



Sertai sekarang

Daftar masuk



JAVA COLLECTIONS

Rangka Kerja Koleksi Java

...



Shivamma Kognure

Diterbitkan 7 Jul 2024

+ Ikuti

11



Suka



Komen



Kongsi

Lihat asal

Memahami Rangka Kerja Koleksi Java:

Di Java kita boleh menyimpan data menggunakan Arrays. Saiz Tatasusunan ditetapkan yang ditakrifkan semasa permulaan tatasusunan. Dalam masa nyata, pembangun memerlukan struktur data yang fleksibel dalam saiz, supaya data boleh ditambah dan dialih keluar atas permintaan.

Untuk mengelakkannya, setiap pembangun perlu melaksanakan struktur data tersuainya, yang memerlukan lebih banyak usaha pengekodan. Perpustakaan Java menyediakan beberapa pelaksanaan lalai untuk ini melalui rangka kerja koleksi. Artikel ini akan merangkumi komponen teras Rangka Kerja Koleksi, termasuk antara muka utamanya, kelas pelaksanaan dan beberapa amalan terbaik.

1. Pengenalan kepada Rangka Kerja Pengumpulan

Koleksi mewakili sekumpulan objek, yang dikenali sebagai elemennya. Koleksi dalam java ialah rangka kerja yang menyediakan seni bina untuk menyimpan dan memanipulasi kumpulan objek. Semua operasi yang anda lakukan pada data seperti mencari, menyusun, menyisipan, manipulasi, pemadaman dsb. boleh dilakukan oleh Rangka Kerja Koleksi Java.

Kelas pembalut:

- Koleksi menyimpan objek sahaja (bukan data primitif).
- Kelas pembungkus yang menyediakan fungsi untuk melaksanakan penukaran seperti
- Primitif kepada Objek (Tinjau)
- Objek kepada Primitif (Un tinju)

Ambil perhatian: Untuk setiap jenis primitif terdapat kelas pembungkus dalam Java

Generik:

- Seperti yang kita tahu, koleksi hanya menyimpan objek.
- Generik digunakan untuk menentukan jenis objek yang dibenarkan untuk

disimpan ke dalam Koleksi.

- Koleksi tanpa Generik: Benarkan menyimpan sebarang jenis Objek.

Sintaks: Koleksi c = Koleksi baru();

- Pengumpulan dengan Generik: Benarkan hanya jenis Objek data tertentu.
- Pengumpulan dengan Generik yang membolehkan sebarang jenis objek:

Collection Framework

Map Interface

2. Antara Muka Teras

Antara muka yang boleh diulang:

Antara muka boleh dibaca ialah antara muka akar untuk semua kelas koleksi.

Membolehkan objek diulangi menggunakan gelung foreach.

Antara muka iterator :

Menyediakan cara untuk mengulangi elemen dalam koleksi secara

eksplisit.

Antara muka iterator menyediakan kemudahan mengulangi elemen dalam arah hadapan sahaja.

Koleksi:

Antara muka akar dari mana semua antara muka koleksi lain mewarisi.

Senarai:

Koleksi yang dipesan (juga dikenali sebagai jujukan). Membenarkan elemen pendua.

Set:

Koleksi yang tidak membenarkan elemen pendua.

Giliran:

Koleksi yang digunakan untuk menyimpan berbilang elemen sebelum diproses.

Peta(Tidak Melanjutkan Pengumpulan) :

Objek yang memetakan kunci kepada nilai, tanpa kunci pendua dibenarkan.

3. Kelas Pelaksanaan Utama

Berikut ialah beberapa kelas pelaksanaan yang paling biasa digunakan untuk antara muka ini:

Senarai Pelaksanaan:

- **Senarai Tatasusunan :**

Kelas ArrayList mengekalkan tertib sisipan dan tidak disegerakkan.

Pelaksanaan tatasusunan yang boleh diubah saiz. Bagus untuk akses rawak.

```
// Creating and adding elements
List<Integer> list = new ArrayList<Integer>();
list.add(10);    // add elements
list.add(20);
list.sort();     // sort elements
// Using foreach loop (implicitly uses Iterable)
for (int i : list){
    System.out.println(i);
}
```

- **Senarai Berpaut:**

Pelaksanaan senarai yang dipautkan dua kali ganda.

Lebih baik untuk memasukkan dan memadam elemen, tetapi lebih perlahan untuk akses rawak.

```
List<String> al = new LinkedList<String>();
al.add("Ravi");
al.add("Vijay");

// Using explicit iterator (also uses Iterable)
Iterator<String> itr = al.iterator();
while(itr.hasNext()){
    System.out.println(itr.next());
}
```

- **Vektor:**

Serupa dengan ArrayList tetapi disegerakkan.

Dicadangkan oleh LinkedIn

Generik Java

Omar Ismail · 4 tahun yang lalu

Kuasai Seni Pemeriksaan Nol dalam Java: 11 Teknik...

Amir Goalmoradi · 7 bulan yang lalu

Menguasai Kutipan Sampah di Jawa

Nephat Muchiri · 10 bulan yang lalu

```

Vector<String> v = new Vector<String>();
v.add("Ayush");
v.add("Amit");
Iterator<String> itr = v.iterator();
while(itr.hasNext()){
    System.out.println(itr.next());
}

```

- **Timbunan:**

Timbunan ialah subkelas Vektor. Ia melaksanakan struktur data terakhir-dalam-pertama-keluar.

