1. Perhatikan pertanyaan di bawah ini terkait JAVA OOP, beri penjelasan secara singkat.
   1. Apa yang anda ketahui tentang 3 pilar dalam konsep OOP dan ciri khas/kata kunci yang digunakan tiap konsep tersebut
   2. Apa yang anda ketahui dari fungsi super dan this
   3. Apa yang anda ketahui perbedaan method overloading dan method overriding
   4. Apa yang anda ketahui perbedaan abstract dan interface

a. 3 pilar dalam konsep OOP dan ciri khas/kata kunci yang digunakan tiap konsep tersebut  
  
Enkapsulasi  
Enkapsulasi adalah konsep OOP yang membungkus data dan fungsi dalam sebuah unit. Ciri khas enkapsulasi adalah penggunaan kata kunci private untuk menyembunyikan data dan fungsi dari luar kelas.

Inheritance  
Inheritance adalah konsep OOP yang memungkinkan suatu kelas mewarisi properti dan perilaku dari kelas lain. Ciri khas inheritance adalah penggunaan kata kunci extends untuk menghubungkan dua kelas.

Polymorphism

Polymorphism adalah konsep OOP yang memungkinkan objek dari kelas yang berbeda dapat merespons panggilan metode yang sama dengan cara yang berbeda. Ciri khas polymorphism adalah penggunaan kata kunci override untuk menimpa metode yang diwariskan dari kelas induk.

b. Fungsi super dan this

Super

Super adalah kata kunci yang digunakan untuk mengakses anggota kelas induk dari kelas turunan.

This

This adalah kata kunci yang digunakan untuk merujuk ke objek saat ini.

c. Perbedaan method overloading dan method overriding

Method overloading

Method overloading adalah kemampuan untuk mendeklarasikan dua atau lebih metode dengan nama yang sama, tetapi dengan parameter yang berbeda.

Method overriding

Method overriding adalah kemampuan untuk mendeklarasikan ulang metode yang diwariskan dari kelas induk.

Perbedaan antara method overloading dan method overriding adalah sebagai berikut:

|  |  |  |
| --- | --- | --- |
| Kriteria | Overloading | Overriding |
| Tujuan | Untuk menyediakan fungsionalitas yang berbeda dengan nama yang sama | Untuk mendefinisikan ulang perilaku metode yang diwariskan |
| Parameter | Parameter harus berbeda | Parameter dapat sama atau berbeda |
| Posisi | Metode overloading dapat dideklarasikan di kelas yang sama atau kelas turunan | Metode overriding hanya dapat dideklarasikan di kelas turunan |
|  |  |  |

d. Perbedaan abstract dan interface

Abstract

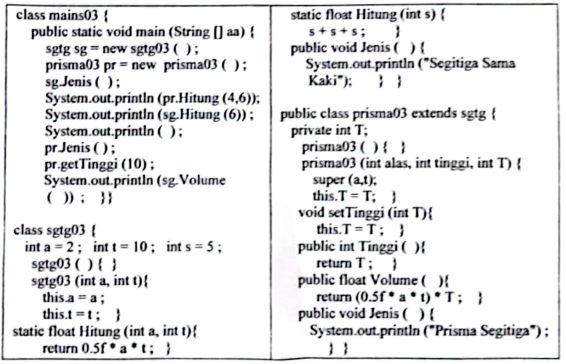
Abstract adalah kata kunci yang digunakan untuk mendeklarasikan kelas abstrak. Kelas abstrak tidak dapat diinstansiasi, tetapi dapat digunakan sebagai dasar untuk kelas turunan.

Interface

Interface adalah kumpulan metode abstrak. Interface tidak dapat diinstansiasi, tetapi dapat digunakan untuk menentukan kontrak antara kelas-kelas.

Perbedaan antara abstract dan interface adalah sebagai berikut:

|  |  |  |
| --- | --- | --- |
| Kriteria | Abstract | Interface |
| Instansiasi | Tidak dapat diinstansiasi | Tidak dapat diinstansiasi |
| Metode | Dapat memiliki metode abstrak dan metode konkret | Hanya dapat memiliki metode abstrak |
| Implementasi | Metode abstrak harus diimplementasikan oleh kelas turunan | Metode abstrak dapat diimplementasikan oleh kelas turunan atau kelas yang mengimplementasikan interface tersebut |



1. Jika ada yang salah dari coding di atas, silahkan perbaiki dan salin syntax yang benar (cukup syntax yang salah = syntax yang diperbaiki saja), tampilkan outputnya.
2. Analisis dan tulis nama konsep OOP, class, object, jenis method dan keyword yang di gunakan pada coding tersebut
3. Kesalahan pada method Hitung(int s):
   * Method ini tidak mengembalikan nilai (return value) dengan return.
   * Perbaikan: return s + s + s;
4. Kesalahan pada akses method Volume():
   * Method Volume() tidak ada dalam class sgtg03. Seharusnya memanggil method Volume() dari object pr.
   * Perbaikan: System.out.println(pr.Volume());
5. Kesalahan pada nama method getTinggi():
   * Method getTinggi() tidak ada dalam class prisma03, seharusnya setTinggi().
   * Perbaikan: pr.setTinggi(10);

b. Analisis konsep OOP:

* Class: mains03, sgtg03, prisma03
* Object: sg, pr
* Inheritance: prisma03 mewarisi dari sgtg03
* Method:
  + Constructor: sgtg03(), sgtg03(int a, int t), prisma03(), prisma03(int alas, int tinggi, int T)
  + Static method: Hitung(int a, int t), Hitung(int s)
  + Instance method: Jenis(), setTinggi(int T), Tinggi(), Volume()
* Keywords: class, public, static, void, int, float, return, this, super, `

3. Buatlah program perawatan salon dari output terdiri dari 3 class (abs\_salon, jenis\_salon, dan main\_salon). Class abs\_salon sebagai abstract class, gunakan konsep inheritance dan inputan scanner. Ketentuan class sebagai berikut

1. 1 class abstract mendeklarasi abstract void jenis, abstract void pelayanan, abstract void cetakdata(String nama), abstract void cetaktotal beserta deklarasi atribut,
2. 1 class turunan mengoverride 4 method abstract dan atribut untuk data pemesanan serta percabangan if majemuk untuk menentukan jenis paket serta orang dan pelayanan,
3. Hitung total biaya dari hasil pilihan paket\*orang+pelayanan dan tampilkan data pesanan beserta total biayanya sebagai struknya
4. 1 class main dimana objek memanggil method dan atribut dari class instance





