## Soal Tugas & Evaluasi

## Apa yang dimaksud dengan Lis, set, queue, dan map

## Jawaban

List adalah tipe data collection yang memungkinkan menyimpan elemen elemen dalam urutan tertentu (berdasarkan index) elemen elemen list bisa di duplikat

Set adalah tipe data collection yang memungkinkan menyimpan kumpulan elemen elemen unik, artinya tidak ada elemen duplikat

Queue adalah tipe data collection yang mewakili antrian (queue) dimanan elemen pertama yang dimasukkan adalah elemen pertama yang dikeluarkan (konsep FIFO first-in-first-out)

Map adalah tipe data collection yang terdiri atas kunci (key) ke nilai(value), mirip dengan konsep dictionary

## Soal Tugas & Evaluasi

1. Pada ilustrasi didata mengapa map tidak termasuk kedalam interface collection?

## Jawaban

Karena map tidak digunakan untuk mengelompokkan atau menyimpan elemen atau objek

## Soal Tugas & Evaluasi

1. Jelaskan apa bedanya arraylist, dan linkedlist!

## Jawaban

Arraylist kita dapat mendefinisikan nilai berapapun yg kita mau, karena arraylist bersitfat dinamis

## Kalau linkedlist menggunakan struktur data berbasis node, di mana setiap node menyimpan data dan referensi ke node berikutnya (dan sebelumnya, jika menggunakan double linked list).

## Soal Tugas & Evaluasi

1. Sebuah bank ingin mengelola antrian pelanggan yang dating untuk mendapatkan layanan. Setiap kali seorang pelanggan selesai dilayanin, pelanggan tersebuat akan keluar dari antrian, dan pelanggan berikutnnya akan dilayanin. Bank juga ingin memungkinkan pelanggan vip untuk masuk ke antrian di posisi tertentu. Data collection apa yang sebaiknya digunakan studi kasus tersebut?

## Jawaban

Queue karena bisa dibuat untuk mengelola antrina dan menyimpan data pelanggan

## Soal Tugas & Evaluasi

1. Buatlah program berdasasrkan dari studi kasus sebelumnya

## Source Code

*import* java.util.ArrayList;  
*import* java.util.List;  
  
*class* bank {  
 List<String> antrian;  
  
 *public* bank() {  
 antrian = *new* ArrayList<>();  
 }  
  
 *public void* tambahAntrian(String nama) {  
 antrian.add(nama);  
 System.out.println("Antrian Atas Nama " + nama + " telah di tambahkan");  
 }  
  
 *public void* antrianVIP(String nama, *int* nomor) {  
 *if* (nomor < 0 || nomor > antrian.size() + 1) {  
 System.out.println("Antrian Tidak valid");  
 }*else* {  
 nomor = nomor 1;  
 antrian.add(nomor, nama);  
 System.out.println("Antrian VIP Atas Nama " + nama + " telah di tambahkan" + ".");  
 }  
 }  
  
 *public void* tampilanAntrian() {  
 System.out.println("Antrian saat ini : ");  
 *for* (String nama : antrian) {  
 System.out.println(antrian.indexOf(nama) + 1 + "."+ nama);  
 }  
 }  
}  
  
  
  
*public class* bab4no5 {  
 *public static void* main(String[] args) {  
 bank bank = *new* bank();  
 bank.tambahAntrian("Adam");  
 bank.antrianVIP("luthfi",2 );  
 bank.tampilanAntrian();  
 }  
}

**Penjelasan**

Membuat class bank dan membuat atribut list untuk menyimpan antrian, lalu membuat metode tambah antrian dengan konstrak nama untuk menambah nama dan mencetak nama, lalu membuat metode antrian vip, memiliki pengkodisian jika kurang dari 0 atau lebih dari 1 akan mencetak antrian tidak valid dan else nomor -1 untuk mencari indeks ke 0 dan menambahkan antrian dengan parameter nomor dan nama dan mencetak antrian vip, lalu membuat tampilan atnriannya dan dicetak. Lalu membaut objek bank untuk menginisialisasinya.

