Queue ✓
- C) List
- D) Set

Feedback

Очередь (Queue) представляет структуру данных в порядке FIFO.

✓ Какой класс представляет структуру данных "последний пришел, первым ушел" (LIFO)? 1/1

- A) Stack ✓
- B) Queue
- C) ArrayList
- D) TreeSet

Feedback

Stack представляет структуру данных в порядке LIFO.

✓ Какой из классов реализует двусвязный список? 1/1

- A) ArrayList
- B) LinkedList ✓
- C) HashSet
- D) TreeSet

✓ Что характерно для Set? 1/1

- A) Сохраняет порядок элементов
- B) Позволяет дублирование элементов
- C) Не допускает дублирование элементов ✓
- D) Использует ключ-значение

✓ Какой интерфейс использует пары "ключ-значение"? 1/1

- A) List
- B) Set
- C) Map ✓
- D) Queue

✓ Какой из классов обеспечивает постоянное время выполнения для основных операций, таких как add, remove и contains? 1/1

- ○ A) TreeSet
- ◉ B) HashSet ✓
- ○ C) LinkedList
- ○ D) TreeMap

Feedback

HashSet обычно предоставляет постоянное время выполнения для основных операций за счет использования хеш-таблицы.

✓ Вам необходимо реализовать коллекцию, в которой вы будете часто проверять наличие элемента. Какую структуру вы выберете? 1/1

- ◉ A) HashSet ✓
- ○ B) ArrayList
- ○ C) LinkedList
- ○ D) Stack

Feedback

HashSet предоставляет постоянное время выполнения для операции contains, благодаря использованию хеш-таблицы.

✓ Вам нужно сохранить пары ключ-значение, где порядок добавления элементов важен. Какую структуру данных вы выберете? 1/1

- A) HashMap
- B) LinkedHashMap ✓
- C) HashSet
- D) TreeMap

Feedback

LinkedHashMap сохраняет порядок вставки элементов.

✓ Вам нужен список, в котором вы будете часто вставлять и удалять элементы из середины. Что вы выберете? 1/1

- A) ArrayList
- B) LinkedList ✓
- C) HashSet
- D) Stack

Feedback

У LinkedList вставка и удаление из середины списка происходит быстрее, чем у ArrayList.

✓ Вам нужно сохранить набор элементов, который автоматически сортируется при добавлении нового элемента. Какую структуру вы выберете? 1/1

- A) HashSet
- B) LinkedHashSet
- C) TreeSet ✓
- D) LinkedList

Feedback

TreeSet автоматически сортирует элементы.

✓ Вам нужна структура данных, которая может эффективно представлять очередь задач, выполняемых по принципу "первым пришел, первым ушел". Что вы выберете? 1/1

- A) Stack
- B) Queue ✓
- C) List
- D) Set

Feedback

Queue представляет структуру данных в порядке FIFO.

✓ Вам требуется хранить уникальные элементы без какой-либо дополнительной сортировки или порядка. Какую структуру вы выберете? 1/1

- (•) A) HashSet ✓
- ( ) B) TreeSet
- ( ) C) LinkedList
- ( ) D) Stack

✓ Вам нужно быстро получать значения по уникальному ключу. Что вы выберете? 1/1

- (•) A) HashMap ✓
- ( ) B) LinkedHashMap
- ( ) C) LinkedList
- ( ) D) HashSet

✓ Основной особенностью этой структуры данных является то, что каждый элемент имеет ссылку на следующий элемент, формируя таким образом последовательность элементов. 1/1

- A) Упорядоченный массив
- B) Связный список (LinkedList) ✓
- C) Красно-черное дерево
- D) Очередь (Queue)

Feedback

Связный список состоит из нод, каждая из которых содержит данные и ссылку на следующий элемент.

✓ Эта структура данных основана на таблице с хеш-функцией, которая отображает ключи на индексы массива, в котором хранятся значения. 1/1

- A) Упорядоченный массив
- B) Двусвязный список
- C) Хеш-таблица (HashMap) ✓
- D) Очередь (Queue)

Feedback

Хеш-таблица использует хеш-функцию для отображения ключей на индексы массива, где хранятся значения.

✓ Эта структура данных обеспечивает хранение элементов в порядке их добавления и позволяет доступ как к началу, так и к концу коллекции. 1/1

- A) Упорядоченный массив
- B) Декью (Dequeue) ✓
- C) Красно-черное дерево
- D) Хеш-таблица (HashMap)

Feedback

Дек позволяет добавлять и удалять элементы как с начала, так и с конца.

✓ Эта структура данных хранит элементы в бинарном дереве, где каждый узел имеет два дочерних элемента: один с меньшим значением и один с большим значением. 1/1

- A) Упорядоченный массив
- B) Двусвязный список
- C) Двоичное дерево поиска (Binary Search Tree) ✓
- D) Очередь (Queue)

Feedback

Двоичное дерево поиска хранит элементы таким образом, что левое поддерево каждого узла содержит элементы с меньшими значениями, а правое поддерево — с большими значениями.

✓ Эта структура данных обеспечивает доступ к элементам на основе "первым пришел, последним ушел" (LIFO), позволяя добавлять и удалять элементы только с верхушки структуры. 1/1

- A) Упорядоченный массив
- B) Двусвязный список
- C) Двоичное дерево поиска (Binary Search Tree)
- D) Стек (Stack) ✓

Feedback

Стек позволяет добавлять и удалять элементы только с верхушки структуры, обеспечивая таким образом порядок LIFO.

✓ Какая структура данных предоставляет следующий метод: .containsValue(Object o)? 1/1

- A) Set
- B) Map ✓
- C) List
- D) Queue

Feedback

Метод .containsValue(Object o) принадлежит интерфейсу Map и позволяет проверить, содержится ли указанное значение в этой карте.

✓ Какая структура данных предоставляет следующий метод: .addFirst(E e)? 1/1

- A) Set
- B) Map
- C) List
- D) Deque ✓

Feedback

Метод .addFirst(E e) принадлежит интерфейсу Deque, который предоставляет методы для добавления элементов в начало или конец двусторонней очереди.

✓ Какая структура данных предоставляет следующий метод: .get(int index)? 1/1

- A) Set
- B) Map
- C) List ✓
- D) Queue

Feedback

Метод .get(int index) принадлежит интерфейсу List и позволяет получить элемент по указанному индексу.

✓ Какая структура данных предоставляет следующий метод: .poll()? 1/1

- A) Set
- B) Map
- C) List
- D) Queue ✓

Feedback

Метод .poll() принадлежит интерфейсу Queue и позволяет извлечь и удалить головной элемент этой очереди, возвращая null, если очередь пуста.

✓ Какая структура данных предоставляет следующий метод: .push(E item)? 1/1

- A) Set
- B) Map
- C) List
- D) Stack ✓

Feedback

Метод .push(E item) принадлежит классу Stack и позволяет добавить элемент на вершину этого стека.