

Class Abstract dan Interface

Class Abstract

Abstract

- Class abstrak adalah class yang masih dalam bentuk abstrak.
 - Karena bentuknya masih abstrak, dia tidak bisa dibuat langsung menjadi objek.
- Abstract class adalah class yang mempunyai setidaknya satu **abstract method**.
- Abstract method adalah method yang tidak memiliki body (hanya deklarasi method).
- Abstract class tidak bisa dibuat obyeknya.
- Obyek hanya bisa dibuat dari non-abstract class (concrete class).

- Kelas abstrak menyediakan kelas dasar yang menjadi tempat perluasan kelas-kelas lain
- Kelas abstrak dapat menyediakan:
 - Metode yang diterapkan: Metode yang diterapkan sepenuhnya, dan kelas anak mereka dapat menggunakannya
 - Metode abstrak: Metode yang tidak memiliki implementasi dan memerlukan kelas anak untuk mengimplementasikannya
- Kelas abstrak tidak mengizinkan konstruksi secara langsung, tetapi Anda dapat membuat subkelas atau kelas bersarang statis untuk membuat instance mereka

Kelas abstrak

- Harus didefinisikan dengan menggunakan kata kunci abstrak
- Tidak dapat dipakai menjadi objek
- Dapat mendeklarasikan variabel lokal
- Dapat memiliki definisi dan implementasi metode
- Harus disubklasifikasikan (diperluas) daripada diimplementasikan (diimplementasikan)

Informasi Lebih Lanjut tentang Kelas Abstrak

- Jika suatu kelas konkrit menggunakan kelas abstrak, ia harus mewarisinya, pewarisan tidak termasuk pewarisan dari kelas lain
- Kelas abstrak penting ketika semua kelas diturunkan harus berbagi metode tertentu
- Alternatifnya adalah dengan menggunakan antarmuka, kemudian mendefinisikan setidaknya satu kelas konkrit yang mengimplementasikan antarmuka tersebut
- Dapat menggunakan kelas konkrit sebagai tipe objek untuk variabel di sejumlah program yang memberikan fleksibilitas lebih besar

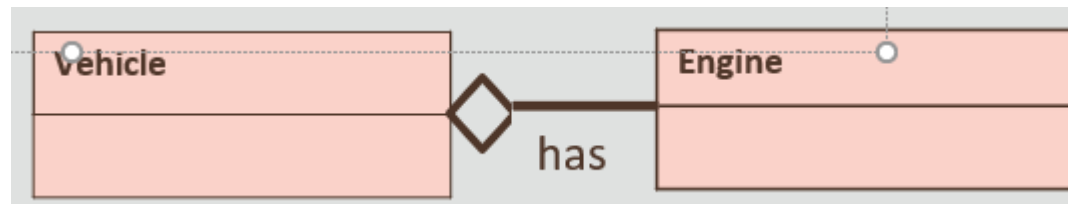
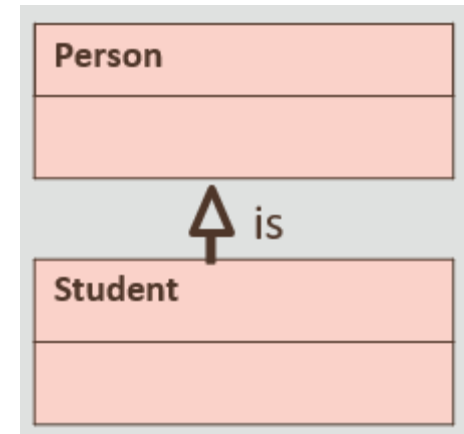
Saat Anda membuat subkelas dari kelas Abstrak, Anda dipaksa untuk mengimplementasikan metode abstraknya

Kelas Abstrak atau Antarmuka

- Tidak ada aturan baku apakah akan menggunakan Antarmuka, kelas Abstrak, atau keduanya
- Kelas abstrak biasanya memiliki hubungan yang lebih kuat antara dirinya dan kelas-kelas yang akan diturunkan darinya dibandingkan antarmuka
- Kelas dan Antarmuka dapat mengimplementasikan banyak antarmuka sedangkan satu kelas hanya dapat menjadi subkelas dari satu kelas abstrak
- Kelas abstrak memungkinkan metode untuk didefinisikan

