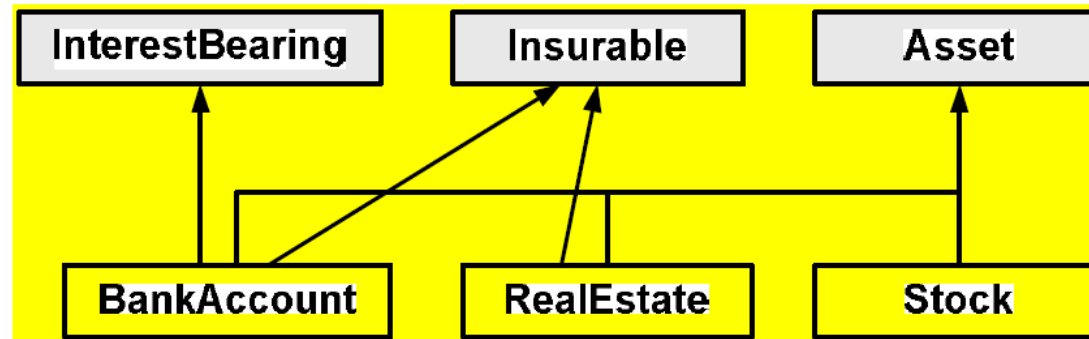


INTERFACE

Multiple Inheritance



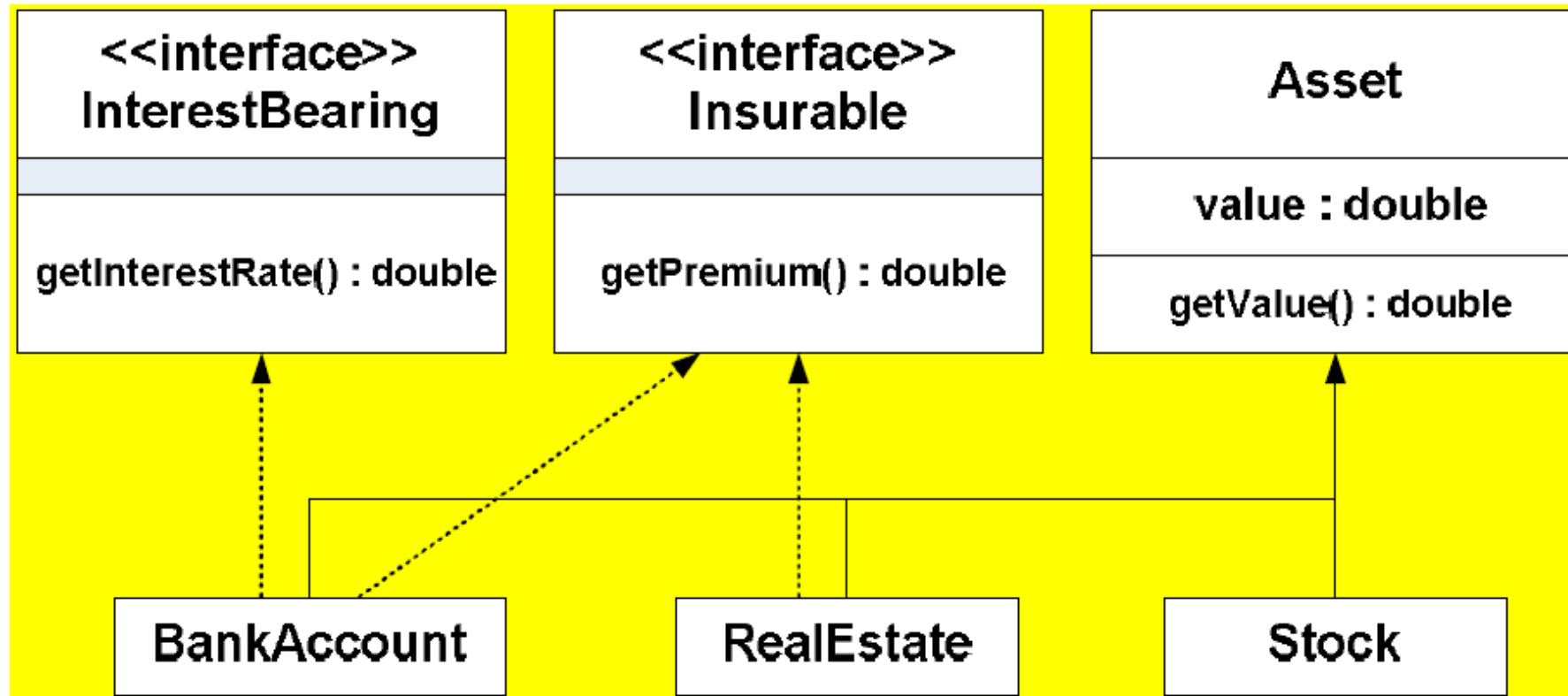
- Bank account (rekening bank) adalah jenis dari aset (asset), tetapi adalah interest bearing (berbunga) dan dapat diasuransi terhadap kehilangan.
- Masalah – suatu subclass dapat mempunyai multiple superclasses dengan perilaku dan struktur yang *overlap*.
 - Contoh:
 - Class InterestBearing menyimpan tanggal pembayaran bunga pertama.
 - Class Insurable menyimpan tanggal premi asuransi pertama di bayar
 - BankAccount diturunkan dari InterestBearing dan Insurable, jadi atribut tanggal (date) mana yang diwarisinya?