**Output**

|  |
| --- |
|  |

## Soal Tugas & Evaluasi

6. Vincent diminta dosennya untuk membuat sebuah sistrm manaejeme perpustakaan sederhana. System ini harus mampu mengelola data buku, anggota perpustakaan , dan transaksi peminjaman buku, tidak hany satu Vincent diwajibkan menggunakan berbagai jenis data collection seperti arraylist, hashmap, dan queue. Bualah program untuk vincent

## Source Code

*import* java.util.ArrayList;  
*import* java.util.HashMap;  
*import* java.util.List;  
*import* java.util.Map;  
  
*class* Buku {  
 String judul;  
 String penulis;  
  
 *public* Buku(String judul, String penulis) {  
 *this*.judul = judul;  
 *this*.penulis = penulis;  
 }  
}  
  
*class* Anggota {  
 String nama;  
 String idAnggota;  
  
 *public* Anggota(String nama, String idAnggota) {  
 *this*.nama = nama;  
 *this*.idAnggota = idAnggota;  
 }  
}  
  
*class* Perpustakaan {  
 *private* List<Buku> daftarBuku;  
 *private* Map<String, Anggota> daftarAnggota;  
 *private* List<String> antrianPeminjaman;  
  
 *public* Perpustakaan() {  
 daftarBuku = *new* ArrayList<>();  
 daftarAnggota = *new* HashMap<>();  
 antrianPeminjaman = *new* ArrayList<>();  
 }  
  
 *public void* tambahBuku(Buku buku) {  
 daftarBuku.add(buku);  
 System.out.println("Buku " + buku.judul + " telah ditambahkan.");  
 }  
  
 *public void* tambahAnggota(Anggota anggota) {  
 daftarAnggota.put(anggota.idAnggota, anggota);  
 System.out.println("Anggota " + anggota.nama + " telah ditambahkan.");  
 }  
  
 *public void* pinjamBuku(String idAnggota, String judulBuku) {  
 *if* (!daftarAnggota.containsKey(idAnggota)) {  
 System.out.println("Anggota tidak ditemukan.");  
 *return*;  
 }  
  
 *for* (Buku buku : daftarBuku) {  
 *if* (buku.judul.equalsIgnoreCase(judulBuku)) {  
 antrianPeminjaman.add(idAnggota);  
 System.out.println("Anggota " + idAnggota + " telah meminjam buku " + judulBuku + ".");  
 *return*;  
 }  
 }  
 System.out.println("Buku tidak ditemukan.");  
 }  
  
  
 *public void* tampilkanDaftarBuku() {  
 System.out.println("Daftar Buku:");  
 *for* (Buku buku : daftarBuku) {  
 System.out.println("- " + buku.judul + " oleh " + buku.penulis);  
 }  
 }  
}  
  
*public class* bab4no6 {  
 *public static void* main(String[] args) {  
 Perpustakaan perpustakaan = *new* Perpustakaan();  
  
 perpustakaan.tambahBuku(*new* Buku("War Machine", "Sulistiawan"));  
 perpustakaan.tambahBuku(*new* Buku("Magic Of Snake", "Tolkien"));  
  
 perpustakaan.tambahAnggota(*new* Anggota("Adam", "1"));  
 perpustakaan.tambahAnggota(*new* Anggota("Nana", "2"));  
  
 perpustakaan.pinjamBuku("1", "War Machine");  
 perpustakaan.pinjamBuku("2", "Magic of snake");  
  
  
 perpustakaan.tampilkanDaftarBuku();  
 }  
}

**Penjelasan**

Membuat class buku dengan attribute judul penulis dan membuat konstruktornya dan membuat class anggota denang atrribut id anggota, dan nama, lalu membuat class perpustakaan dengan membuat attribute dan membuat list buku untuk menyimpan buku, map string anggota untuk untuk menyimpan daftar anggota, dan list string untuk menyimpan peminnjaman, lalu membuat metod tambah buku, tambah anggota, dan pinjam buku, dan tampilkan daftar buku, lalu membuat objek perpus untuk menginisialisasi dan memanggil untuk menampilkannya

**Output**

|  |
| --- |
|  |