Kelas Abstrak atau Antarmuka

- Kelas abstrak seringkali memiliki hubungan “Is-A”.
- Antarmuka sering kali memiliki hubungan “Has-A”.
 - Kendaraan memiliki Mesin, jadi Mesin mungkin lebih baik sebagai Antarmuka



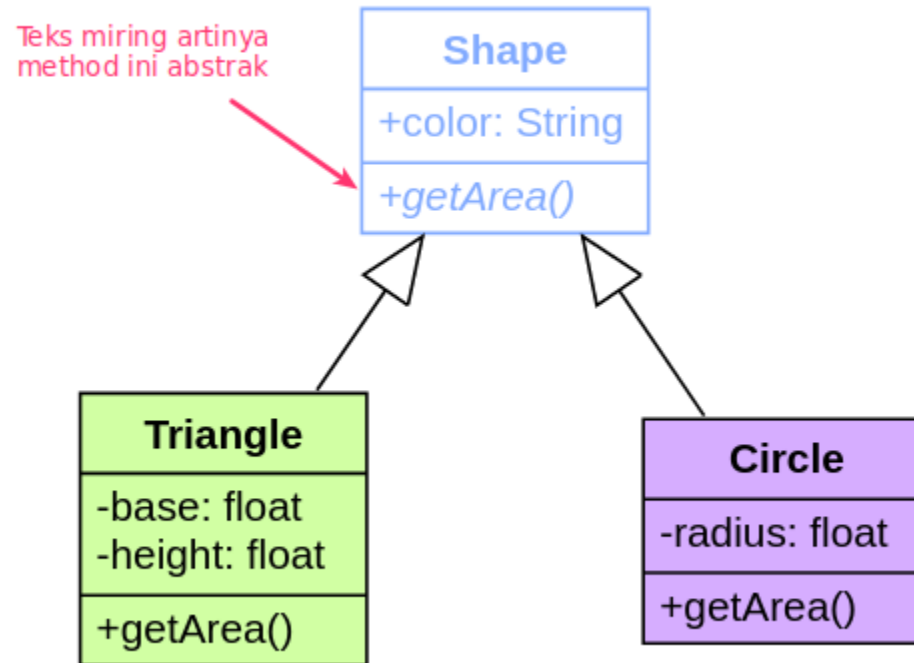
Kegunaan Class Abstract

- Class Abstract berisi beberapa method dan beberapa method abstract. Class Abstract berisi sebagian implementasi, dan subclass yang melengkapi implementasinya. Dengan kata lain Class Abstract memiliki beberapa kesamaan (Bagian yang diimplementasikan oleh subclass) dan memiliki perbedaan (method yang dimiliki sendiri) abstract)
- Deklarasikan method abstract, jika ada satu atau lebih subclass yang diharapkan mempunyai fungsionalitas yang sama tapi implementasi berbeda.
- Gunakan class abstract untuk mendefinisikan behavior secara umum sebagai superclass, sedangkan subclass menyediakan implementasi detail.
- Jika class abstract semua method merupakan method abstract, sebaiknya class abstract tersebut diganti menjadi Interface.

Contoh Class Abstract

- Kita dapat menggambar lingkaran, persegi panjang garis, kurva Bezier dan object-object graphic lainnya.
- Object tersebut mempunyai state tertentu (seperti: position, orientation, line color, fill color) dan behaviour secara umum(contoh: moveTo, rotate, resize, draw).
- Beberapa state dan behavior ini sama untuk semua object graphic contoh: position, fill color, and moveTo.
- Implementasi yang berbeda sebagai contoh resize or draw. Semua Object Graphic harus mengetahui bagaimana cara draw dan resize

Contoh Program Class Abstrak



- Class Shape adalah class abstrak, karena dia punya method abstrak `getArea()` (hitung luas).
- Jika kita membuat objek dengan class Shape dan memanggil method `getArea()`, maka si class Shape akan bingung..
- Shape atau bentuk bangun datar yang mau dihitung luasnya seperti apa?
- Rumusnya gimana?
- Karena itu, class ini kita jadikan abstrak. Soalnya belum jelas cara ngitung luasnya.
- Dan kita juga semua tahu, kalau Shape atau bangun datar pasti memiliki luas. Tapi cara ngitungnya berbeda-beda.

