LAPORAN PRAKTIKUM ALGORITMA PEMOGRAMAN

STACK PADA JAVA



OLEH:

FIKHRI HANIF

(2311533007)

DOSEN PENGAMPU:

DR. WAHYUDI, S.T, M.T

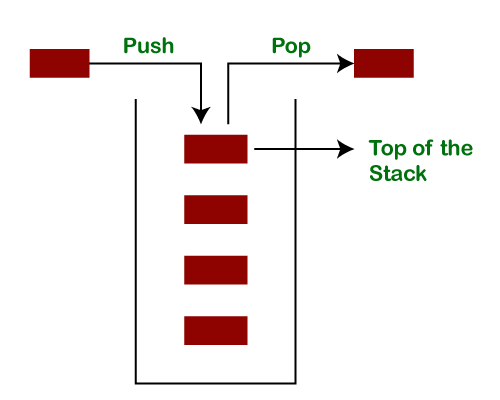
FAKULTAS TEKNOLOGI INFORMASI

DEPARTEMEN INFORMATIKA

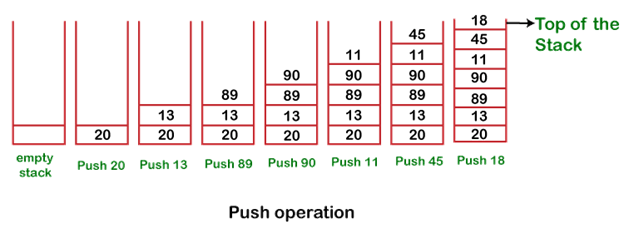
UNIVERSITAS ANDALAS

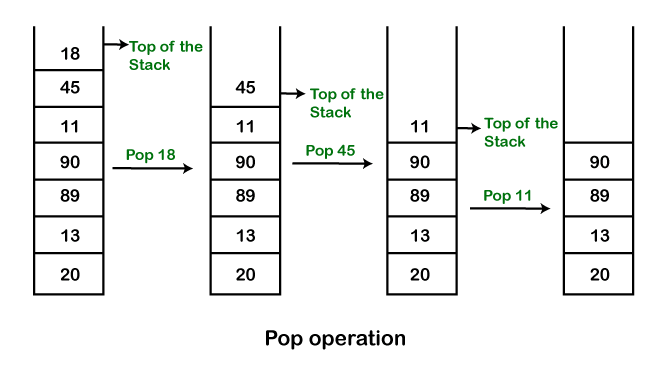
1. **Pendahuluan**

Stack pada bahasa pemrograman Java adalah sebuah Class array yang terdapat pada java.util package, tepatnya pada java.util.Stack;. Stack merupakan struktur data linear yang dapat menyimpan kumpulan dari beberapa objek. Stack didasari oleh prinsip **Last-In-First-Out (LIFO).** Artinya, sebuah data hanya bisa dimasukkan atau diambil hanya bagian dari index teratas saja. Stack dapat diibaratkan seperti sebuah tumpukan balok yang disusun keatas. Jika kita ingin mengambil balok yang tengah, kita harus mengambil balok paling atas terlebih dahulu, lalu balok yang tengah.



Ada dua method yang paling penting di Stack, yaitu push dan pop. Method push() berfungsi sebagai memasukkan data kedalam stack, sedangkan pop() mengambil data paling atas dan menghilangkan data paling atas. Untuk gambaran lebih detail, perhatikan gambar berikut:





Secara umum, ArrayList memiliki sintaks deklarasi seperti berikut:

**Stack<tipe\_data> nama\_variabel = new Stack<>(parameter);**

Beberapa method penting pada Stack di Java adalah sebagai berikut:

1. isEmpty() = return boolean, berfungsi untuk cek apabila sebuah stak kosong atau tidak
2. push() = memasukkan data ke stack
3. pop() = mengambil data di index paling atas dalam sebuah stack dan menghapus index paling atas
4. peek() = mengambil data di index paling atas dalam sebuah stack tanpa menghapus index paling atas
5. search(Object) = mencari posisi object yang terdapat pada stack.

Class Stack pada Java sering diaplikasikan pada beberapa program, contohnya dalam penggunaan backtracking dan manajemen memori sistematis. Stack juga memiliki beberapa keuntungan, yaitu:

1. Dapat membantu mengolah data dengan metode “Last-In-First-Out”
2. Stack dapat digunakan ketika variabel tidak digunakan diluar sebuah fungsi dalam program
3. Dapat mengontrol alokasi memory dengan baik
4. Otomatis membersihkan sebuah object.

