

Что такое Iterator?

Iterator - это интерфейс в языке программирования Java, предназначенный для перебора элементов коллекций (например, списков, множеств, карт и т. д.). Он позволяет последовательно перебирать элементы коллекции и выполнять операции чтения и удаления.

В чем различие Iterator от Iterable?

Iterator - это объект, который позволяет перебирать элементы коллекции.

Iterable - это интерфейс, который определяет метод iterator(), возвращающий объект типа Iterator. Он используется для указания того, что коллекция может быть перебрана.

В чем преимущество Iterator по сравнению с Enumeration?

Iterator был введен в Java для замены устаревшего Enumeration. Преимущества Iterator включают в себя:

Возможность удаления элементов из коллекции во время перебора.

Поддержку типизации (Generics).

Улучшенную безопасность типов.

Какие особенности есть у ListIterator?

ListIterator - это расширение интерфейса Iterator, которое используется для перебора элементов в списке (например, ArrayList или LinkedList) в обоих направлениях (вперед и назад). ListIterator также позволяет вставлять и удалять элементы в любом месте списка.

Какие методы обязательны к реализации в Iterator?

В интерфейсе Iterator определены два обязательных метода:

boolean hasNext(): Возвращает true, если есть следующий элемент, и false, если достигнут конец коллекции.

E next(): Возвращает следующий элемент коллекции.

Какую функциональность предоставляем Iterable?

Интерфейс Iterable определяет единственный метод iterator(), который возвращает объект типа Iterator. Он позволяет коллекции быть "итерируемой", то есть перебираемой при помощи цикла for-each.

Для чего преимущественно нужен Splitter?

Splitter (от "split-iterator" или "splittable iterator") представляет собой интерфейс, предназначенный для параллельного разделения (splitting) итерируемых данных на части. Он используется в потоковых операциях (Stream API) для эффективного распределения обработки данных на несколько ядер процессора.

Какие способы прохода по коллекции знаете?

Существует несколько способов прохода по коллекциям в Java:

Цикл for-each (для объектов, реализующих Iterable).

Использование итератора (Iterator или ListIterator).

Потоковые операции с использованием Stream API.

Цикл for с индексами (для списков и массивов).

Рекурсивный обход для древовидных структур данных (например, деревьев).

Выбор метода зависит от типа коллекции, требований к производительности и цели обхода.