Interface

Jadi interface adalah antarmuka.

- Kata ini mungkin sering kamu dengar. Contohnya seperti: GUI (Graphical User Interface) dan CLI (Command Line Interface).
- Secara umum, interface berfungsi sebagai penghubung antara sesuatu yang 'abstrak' dengan sesuatu yang nyata.
 - Coba perhatikan handphone-mu, ada berapa tombol di sana?
 - Kalau HP zaman sekarang, paling cuma ada dua.. yakni tombol power dan tombol volume.
 - Tombol-tombol inilah yang dimaksud interface. Sementara isi dalam HP-nya, kamu tak peduli atau tak akan tahu seberapa kompleks dan abstrak sistem yang ada di sana.

- Pengertian umum Interface dalam konsep PBO adalah sebuah modul yang berisi kumpulan deskripsi method-method. Dalam hal ini, method-method tersebut bersifat abstrak atau tidak memiliki body.
- Interface merupakan bentuk perluasan dari kelas abstrak.
- Selain method, interface juga dapat berisi sekumpulan variable, namun variable yang dideklarasikan di dalam interface harus bersifat final(nilainya tidak dapat diubah /konstan).
- Keberadaan interface dalam sebuah kode program sebenarnya sifatnya hanyalah opsional saja.

- Sebagai contoh :

Dalam kehidupan nyata dapat diketahui ada manusia yang bekerja sebagai tentara, penyanyi, pengacara, dan sebagainya, tentunya manusia-manusia tersebut selain harus memiliki method standard sebagai seorang manusia, juga harus memiliki method yang sesuai dengan pekerjaannya.

Dengan demikian untuk membuat objek manusia yang bekerja sebagai penyanyi, harus dibuat kelas yang merupakan turunan kelas manusia yang meng-implementasikan interface penyanyi.

Perbedaan Abstract Class dan Interface

- Interface hanya bisa mendefinisikan Method, tidak bisa mengimplementasikan, sedangkan Abstract Class bisa mengimplementasikan Method
- Interface tidak boleh terdiri dari field atau property, Abstract Class bias
- Interface tidak boleh terdiri dari konstruktor dan destruktor, sedangkan Abstract class bisa:

Abstract class	Interface
Bisa berisi abstract dan non-abstract method.	Hanya boleh berisi abstract method.
Kita harus menuliskan sendiri modifiernya.	Kita tidak perlu susah2 menulis public abstract di depan nama method. Karena secara implisit, modifier untuk method di interface adalah public dan abstract .
Bisa mendeklarasikan constant dan instance variable .	Hanya bisa mendeklarasikan constant . Secara implisit variable yang dideklarasikan di interface bersifat public , static dan final .
Method boleh bersifat static .	Method tidak boleh bersifat static .
Method boleh bersifat final .	Method tidak boleh bersifat final .
Suatu abstract class hanya bisa meng- <i>extend</i> satu abstract class lainnya.	Suatu interface bisa meng- <i>extend</i> satu atau lebih interface lainnya.
Suatu abstract class hanya bisa meng- <i>extend</i> satu abstract class dan meng- implement beberapa interface.	Suatu interface hanya bisa meng- <i>extend</i> interface lainnya. Dan tidak bisa meng- <i>implement</i> class atau interface lainnya.

Apa Bedanya Class Abstrak dengan Interface?

- Class abstrak dan interface sama-sama digunakan untuk membuat abstraksi.
- Keduanya hampir memiliki sifat yang sama.
- Tapi ada bedanya.
 1. di class abstrak kita bisa buat properti atau variabel sedangkan di interface kita cuma bisa buat konstanta saja
 2. di class abstrak kita bisa implementasikan kode method seperti class biasa, sedangkan di interface harus menggunakan default
 3. di class abstrak dapat memiliki member private dan protected sedangkan interface harus publik semua
 4. Class abstrak diimplementasikan dengan pewarisan (extends) sedangkan interface menggunakan implements

Sumber

- <https://www.petanikode.com/java-oop-abstract/>
- Materi Kuliah PBO ITS Surabaya