- Interface berbeda dengan class.
- Interface berisi method kosong dan konstanta.
- Method dalam interface tidak mempunyai statement.
- Sehingga deklarasi method dalam interface sama dengan deklarasi abstract method pada abstract class.
- Method yang dideklarasikan didalam interface secara otomatis adalah public dan abstract.
- Variable dalam interface secara otomatis adalah public, static, dan final.

Contoh Interface

```
public interface Relation {  
    public boolean isGreater( Object a, Object b);  
    public boolean isLess( Object a, Object b);  
    public boolean isEqual( Object a, Object b);  
}
```

Hirarki Penggunaan Interface



```
public abstract class Asset {  
    protected double value;  
  
    public double getValue() {  
        return value;  
    }  
}
```

```
public interface Insurable {  
    public abstract double getPremium();  
}
```

```
public interface InterestBearing {  
    public abstract double getInterestRate();  
}
```

```
public class RealEstate extends Asset implements
Insurable {
    public RealEstate(double v) {
        value=v;
    }

    public double getPremium() {
        // insurance premium 5% dari value
        return value * 0.05;
    }
}
```

```
public class BankAccount extends Asset implements
InterestBearing, Insurable {
    public BankAccount(double v) {
        value=v;
    }

    public double getInterestRate() {
        return 5.2;
    }

    public double getPremium() {
        // insurance premium 1% dari value
        return value*0.01;
    }
}
```

Mengimplementasikan Interface

- Bila sebuah class mengimplementasikan suatu interface, maka semua konstanta dan method interface akan dimiliki oleh class ini.
- Method pada interface harus diimplementasikan pada class yang mengimplementasikan interface ini.
- Bila class yang mengimplementasikan interface tidak mengimplemetasikan semua method dalam interface, maka class tersebut harus dideklarasikan abstract.

Inheritance pada Interface

- Kita bisa membuat subinterface dengan menggunakan kata extends.
- Satu class boleh mengimplementasikan lebih dari satu interface.
- Suatu interface boleh mengextends lebih dari satu interface.

Inheritance pada Interface

- Interface bukan bagian dari hirarki class
- Namun interface dapat mempunyai relasi inheritance

```
public interface PersonInterface {  
    void doSomething();  
}  
  
public interface StudentInterface extends PersonInterface {  
    void doExtraSomething();  
}
```

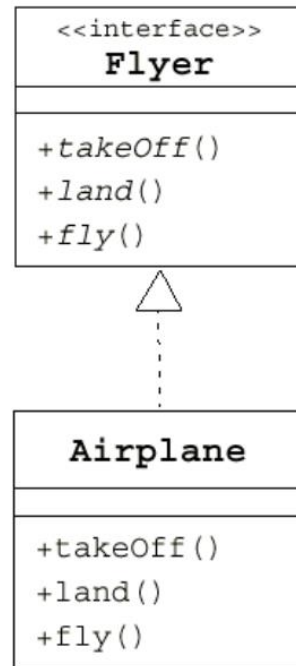
Mengimplementasikan Multiple Interface

- Satu class boleh mengimplementasikan lebih dari satu interface.
- Bila suatu class akan dijadikan subclass dan akan mengimplementasikan interface, maka kata extends harus lebih dulu dari implements.

Kegunaan Interface

- Semua class yang mengimplementasikan sebuah interface tertentu berarti class-class tersebut mengimplementasikan methods yang sama dengan kata lain class-class tersebut mempunyai fungsionalitas yang sama.

