Collections

Tim Ajar Algoritma dan Struktur Data
Genap 2022/2023

Tujuan

- Memahami jenis-jenis Collection dan contoh penggunaannya
- Dapat mengimplementasikan Collection pada library Java untuk menyelesaikan studi kasus yang sesuai

Outline

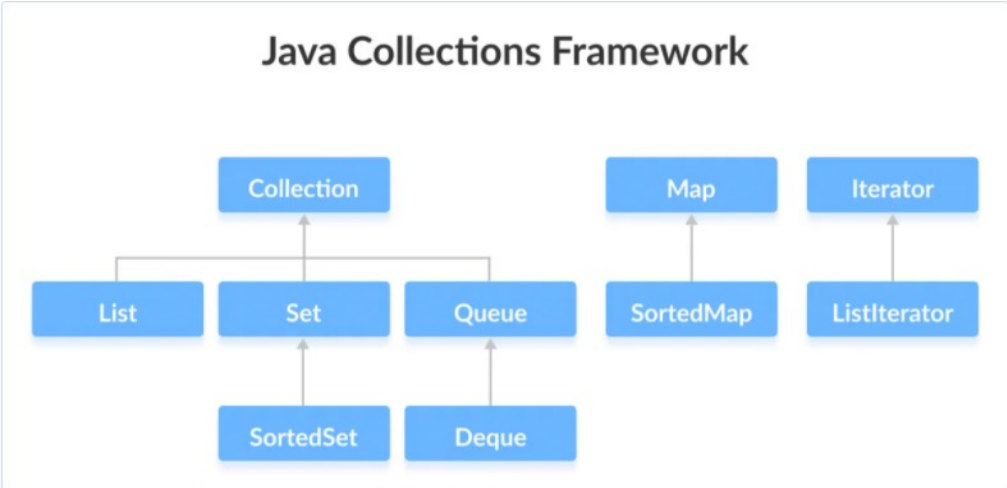
- Java Collection Framework
- Manfaat Java Collection Framework
- Sorting dan Searching
- Subinterface Collection
- Tugas Latihan

Pendahuluan

- Ingat, List, Stack dan Queue merupakan struktur data yang bersifat linear (Linear List) dan non linear (tree ataupun graph)
- Linear List digunakan untuk data yang terurut secara serial, sebaliknya non linear berarti data tersebut disajikan dalam bentuk yang tidak serial (hierarki ataupun tidak teratur). contoh: antrian, nama hari dalam minggu, nama bulan dalam tahun, jejaring medsos, topologi jaringan, dan yang lain.
- Silakan dibayangkan jika contoh di atas dibuat menggunakan teknik konvensional yang hanya memanfaatkan konsep kelas dan objek.
- Konsep sangat penting sebelum mempelajari teknik yang lebih canggih dan mudah. ☺
- Jenis struktur baik linear atapun non-linear akan dibungkus ke dalam Java Collection Framework.

Java Collection Framework

Sebuah Java Collection Framework menyediakan kumpulan dari sebuah interface ataupun class yang mengimplementasikan algoritma struktur data



https://www.programiz.com/java-programming/collections

Manfaat Java Collection Framework

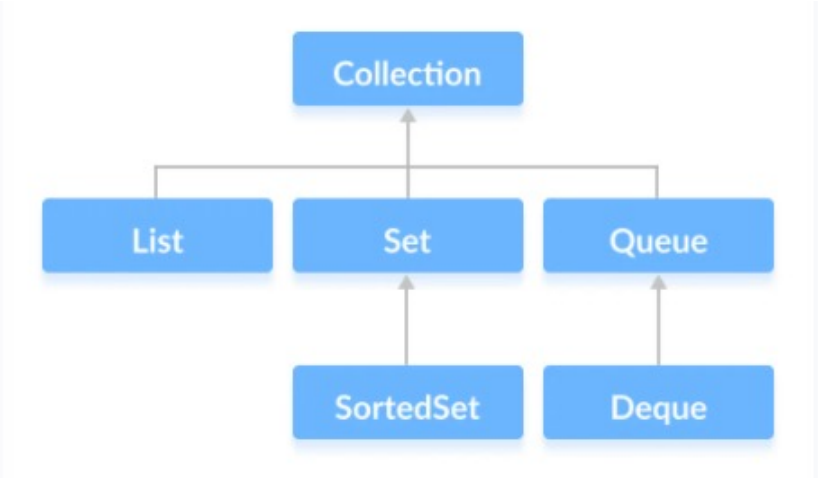
Java Collection Framework menyediakan struktur data dan algoritma yang bisa langsung kita gunakan. Adapun keuntungan/manfaat adalah sebagai berikut

- Kita tidak secara manual membuat kode ketika akan mengimplementasikan
- Kode yang kita buat akan jauh lebih efisien
- Kita juga bisa menggunakan java collection tertentu sesuai dengan struktur datanya. Misalkan ketika kita ingin menampung data yang unik, gunakan collection *Set*.

