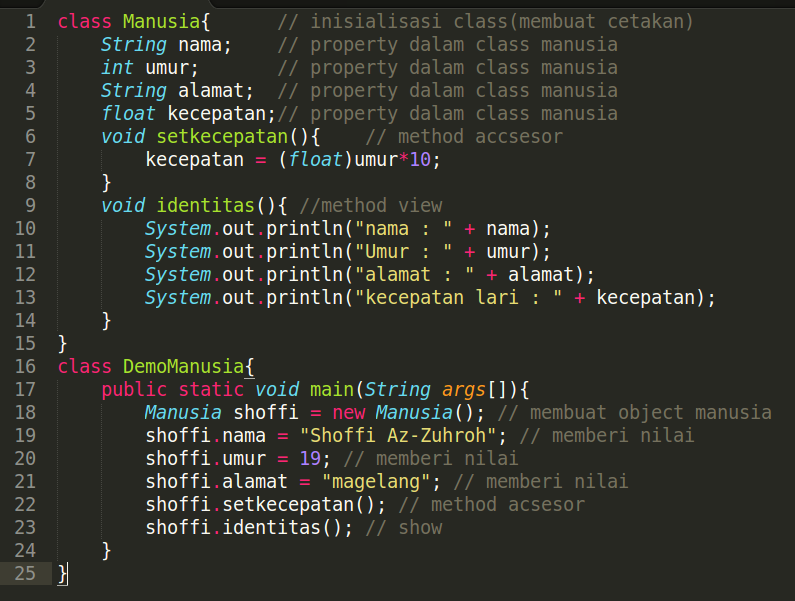
Pertemuan ke – 5   
OOP part 2

refresh....  
apa itu class ?  
Apa itu method ?

Apa itu property ?

Apa itu object ?

Let's check this code …



jadi :

class adalah sebuah abstraksi atau cetakan atau blueprint dari sebuah object. Class berisi apa-apa yang akan di miliki oleh objeck bentukanya .

Object adalah bentukan dari class, object memiliki apa-apa yang di miliki oleh class rujukanya.

Property adalah variabel yang dimiliki oleh object atau class

method adalah fungsi yang dimiliki oleh class.

Lanjut ke sifat OOP yang ke dua yaitu

* Inheritance

adalah pewarisan. artinya sebuah class dapat mewarisi atribut dan metho dari class lain  
dalam hal ini ada 2 class yaitu (penyebutan)

* + Super Class / Parent Class  
    adalah class indukan dimana sifatnya(attribut dan method) dicontoh atau di turunkan ke class lain
  + Sub Class / Child Class  
    adalah class turunan yaitu kelas yang mencotoh sifat(attribut dan method) dari class lain.

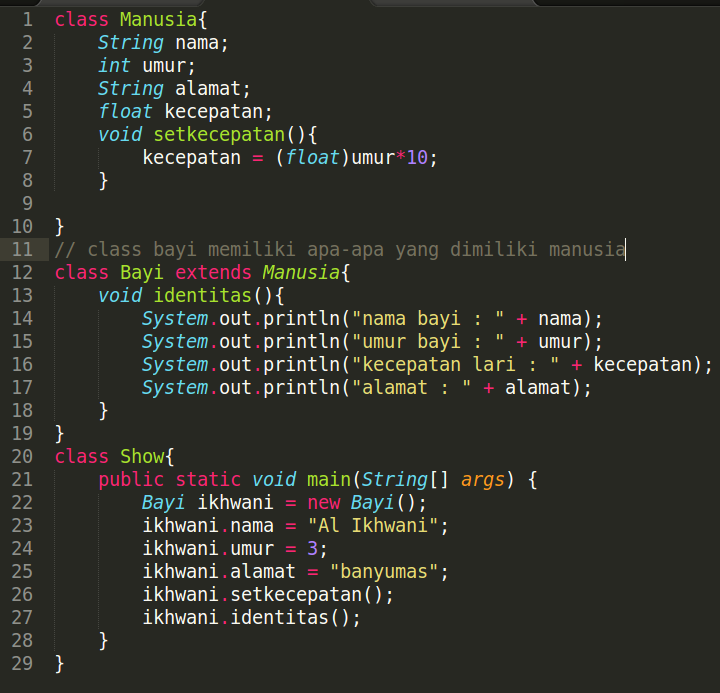
Rumus melakukan penurunan adalah sebagai berikut :

class namaClassChild extends namaClassParent {

}

dari rumus di atas dapat dilihat untuk menurunkan suatu class ke class lainya menggunakan kata kunci ***extends***

***contoh program***



Lanjut ke sifat OOP yang ke dua yaitu :

* **Polymorpism**

filosofi pemrograman “pada dasar nya pemrogram itu sangat mudah, yang membuatnya sulit adalah penerapan yang kompleks”.

Seperti yang kita bahas sebelumnya-sebeumnya perkembangan dunia pemrograman dalam han ini metode squensial kemudian OOP kemudian modular programming. Begitu pula dengan entity dalam pemrograman itu sendiri, yang awalnya variabel tunggal, kemudian naik menjadi array kemudian fungsi eksternal lalu terahir adalah class.

Nah perkembangan semua itu di karenakan tingkat ke kompleksan dari penggunaan.

Masih ingan percabangan ?  
Nah dalam percabangan kuncinya adalah kondisi yang berbeda sehingga penanganan yang berbeda.

Dalam class karena tingkat kekompleksan yang terus berkembang di butuhkan handling atao penanganan yang berbeda dari beberapa kondisi yang di terima oleh class nah dalam hal ini class harus memiliki fleksibilitas dalam menerima inputan dari user.