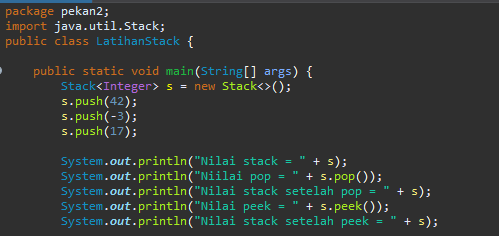
Stack juga memiliki beberapa kekurangan, diantaranya sebagai berikut:

1. Sangat susah untuk stack dibuat banyak objet karena akan meningkatkan resiko Stack Overvlow
2. Memori nya sangat dibatasi
3. Tidak dapat menggunakan random access
4. **Tujuan**

Tujuan dilakukannya praktikum ini adalah sebagai berikut:

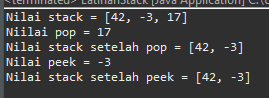
1. Memahami apa yang dimaksud dengan Stack
2. Memahami cara kerja penambahan data di Stack
3. Memahami cara kerja akses data di Stack
4. Memahami cara kerja penghapusan data di Stack
5. **Langkah Kerja Praktikum**
6. Cara kerja stack

Perhatikan program berikut:



Pada program tersebut, kita dapat menambahkan data kedalam stack dengan cara menggunakan method push(); untuk melihat isi dari stak tersebut secara keseluruhan, cukup panggil variabel stack tersebut. Jika kita ingin memanggil nilai stack paling atas dan menghapus data paling atas, dapat menggunakan method pop(). Jika kita hanya ingin memanggil nilai stack paling atas saja, dapat menggunakan method peek();

Contoh keluarannya adalah sebagai berikut:



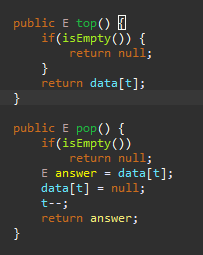
1. Pengenalan lebih dalam mengenai stack

Perhatikan kode berikut:

1. Class ArrayStack

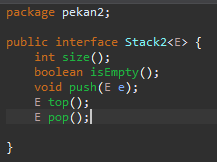
Muhammad Farhan Saputra  
2311532018  
14 Mei 2024





1. Interface Stack2

Muhammad Farhan Saputra  
2311532018  
14 Mei 2024



Program tersebut adalah rincian mengenai cara kerja stack. Kita bisa membuat sistem stack sendiri tanpa mengimport dari java.util.Stack;. Program tersebut bernama ArrayStack yang mengimplement interface Stack2. Cara kerja ArrayStack sama seperti Stak pada umumnya yang mengimplementasi menggunakan Array.

Berikut rincian dari setiap method:

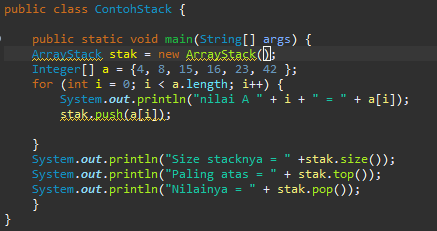
* “public static final int CAPACITY = 1000;”: berisikan batas maksimal element yang dapat stack tersebut simpan, yaitu 1000 buah.
* “private E[] data;”: pada program ini, dasar stacknya yaitu array yang direpresentasikan oleh variabel data. Jadi, data yang masuk akan tersimpan pada variael ini.
* “private int t = -1;”: method ini berisikan tentang informasi jumlah index pada stack. -1 menunjukkan bahwa stack tersebut kosong.
* public ArrayStack(): Merupakan default constructor dari class ArrayStak, yang mana memiliki kapasitas default 1000
* public ArrayStack(int capacity): Merupakan overloaded constructor yang mana jika dipanggil, akan digunakan kapasitas sesuai input yang diberikan.
* public int size(): Merupakan method untuk mengeluarkan ukuran dari stack tersebut. Cara method ini melihat ukuran stack yaitu dengan mereturn t+1
* public boolean isEmpty(): Method ini mengecek apakah t = -1 yang berarti stack kosong. Jika stack kosong, akan bernilai true. Jika stack tidak -1, maka bernilai false.
* public void push(E e) throws IllegalStateException: Method ini memasukkan data kedalam stack. Method ini juga melihat kapasitas yang disediakan saat pendeklarasian. Jika method dipanggil lebih dari kapasitas, akan mengeluarkan “Stack is full”.
* public E top(): Method ini akan mengeluarkan data teratas dari stack. Jika stack kosong, bernilai null

Muhammad Farhan Saputra  
2311532018  
14 Mei 2024

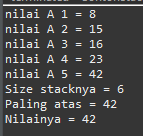
* public E pop(): Method ini mengeluarkan data teratas dan menghapus data teratas di stack. Jika stack kosong, return null. Method ini juga akan mengurangi nilai t.