Sorting dan Searching Java Collection Framework

- Java Collection Framework menyediakan operasi standard dalam struktur data yaitu pengurutan dan pencarian data.
- Kedua fungsi tersebut terdapat di dalam paket java.util.Collections.
- Untuk mengurutkan data menggunakan fungsi static yaitu sort(), Collections.sort().
- Fungsi sort() pada Collections menerapkan algoritma merge sort.
- Untuk mencari data bisa menggunakan Collections.binarySearch()

Java Collection Interface



https://www.programiz.com/java-programming/collection-interface

- Interface Collection adalah root interface dari Java collections framework
- Untuk mengimplementasikan harus lewat sub-interface, seperti List, Set, dan Queue. Misalkan class Arraylist yang mengimplementasikan class List yang merupakan sub-interface dari Collection

Subinterface Collection

Interface Collection yang didalam terdapat subinterface yang dapat diimplementasikan oleh berbagi macam class di Java

- Interface List

Sebuah collection atau struktur data yang memungkinkan untuk menambahkan/menghapus elemen seperti sebuah array.

- Interface Set

Dengan Interface Set kita dapat menyimpan elemen dalam collection tidak akan terdapat duplikasi data.

- Interface Queue

Digunakan untuk menyimpan dan mengakses elemen dengan cara First In, First Out(FIFO)

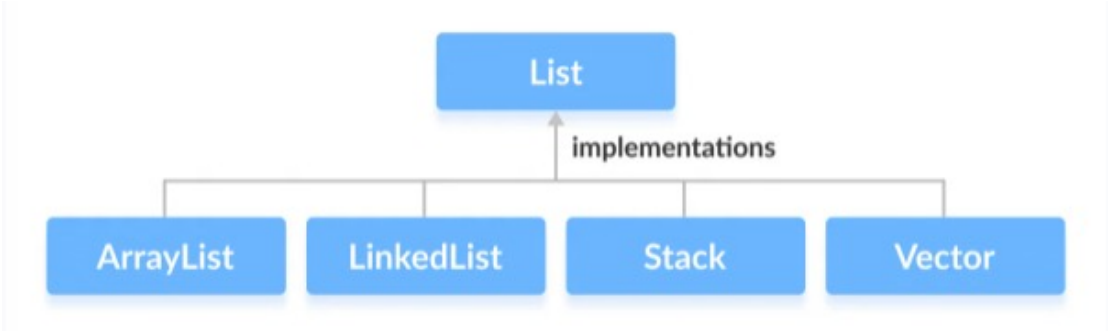
Metode/Fungsi Collection

Inteface Collection memiliki fungsi/metode yang bisa digunakan untuk memudahkan dalam struktur data, metode tersebut juga berlaku pada subinterface.

- add() – menambahkan elemen
- size() – mengembalikan ukuran collection
- remove() – menghapus elemen
- interator() – mengembalikan sebuah iterator untuk mengakses elemen-elemen
- addAll() – menambahkan semua elemen(collection) pada collection
- removeAll() – menghapus semua elemen dari collection tertentu
- clear() – menghapus semua elemen

Interface List

Interface List merupakan collection terurut yang memungkinkan untuk menyimpan dan mengakses elemen secara sequensial. Beberapa class yang mengimplementasikan interface List adalah ArrayList, LinkedList, Vector, dan Stack.



https://www.programiz.com/java-programming/list

Cara Penggunaan List

Karena List adalah sebuah interface, sehingga kita tidak bisa secara langsung membuat sebuah instance/object dari interface.

```
List list = new ArrayList(); (1)
List<String> list = new Stack<>(); (2)
```

- Untuk menggunakan List harus import java.util.List
- Objek list tidak bisa diinstance dari List tetapi dari implementasi class yaitu ArrayList
- Bentuk yang pertama adalah sebuah ArrayList yang bisa menampung semua tipe data
- Bentuk yang kedua adalah stack yang hanya bisa menampung String.
- Simbol <> adalah sebuah generic atau dibaca *of type String*.

