

Programació

UT8.4 Herència per implementació

Concepte

- L'herència per implementació es basa en la realització de les conegudes com a *interfícies*.
- Les interfícies són classes abstractes pures que no contenen cap atribut ni la definició de cap mètode, només contenen mètodes abstractes.
- Poden ser pares de classes o d'altres interfícies

Definició d'una interfície

```
interface <interfazBase> {  
    <cabeceraMetodo1>;  
    ...  
    <cabeceraMetodoN>;  
}  
  
[abstract] class <claseDerivada> implements <interfazBase> {  
    ...  
}  
  
interface <interfazDerivado> extends <interfazBase> {  
    ...  
}
```

on tots els mètodes de la interfície són públics i abstractes

```
public interface Comestible {  
    /** Describe cómo se come */  
    public abstract void comoSeCome();  
}
```


Limitacions de les interfícies

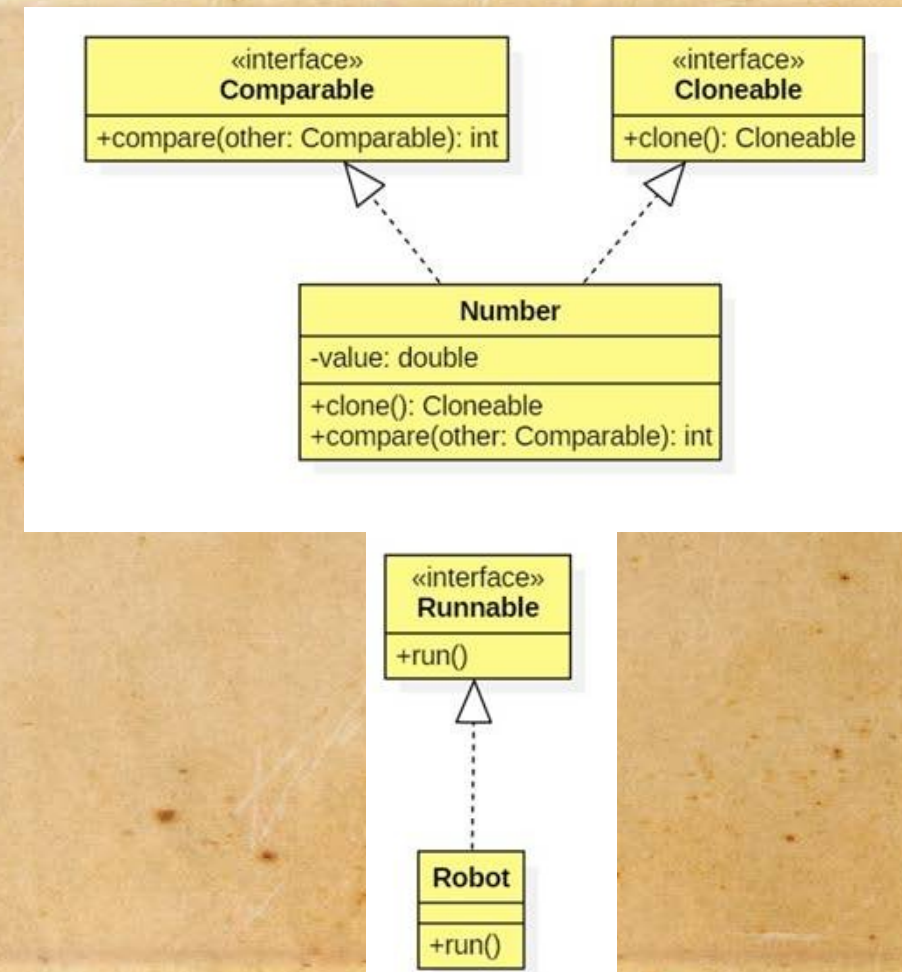
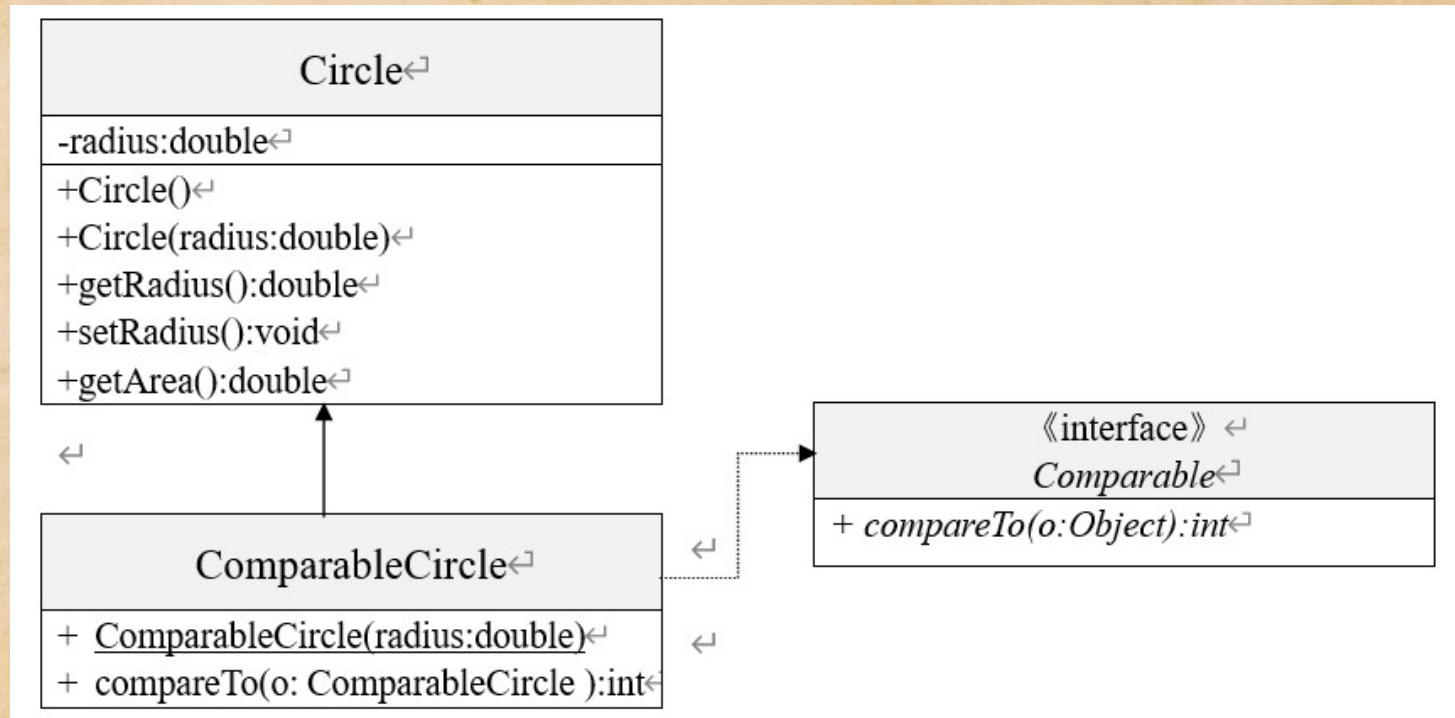
- Es fa ús de les interfícies més o menys de la mateixa manera que d'una classe abstracta
- Per tant, es pot utilitzar una interfície com un tipus de dada per una referència en una variable.
- Tampoc es poden instanciar interfícies, de la mateixa manera que passa amb les classes abstractes.

Exemple Comestibles i Animals

- Es pot utilitzar el interfície `Comestible` per especificar si un objecte concret és comestible.
- Això s'aconsegueix deixant a la classe que implemente eixa interfície. Per això es fa ús de la paraula reservada `implements`
- Per exemple, les classes `Pollastre` i `Fruita` implementen esta interfície.
- La relació entre la classe i la interfície es coneix com herència per implementació.

Descàrrega el projecte "Interficies.zip"

Exemple interfície Comparator



Quina definició és correcta?

```
interface A {  
    void print() { };  
}
```

(a)

```
abstract interface A extends I1, I2 {  
    abstract void print() { };  
}
```

(b)

```
abstract interface A {  
    print();  
}
```

(c)

```
interface A {  
    void print();  
}
```

(d)