

## Pro Java №7

`Arrays.asList()`, который позволяет быстро создать и инициализировать список значениями.

Итератор (`Iterator`) в Java используется для последовательного перебора элементов коллекции.

Метод `next()` итератора позволяет извлечь следующий элемент коллекции и продвигает курсор итератора к следующему элементу.

Метод `hasNext()` проверяет, есть ли следующий элемент в коллекции, которую итератор обходит.

Когда работаешь с итератором в Java, важно понимать последовательность вызова методов и их правильное использование, чтобы избежать ошибок и исключений.

Итераторы доступны у всех коллекций в Java, так как большинство коллекций реализуют интерфейс `Collection`, который включает метод `iterator()`. Каждая коллекция предоставляет свою собственную реализацию итератора, адаптированную для работы с конкретной структурой данных.

Существуют несколько устаревших классов коллекций в Java, которые в основном заменены более современными реализациями в пакете `java.util`. Вот краткий обзор некоторых из них:

### 1. **`Vector`**:

- Вектор является динамическим массивом, который синхронизирован и расширяется при необходимости.
- Заменен на `ArrayList`, который не синхронизирован, но может быть синхронизирован вручную при необходимости.

### 2. **`Hashtable`**:

- Хеш-таблица хранит данные в виде пар ключ-значение и синхронизирована.
- Заменена на `HashMap`, который не синхронизирован, но может быть синхронизирован вручную при необходимости.

### 3. **`Stack`**:

- Стек является синхронизированной коллекцией, работающей по принципу LIFO (Last In, First Out).
- Заменен на `Deque` (например, `ArrayDeque` или `LinkedList`), который может быть использован в качестве стека.

Эти устаревшие классы по-прежнему работают, но современные реализации коллекций предоставляют более гибкие и эффективные способы работы с данными.

`System.currentTimeMillis()` возвращает текущее время в миллисекундах.