Class ArrayList

ArrayList merupakan sebuah class dari Java Collection Framework yang menyediakan fungsi seperti array tetapi lebih dinamis dalam hal ukurannya.

https://www.programiz.com/java-programming/arraylist

Cara Menggunakan ArrayList

- Pertama yang harus dilakukan adalah import java.util.ArrayList

```
ArrayList<Type> arrayList= new ArrayList<>(); (1)
ArrayList<Integer> arrayList = new ArrayList<>(); (2)
ArrayList<String> arrayList = new ArrayList<>(); (3)
```

- Sintak umum diperlihatkan pada baris pertama, Type merupakan sebuah class wrapper agar objek arrayList spesifik menampung objek dari class tertentu
- Baris kedua artinya objek arrayList hanya dapat menyimpan data tipe integer
- Baris ketiga artinya objek arrayList hanya dapat menyimpan data tipe string
- Jika tidak disebutkan secara spesifik atau tidak ada Type maka objek arrayList bisa menampung data apa saja. ☹

Fungsi pada Class ArrayList

- size() – mengembalikan panjang arraylist
- sort() – mengurutkan elemen arraylist
- clone() – membuat arraylist baru dengan elemen, ukuran, dan kapsitas yang sama dari sebuah arraylist.
- contains() – mencari sebuah elemen pada arraylist dan mengembalikan nilai boolean
- ensureCapacity() – menentukan elemen arraylist yang dapat ditampung
- isEmpty() – mengecek apakah sebuah arraylist kosong
- indexOf() – mengembalikan index elemen pada sebuah arraylist

Class Stack

- Stack merupakan sebuah class dari Java Collection Framework yang menyediakan fungsi tumpukan pada struktur data.
- Pada sebuah stack, elemen dapat ditambahkan atau diakses menggunakan konsep Last In First Out(LIFO)
- Elemen ditambahkan(push) pada top stack dan diambil(pop) dari top stack.

https://www.programiz.com/java-programming/stack

Cara Menggunakan Stack

- Pertama yang harus dilakukan adalah import java.util.Stack

```
Stack<Type> stacks = new Stack<>(); (1)
Stack<Integer> stacks = new Stack<>(); (2)
Stack<Mahasiswa> stacks = new Stack<>(); (3)
```

- Sintak umum diperlihatkan pada baris pertama.
- Baris kedua artinya objek stacks hanya dapat menyimpan data tipe integer
- Baris ketiga artinya objek stacks hanya dapat menyimpan objek yang dinstansiasi dari objek Mahasiswa.

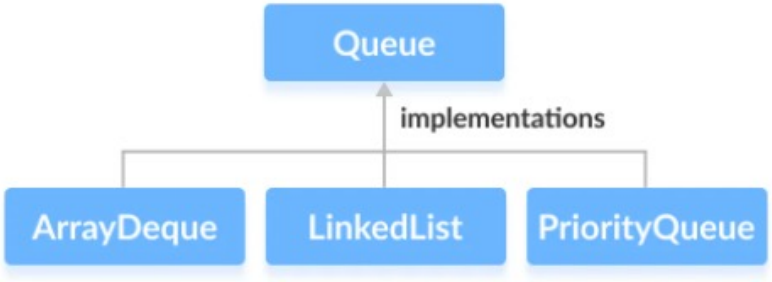
Fungsi pada Class Stack

Semua fungsi/metode yang dimiliki oleh class Vector maka dimiliki oleh class Stack, perbedaannya adalah sebagai berikut

- push() – menambahkan elemen pada stack
- pop() – mengeluarkan elemen pada stack
- peek() – mengembalikan elemen yang terdapat pada top stack
- search() – mengambalikan posisi elemen pada sebuah stack
- empty() – mengecek apakah sebuah stack kosong

Interface Queue

- Queue merupakan sebuah interface dari Java Collection Framework yang menyediakan fungsi antrian pada struktur data.
- Queue adalah sebuah interface sehingga tidak bisa langsung diimplementasikan, diinstansiasi
- Pada queue, elemen dapat ditambahkan dari belakang dan diakses/dihapus dari depan, First In First Out(FIFO)



https://www.programiz.com/java-programming/queue

Menggunakan Interface Queue

- Pertama yang harus dilakukan adalah import java.util.Queue

```
Queue<Type> queues = new LinkedList<>(); (1)
Queue<Integer> queues = new ArrayDeque<>(); (2)
Queue<Mahasiswa> queues = new PriorityQueue<>(); (3)
```

- Sintak umum diperlihatkan pada baris pertama.
- ArrayDeque merupakan sebuah class yang mengimplementasi dari interface Queue dan Deque. Dequeu sendiri subinterface dari Queue, perbedaannya adalah dapat menambahkan/mengurangi elemen tidak hanya dari depan tapi juga dari belakang
- PriorityQueue implementasi dari Interface Queue akan tetapi elemen yang memiliki nilai paling kecil selalu diletakkan pada bagian depan, walaupun ketika menambahkan tidak pertama kali.