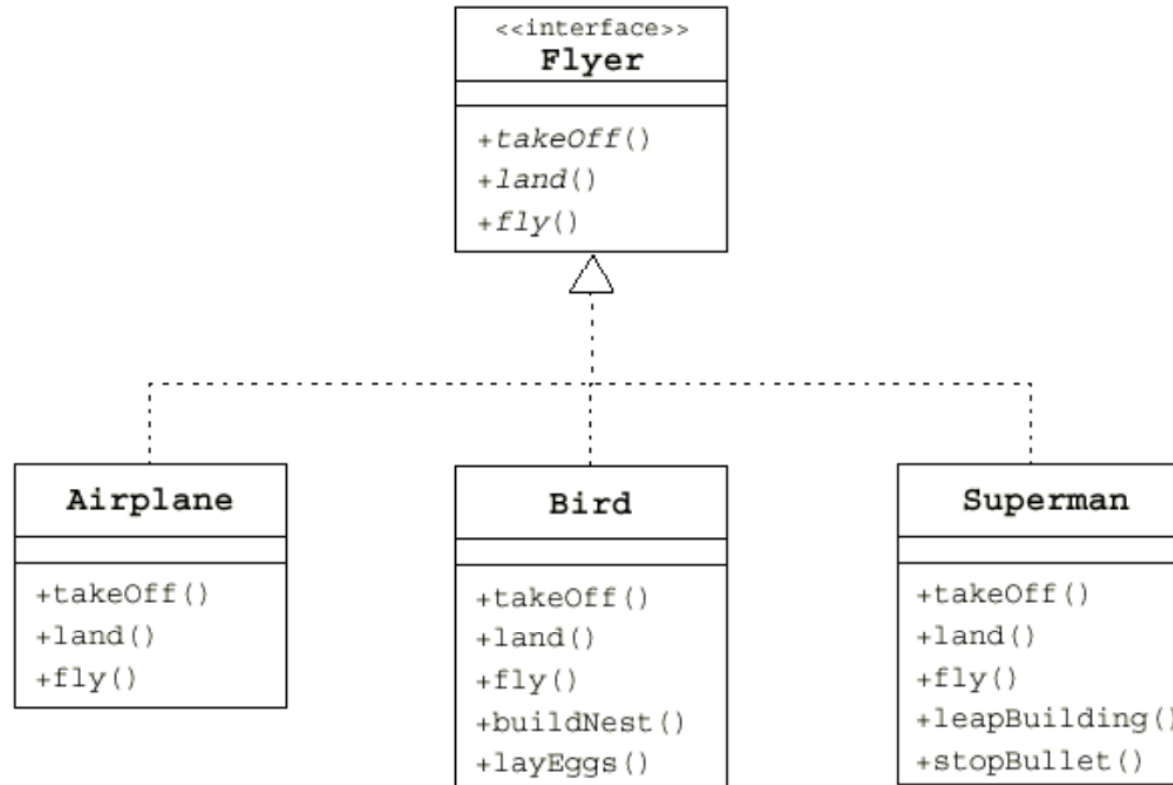
Interface Flyer dan Airplane implementation



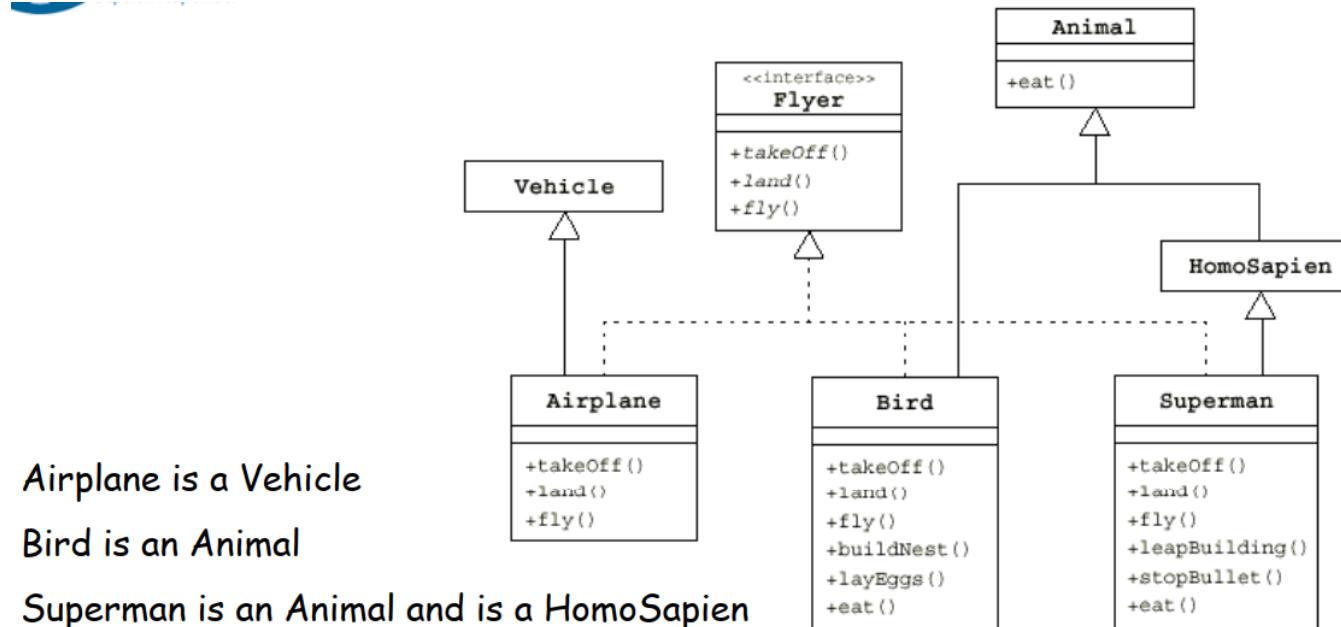
```
public interface Flyer {
    public void takeOff();
    public void land();
    public void fly();
}

public class Airplane implements Flyer {
    public void takeOff() {
        // accelerate until lift-off
        // raise landing gear
    }
    public void land() {
        // lower landing gear
        // decelerate and lower flaps until touch-down
        // apply breaks
    }
    public void fly() {
        // keep those engines running
    }
}
```

