

Nama : Lailatul Hasanah

NIM : 222011364

Kelas : 2KS4

Kode : 0424QA6

Abstract Class

- Abstract class adalah sebuah class dengan keyword abstract di depan nama class.
- Abstract method adalah sebuah method tanpa implementasi, yang dimaksudkan untuk diimplementasikan oleh subclassnya.
- Abstract class memiliki paling tidak satu abstract method.
Jika suatu class memiliki satu atau lebih abstract method, maka kelas tersebut harus dideklarasikan sebagai abstract class.
- Abstract class bisa memiliki concrete method.
- Abstract class tidak bisa diinstansiasi/dibuat objeknya.
Namun abstract class bisa digunakan sebagai referensi objek.

Interface

- Interface menentukan method apa saja yang bisa dilakukan oleh suatu class
- Interface tidak menentukan bagaimana suatu class melakukan method tersebut
- Satu class dapat meng-implements banyak interface
Namun satu class hanya bisa meng-extends satu kelas.
- Dalam deklarasi interface juga dapat dideklarasikan variable, namun secara implisit bersifat final dan static (tidak bisa diubah oleh kelas yang mengimplementasikan)
- Interface tidak dapat diinstansiasi/dibuat objeknya
Namun dapat digunakan sebagai referensi objek.

Abstract vs interface class

- Tipe method: Interface hanya bisa memiliki method abstract. Abstract class dapat memiliki method abstract maupun concrete.
- Final variabel: Variabel yang dideklarasikan di Interface adalah final. Class abstract bisa memiliki variabel non-final.
- Tipe variabel: Class abstract dapat memiliki variabel final, non-final, static dan non-static. Interface hanya memiliki variabel static and final.
- Implementasi: Class abstract dapat berisi implementasi interface. Interface tidak dapat berisi implementasi class abstract.
- Inheritance vs Abstraction: Interface dapat diimplementasikan dengan keyword “implements” dan abstract class dapat di-extend dengan keyword “extends”.

- Multiple implementations: Interface hanya dapat di-extend dari interface lainnya, abstract class dapat meng-extend satu class dan mengimplementasi banyak interface.
- Akses data: Variabel pada interface bersifat public secara default. Abstract class dapat bersifat private, protected, dll.

Gunakan Abstract Jika:

- Jika ingin menyediakan kelas dasar yang berisi fungsionalitas yang umum.
- Jika ingin mendeklarasikan member non-public.
- Jika ingin menambahkan method baru di masa depan.
Jika menambahkan method pada interface, maka seluruh class yang mengimplementasikan interface tersebut harus disesuaikan.
- Jika menginginkan forward compatibility
Menambahkan method pada abstract class tidak perlu mengubah kode seluruh kelas turunannya.

Gunakan Interface jika:

- Jika fungsionalitas umum yang diinginkan akan berguna untuk berbagai macam objek dari kelas yang berbeda-beda.
 1. Abstract class digunakan untuk kelas-kelas yang terkait dan berhubungan dekat.
 2. Interface paling baik digunakan untuk menyediakan fungsionalitas umum untuk kelas yang tidak saling berhubungan.
- Jika kita tidak berencana mengubah API dalam waktu dekat.
- Jika kita menginginkan sesuatu seperti multiple inheritance karena satu kelas dapat mengimplementasikan banyak interface.
- Jika kita mendesain fungsionalitas yang berskala kecil dan ringkas.
Gunakan abstract class jika kita ingin mendesain fungsionalitas berskala besar.