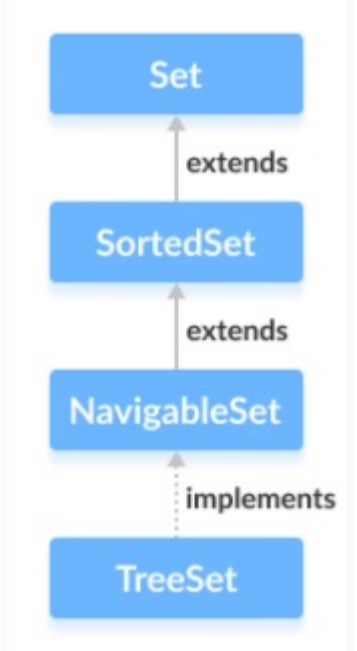
Fungsi pada Queue

Semua fungsi/metode yang dimiliki oleh interface Collection maka dimiliki juga oleh Queue, perbedaannya adalah sebagai berikut

- add() – menambahkan elemen tertentu pada queue
- offer() – menambahkan elemen tertentu pada queue
- element() – mengembalikan elemen yang terdapat pada head dari queue
- peek() – mengembalikan elemen yang terdapat pada head dari queue
- remove() – mengembalikan dan menghapus head pada queue
- poll() - mengembalikan dan menghapus head pada queue

Class TreeSet

- Sebuah class dari Java collection framework yang mengediakan fungsi dari struktur data tree.
- Struktur data jenis ini tidak diizinkan menampung data yang sama
- Sebenarnya class ini tidak secara spesifik dan detail seperti konsep tree yang telah dipelajari sebelumnya, tetapi bisa digunakan untuk struktur data tree yang sederhana. ☹



https://www.programiz.com/java-programming/treeset

Instansisasi TreeSet

- Pertama yang harus dilakukan adalah import java.util.TreeSet

```
TreeSet<Integer> trees = new TreeSet<>();
```

- Ketika instansiasi tidak diberikan argument, secara otomatis datanya terurut secara ascending.

Fungsi pada TreeSet

Semua fungsi/metode yang dimiliki oleh interface Collection dan turunannya maka otomatis akan dimiliki oleh class TreeSet.

- add() – menambahkan elemen tertentu pada tree
- addAll() – menambahkan elemen tertentu pada queue
- element() – mengembalikan elemen yang terdapat pada head dari queue
- peek() – mengembalikan elemen yang terdapat pada head dari queue
- remove() – mengembalikan dan menghapus head pada queue
- poll() - mengembalikan dan menghapus head pada queue

Framework Graph

- Java secara khusus tidak memiliki class yang menangani operasi-operasi dari graph, disarankan membuat class sendiri untuk mengimplementasikan.
- Class yang dibuat pada materi sebelumnya telah kita pelajari.
- Beberapa library yang bisa digunakan untuk mendukung operasi-operasi graph
  - JGraphT
  - JUNG — the Java Universal Network/Graph Framework
  - GraphStream

Latihan

Silahkan jawab pertanyaan di bawah ini

1. Jelaskan perbedaan dan persamaan antara List dan Set. Kemudian dalam kasus apa ketika menggunakan kedua struktur data tersebut?
2. Apakah perbedaan dari fungsi clear() dan removeAll() pada interface List dan jelaskan alasannya?
3. Apakah perbedaan fungsi add() dan offer() pada Queue, jelaskan?
4. Menurut Anda, mengapa Java tidak spesifik memiliki class yang menangani operasi-operasi Graph?
5. Jelaskan pengertian tipe data generic!
6. Mengapa instansiasi pada collections dapat menggunakan konstruktor yang berbeda (bukan class ybs)!


thank you
danke
謝謝
ngiyabonga
شكراً جزيلاً
teşekkür ederim
спасибо
gracias
tapadh leat
bedankt
dziękuję
obrigado
go raibh maith agat
sukriya
kop khun krap
arigatô
dakujem
merci
terima kasih
grazie
takk
мерси
рахмат
Баярлалаа
faafetai lava
vinaka
blagodaram
dank je
misaotra
matondo
paldies
grazzi
mahalo
хвала
asante
manana
obrigada
tenki
mochchakkeram
mamnun
chnorakaloutioun
gratias ago
grācies
sulpáy
taiku
sagolun
dakujem
trugarez
shukriya
dhanyavadagalu
diolch
xiexie
감사합니다
ευχαριστώ
mauruuru
hvala
dankie
kiitos
dhanyavad
köszönöm
dankon
akun
enkosi
sobodi
dēkuji
mèsi
didi madloba
najis tuke
kam sah hamnida
rahmat
tanemirt
rahmet
merci