Multiple Implementation of the Flyer Interface



Gabungan Inheritance dan Implementation



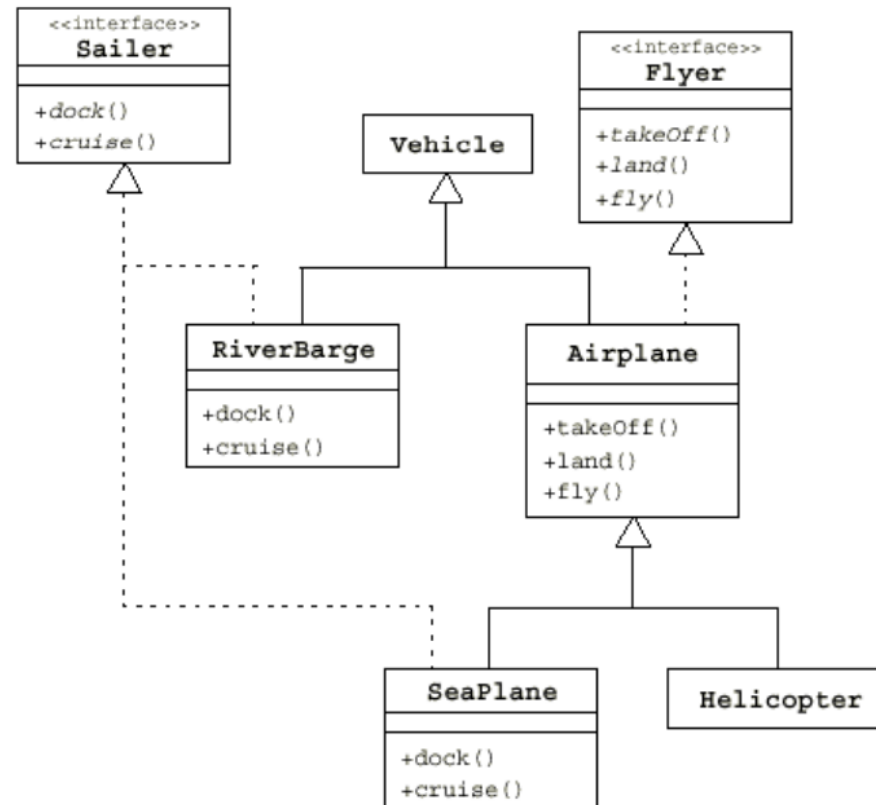
Airplane is a Vehicle

Bird is an Animal

Superman is an Animal and is a HomoSapien

Kelemahan multiple inheritance adalah suatu class bisa mewarisi method dari lebih dari satu class dimana method ini tidak diharapkan. Dengan Interface maka hal ini bisa dihindari.

Example: Multiple Interface



Rangkuman

Class Abstrak

- Dapat menyediakan implementasi satu atau lebih metode
- *Involve inheritance* – suatu subclass yang mewarisi suatu abstract class harus meng-override semua metode abstrak (dari superclass)
- Suatu abstract class dapat menyediakan atribut variabel dan konstan.

Interface

- Tidak pernah menyediakan detail implementasi. Semua metode bersifat abstrak secara implisit (tidak langsung)
- *Involve a contract* - suatu class yang mengimplementasikan interface harus menyediakan implementasi bagi setiap metode
- Suatu interface boleh hanya menyediakan atribut-atribut konstan, atribut yang dideklarasikan **static** dan **final**.