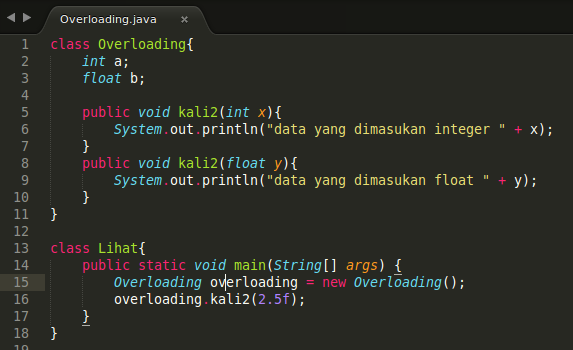
Polymorpism memiliki arti banyak bentuk.

Polymorpism dalam dalam hal ini adalah method dari sebuah class memiliki banyak bentuk menyesuaikan inputan parameter dari user.

dalam Polymorpism ada 2 sifat yaitu :

1. Overloading

Overloading adalah pemakaian beberapa methods ataupun properties dengan nama yang sama, tetapi memiliki daftar parameter/argument yang berbeda. Perbedaan yang dimaksud adalah beda jumlah parameter, beda tipe data, atau beda keduanya (jumlah parameter dan tipe data).



1. Overriding

overiding adalah Bentuk polimorpism antara method yang berbeda class  
Overriding Merupakan Suatu Keadaan Dimana kelas anak dapat mengubah atau bisa kita bilang memodifikasi atau memperluas data dan method pada kelas induk

Contructor

adalah sebuah sebuah method yang menujukan bagaimana sebuah object di bentuk   
berikut sifat-sifat constructor :

1. nama constructor harus sama dengan class
2. constructor secara otomatis di jalankan saat kita membuat sebuah object
3. constructor tidak memiliki nilai kembali dan return statement
4. contructor adalah cara menetukan bagaimana sebuah class di buat

coba di ingat kembali saat anda membuat object : ada 2 step kan :

Manusia shoffi;

shoofi = new Manusia();

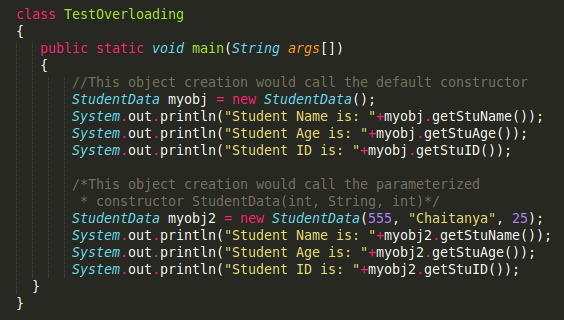
perhatikan “Manusia();”

bukan kah itu sebuah fungsi ?  
Ya itu lah contructor default dari class yang di buat oleh sistem java

tapi selain menggunakan constructor default kita juga bisa membuat constructor kita sendiri dengan bentuk yang berbeda. Nah itulah yang namanya Overloading constructor

Overloading Constructor adalah constructor yang memiliki beda bentuk(parameter) sehingga akan meng overload konstructor lainya:





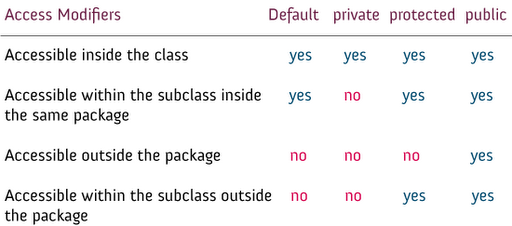
* Access Specifier

ok materi terahir untuk dasar oop untuk memahami pemrograman android ialah Accses specifiers.

Access Specifier adalah pengaturan hak akses terhadap elemen class ataupun class itu sendiri, element class dalam hal ini adalah method maupun property

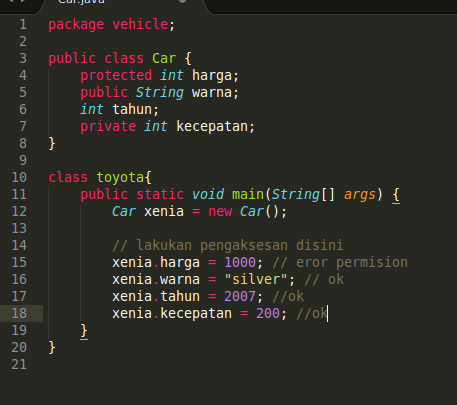
jadi ibarat sebuah rumah :   
tiap tiap ruangan nya memiliki tingkat akses yang berbeda, misalnya bukan kah sangat tidak wajar apabila ada seorang tamu yang masuk kamar tanpa seizin pemilik rumah? Seorang tamu hanya bisa bebas masuk dan keluar di halaman teras saja.

Berikut tabel spesifikasi dari Acces specifiers



buat file dengan nama Car.java

lalu compile dan perhatikan pesan eror

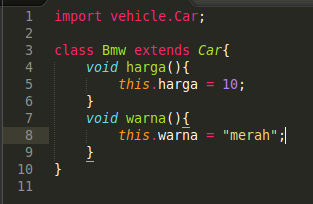


lakukan perintah

javac -d . Car.java

lalu

buat file lagi dengan nama Kendaraan.java

seperti di bawah  


ini adalah contoh pengaksesan dari luar package (import package)

class protected mengijin kan pengaksesan dari class turunan (subclass) dari package lain.