

PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK

Nama	Ruben Kristanto	No. Modul	4
NPM	2306214624	Tipe	TP

1. Collection adalah sebuah struktur data yang digunakan untuk menyimpan banyak data dengan tipe data yang berbeda beserta method-method untuk mengelolanya. Collection bersifat dynamic yang isinya dapat bertambah atau berkurang tidak harus dipatok oleh ukuran dari struktur data yang dipakai. Ada 5 jenis
2. List seperti mirip seperti array menyimpan elemen dalam indeks dalam urutan tertentu dengan tipe data yang berbeda, ada 4 jenis list yang berbeda yaitu Stack, Array List, Vector, dan Linked List

a. ArrayList

Kelas ArrayList yang diimplementasikan dalam kerangka koleksi (Collection Framework) menyediakan array dinamis di Java. Meskipun mungkin lebih lambat dibandingkan array standar, ArrayList sangat berguna dalam program yang membutuhkan banyak manipulasi pada array. ArrayList memungkinkan penambahan, penghapusan, dan perubahan elemen dengan mudah. Karena ukurannya yang dinamis, ia dapat secara otomatis menyesuaikan kapasitasnya ketika jumlah elemen bertambah atau berkurang.

b. Vector

Vector adalah kelas dalam kerangka koleksi yang mengimplementasikan array yang dapat tumbuh atau menyusut sesuai kebutuhan. Kelas ini mirip dengan ArrayList, tetapi perbedaannya adalah Vector secara otomatis disinkronisasi sehingga aman untuk digunakan dalam lingkungan multithreaded. Seperti array biasa, Vector memungkinkan akses elemen menggunakan indeks. Meskipun termasuk dalam kelas warisan, Vector sekarang kompatibel sepenuhnya dengan koleksi modern di Java.

c. Stack

Stack adalah kelas dalam kerangka koleksi yang memperluas kelas Vector untuk mengimplementasikan struktur data Stack. Stack mengikuti prinsip Last-In-First-Out (LIFO), di mana elemen terakhir yang ditambahkan adalah elemen pertama yang dihapus. Kelas ini menyediakan fungsi dasar seperti push (menambahkan elemen), pop (menghapus elemen terakhir), serta fungsi tambahan seperti peek (melihat elemen teratas tanpa menghapusnya), empty, dan search.

D. LinkedList

LinkedList adalah kelas dalam kerangka koleksi yang mengimplementasikan struktur data linked list. Tidak seperti array, elemen-elemen dalam LinkedList tidak disimpan secara bersebelahan di memori. Setiap elemen disebut sebagai node, yang berisi data serta referensi ke elemen berikutnya. Hal ini membuat LinkedList unggul dalam hal penambahan dan penghapusan elemen di posisi mana pun dalam daftar, karena tidak perlu menggeser elemen seperti pada ArrayList.

Referensi :

[1]

GeeksforGeeks, "List Interface in Java with Examples - GeeksforGeeks," *GeeksforGeeks*, Oct. 27, 2016. <https://www.geeksforgeeks.org/list-interface-java-examples/>

[2]

"MODUL 4: Generic Type dan Collection | LEARN.NETLAB," *Netlabdte.com*, 2024.
<https://learn.netlabdte.com/docs/OOP/GenericTypedanCollection>

3. Map adalah Collection yang menyimpan pasangan kunci-nilai, dan memungkinkan objek-objek tersebut diakses berdasarkan kunci.

HashMap adalah bagian dari koleksi Java sejak Java 1.2. Kelas ini menyediakan implementasi dasar dari antarmuka Map di Java. HashMap menyimpan data dalam pasangan (Key, Value). Untuk mengakses sebuah nilai, kita harus mengetahui kuncinya. Kelas ini menggunakan teknik yang disebut Hashing. Hashing adalah teknik untuk mengubah String besar menjadi String kecil yang mewakili String yang sama. Nilai yang lebih pendek membantu dalam pengindeksan dan pencarian yang lebih cepat.