Memasukkan dan memadam elemen dari satu hujung.

```

Stack<String> stack = new Stack<String>();
stack.push("Ayush");    // push or add elements
stack.push("Garvit");
stack.push("Amit");
stack.pop();            // pop or remove elements
System.out.println(stack);    // can print in this way also

```

Tetapkan Pelaksanaan:

- **Set Hash:**

Satu set yang disokong oleh jadual cincang. Akses pantas tetapi tiada pesanan terjamin.

```
//Creating HashSet and adding elements
Set<String> set = new HashSet<String>();
set.add("Ravi");
set.add("Vijay");

Iterator<String> itr = set.iterator();           //Traversing
while(itr.hasNext()){
    System.out.println(itr.next());
}
```

- **LinkedHashSet:**

HashSet yang mengekalkan susunan sisipan.

```
LinkedHashSet<String> set = new LinkedHashSet<String>();
set.add("Ravi");
set.add("Vijay");

Iterator<String> itr = set.iterator();
while(itr.hasNext()){
    System.out.println(itr.next());
}
```

- **Set Pokok:**

Set yang disusun disokong oleh pokok. Unsur-unsur disusun berdasarkan nilainya.

```
SortedSet<String> set = new TreeSet<String>();
```



```

set.add("Ravi");
set.add("Vijay");
Iterator<String> itr = set.iterator();
while(itr.hasNext()){
    System.out.println(itr.next());
}

```

Pelaksanaan Baris Gilir:

- **Barisan Keutamaan:**

Pelaksanaan timbunan keutamaan.

Unsur-unsur disusun berdasarkan susunan semula jadi mereka atau oleh pembeding.

```

// Creating and adding elements
PriorityQueue<String> queue=new PriorityQueue<String>();
queue.add("Amit Sharma");
queue.add("Vijay Raj");

System.out.println("head:"+queue.peek()); // first element

// iterating the queue elements
Iterator itr=queue.iterator();
while(itr.hasNext()){
    System.out.println(itr.next());
}

```

- **Selesaikan:**

Baris gilir dua hujung, membolehkan penyisipan/penyingkiran elemen di

kedua-dua hujungnya.

```
Deque<String> deque = new ArrayDeque<String>();
    deque.add("Gautam");
    deque.add("Karan");
    deque.add("Ajay");
    for (String str : deque) {
        System.out.println(str);
    }
```

Pelaksanaan Peta:

- **Peta Hash :**

Pelaksanaan berasaskan jadual cincang. Membenarkan nilai nol dan satu kunci nol. Tiada pesanan yang dijamin.

```
// Creating Map with key in Integer and value in String
Map<Integer,String> map = new HashMap<Integer,String>();
    map.put(10, "Ten");
    map.put(20,"Twenty");
    System.out.println("Map :" + map); // it will print
```

- **Peta Hash Pautan :**

HashMap dengan susunan lelaran yang boleh diramalkan.

```
Map<Integer,String> map = new    LinkedHashMap<Integer,String>
```

```
map.put(20,"Twenty");  
map.put(10,"Ten");  
System.out.println("Map :" + map);
```

- **Peta Pokok :**

Peta yang disusun disokong oleh pokok. Memesan elemen berdasarkan kunci mereka.

```
Map<Integer,String> map = new TreeMap<Integer,String>();  
map.put(20,"Twenty");  
map.put(10, "Ten");  
System.out.println("Map :" + map);
```

4. Amalan Terbaik

Pilih Koleksi yang Betul

Gunakan jenis koleksi yang paling sesuai untuk keperluan anda (cth, gunakan Senarai untuk data tersusun, Set untuk elemen unik dan Peta untuk pasangan nilai kunci).

Minimumkan Pemutus

Lebih suka menggunakan generik untuk mengelakkan pemutus dan memastikan keselamatan jenis.

Optimumkan Prestasi

Berhati-hati dengan ciri-ciri prestasi koleksi yang berbeza. Sebagai

contoh, ArrayList lebih baik untuk akses rawak pantas, manakala LinkedList lebih baik untuk sisipan dan pemadaman yang kerap.

Gunakan Antara Muka

Program kepada antara muka, bukan pelaksanaan. Ini menjadikan kod anda lebih fleksibel dan lebih mudah untuk menukar pelaksanaan kemudian.

5. Kesimpulannya

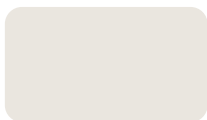
Rangka Kerja Koleksi Java adalah lebih daripada sekadar satu set struktur data. Ia mewakili aspek asas pengaturcaraan Java yang meningkatkan produktiviti, memastikan pengurusan data yang cekap dan menggalakkan amalan pengekodan terbaik.

Memahami dan menggunakan Rangka Kerja Koleksi Java dengan berkesan bukan sekadar pencapaian teknikal tetapi satu langkah ke arah menulis kod Java yang lebih profesional, boleh diselenggara dan berkualiti tinggi.

Selamat Pengekodan!

Untuk melihat atau menambahkan komen, [daftar masuk](#)

Orang lain turut melihat



Temubual #142: Java: Perbezaan antara Array dan ArrayList?

Software Testing Studio | WhatsApp 91-6232667387 · 8bln

© 2025

Kemudahan Akses

Dasar Privasi

Dasar Hak Cipta

Kawalan Tetamu

Bahasa

Perihal

Perjanjian Pengguna

Dasar Kuki

Dasar Jenama

Garis Panduan Komuniti