1. Implementasi Stack

Dari program ArrayStack diatas, kita dapat membuat contoh program seperti berikut:

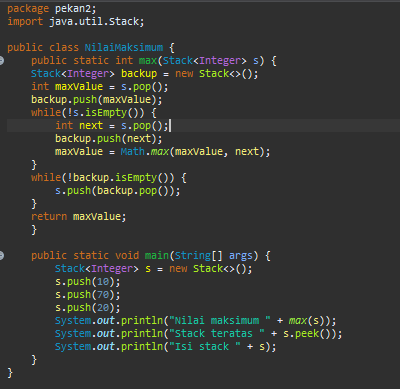


Alih-alih menggunakan stack, kita bisa menggunakan class yang telah kita buat sebelumnya yaitu ArrayStack. Untuk sistemnya sama seperti stack pada umumnya, berikut contoh keluarannya:



Berikut contoh lainnya dalam implementasi stack:

Muhammad Farhan Saputra  
2311532018  
14 Mei 2024



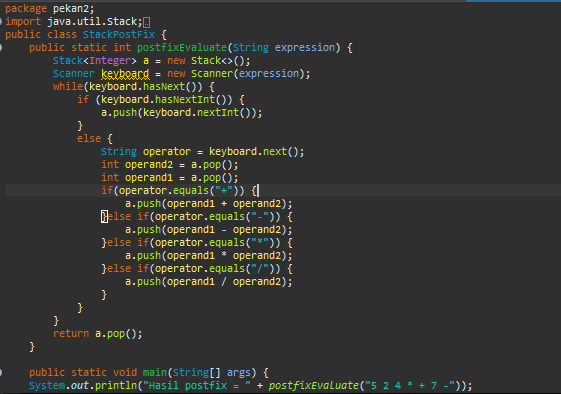
Dari program tersebut, kita membuat 2 stack, yaitu stack utama pada method main, dan stack backup pada method max. Program ini berfungsi untuk mencari nilai maksimum dari sebuah stack. Cara kerjanya yaitu stack akan dimasukkan ke method max(), lalu stack yang telah dimasukkan di pop() dan dimasukkan ke variabel maxValue. Kita juga membuat variabel kedua yaitu variabel next untuk memasukkan data kedua sebagai perbandingan dari nilai maksimal.

Contoh keluarannya adalah sebagai berikut:



Berikut contoh lainnya dalam penggunaan stack

Muhammad Farhan Saputra  
2311532018  
14 Mei 2024



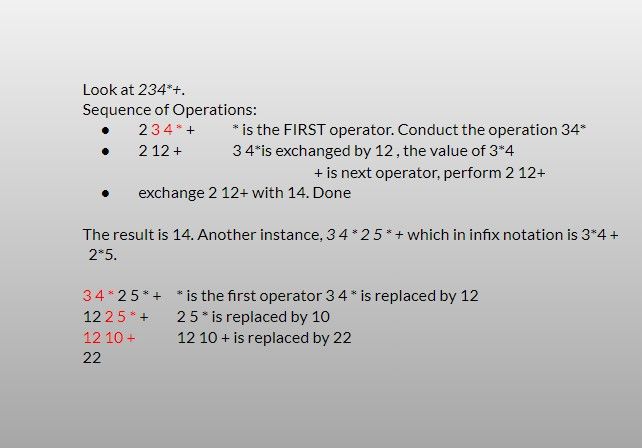
Program tersebut akan menerjemahkan operasi postfix menjadi nilai sebenarnya. Seperti yang kita ketahui, operasi yang biasa kita lakukan sehari hari bernama infix, contohnya seperti berikut:

2 \* 3 + 5

Cara kerja infix yaitu kita membaca dari kiri ke kanan, lalu mengoperasikan perkalian dan pembagian terlebih dahulu, lalu penjumlahan dan pengurangan.

Pada postfix, cara kerja operasinya adalah berikut:

Muhammad Farhan Saputra  
2311532018  
14 Mei 2024



Contoh keluaran program tersebut adalah sebagai berikut:



1. Kesimpulan

Berdasarkan praktikum yang penulis telah lakukan, kita dapat menarik kesimpulan bahwa Stack pada bahasa pemrograman Java adalah sebuah Class array yang terdapat pada java.util package, tepatnya pada java.util.Stack. ;. Stack merupakan struktur data linear yang dapat menyimpan kumpulan dari beberapa objek. Stack didasari oleh prinsip **Last-In-First-Out (LIFO).**