LinkedHashMap mirip dengan HashMap dengan fitur tambahan yaitu mempertahankan urutan elemen yang dimasukkan ke dalamnya. HashMap memberikan keuntungan dalam hal penyisipan, pencarian, dan penghapusan yang cepat, tetapi tidak mempertahankan urutan dan jejak penyisipan, yang merupakan fitur dari LinkedHashMap di mana elemen dapat diakses dalam urutan penyisipan mereka.

TreeMap di Java digunakan untuk mengimplementasikan antarmuka Map dan NavigableMap serta Kelas Abstrak. Map ini diurutkan sesuai dengan pengurutan alami dari kunci-kuncinya, atau menggunakan Comparator yang disediakan saat pembuatan map, tergantung pada konstruktor yang digunakan. Ini terbukti sebagai cara yang efisien untuk mengurutkan dan menyimpan pasangan key-value. Urutan penyimpanan yang dipertahankan oleh TreeMap harus konsisten dengan metode equals, seperti map terurut lainnya, terlepas dari adanya Comparator eksplisit.

Referensi :

[1]

GeeksforGeeks, "Map Interface in Java," *GeeksforGeeks*, Oct. 29, 2016.
<https://www.geeksforgeeks.org/map-interface-java-examples/>

[2]

"MODUL 4: Generic Type dan Collection | LEARN.NETLAB," *Netlabdte.com*, 2024.
<https://learn.netlabdte.com/docs/OOP/GenericTypedanCollection>

4. Generic type adalah fitur yang memungkinkan kita untuk membuat class, interface, dan method yang dapat bekerja dengan berbagai jenis data. Dengan menggunakan generic type, kita dapat merancang class, interface, dan method yang bisa menerima tipe data apa saja sebagai parameter, tanpa harus menentukan jenis data tertentu.

Keamanan Tipe (Type Safety): Dengan memanfaatkan generic type, kita bisa memastikan bahwa jenis data yang digunakan dalam kelas, antarmuka, atau metode tersebut sesuai dengan yang diharapkan.

Penggunaan Kembali (Reusability): Generic type memungkinkan kita membuat kelas, antarmuka, atau metode yang bisa digunakan dengan berbagai jenis data tanpa perlu menulis ulang implementasi untuk setiap tipe.

Keterbacaan Kode (Code Readability): Dengan penerapan generic type, kita dapat menulis kode yang lebih mudah dipahami dan dimengerti oleh programmer lain.

Referensi:

[1] "MODUL 4: Generic Type dan Collection | LEARN.NETLAB," *Netlabdte.com*, 2024.
<https://learn.netlabdte.com/docs/OOP/GenericTypedanCollection>

5. Penggunaan generic type dapat diibaratkan dengan polymorphism dimana parameter yang kita inisialisasikan saat pembuatan class disebalah nama class dengan <"isi tipe data"> tidak memiliki tipe data yang tetap yang berarti kita dapat mengisinya dengan berbagai macam tipe data untuk return type dan field, hanya saja kita harus tetap menentukan tipe data sebuah objek Ketika membuatnya

Ada beberapa aturan yang perlu diperhatikan saat menggunakan generic type dalam Java, antara lain:

- Generic type parameter harus ditentukan dengan menggunakan tanda kurung sudut (< dan >), diikuti oleh nama tipe data yang akan digunakan. Contoh: `Box<T>`.
- Generic type argument harus ditentukan saat membuat objek dari kelas atau interface yang menggunakan generic type. Contoh: `Box<Integer> box = new Box<>(10);`.

- Parameter tipe dapat digunakan untuk mendeklarasikan return type dari metode yang menggunakan generic type. Contoh: `public T getValue() { return value; }`.
- Java menggunakan konsep type erasure untuk menghapus informasi tipe data generic saat kode dikompilasi. Hal ini berarti bahwa informasi tipe data generic tidak tersedia saat kode dijalankan.

Referensi:

[1] “MODUL 4: Generic Type dan Collection | LEARN.NETLAB,” *Netlabdte.com*, 2024.
<https://learn.netlabdte.com/docs/OOP/GenericTypedanCollection>