

Il Linguaggio Java

Le interfacce

Ordinamento dei conti



- **PROBLEMA:** si vogliono ordinare i libretti di risparmio (**SavingsAccount**) in base al loro tasso di interesse
- **REQUISITO:** La classe dovrebbe essere *ordinabile*, cioè gli oggetti della classe devono essere *confrontabili* tra di loro



UN POSSIBILE APPROCCIO

1. Derivare **SavingsAccount** da una "superclasse" **Comparable** in modo che gli oggetti siano confrontabili

```
public class SavingsAccount extends BankAccount,  
                                Comparable {  
    // ...  
}
```

2. Utilizzare il metodo **Arrays.sort**, che ordina oggetti confrontabili
 - Sfortunatamente questo approccio non funziona perché in Java l'ereditarietà multipla **non è** permessa

Interfacce



- Un'interfaccia definisce un **protocollo di comportamento** che può essere **implementato** da **qualunque** classe nella gerarchia delle classi
- Un'interfaccia definisce un insieme di metodi ma non li implementa
- Una classe che implementa l'interfaccia implementa tutti i metodi facendo proprio il comportamento definito dall'interfaccia

Differenze tra classe (astratta) ed un'interfaccia



- Un'interfaccia non può avere variabili istanza
- Tutti i metodi di un'interfaccia sono astratti
- Tutti i metodi dell'interfaccia sono automaticamente pubblici
- Un'interfaccia non si estende ma si *implementa*
- Una classe può estendere solo una superclasse ma può implementare molte interfacce
- Un'interfaccia non fa parte della gerarchia delle classi; classi scorrelate possono implementare la stessa interfaccia

L'interfaccia Comparable



```
public interface Comparable {  
    int compareTo(Object other); // public di default  
}
```

- L'interfaccia **Comparable** (`java.lang`) permette di imporre un *ordine naturale* tra gli oggetti di una classe
- L'unico metodo **compareTo** viene detto il *metodo naturale di confronto*

Il metodo **compareTo** confronta questo oggetto con l'oggetto **other** e ritorna un valore negativo, il valore zero o un valore positivo se questo oggetto è minore, uguale o maggiore di **other**



```
public class SavingsAccount extends BankAccount
                               implements Comparable {
    public int compareTo(Object other) {
        SavingsAccount o = (SavingsAccount)other;
        if ( interestRate < o.interestRate ) return -1;
        if ( interestRate > o.interestRate ) return 1;
        return 0;
    }
    // ...
}
```

Il metodo **sort** di **Arrays**



- Il metodo **public static void sort(Object[] a)** ordina l'array **a[]** di oggetti in ordine crescente, secondo l'ordine naturale degli elementi. Cioè,
- gli elementi dell'array **a[]** devono
 - implementare l'interfaccia **Comparable**
 - essere mutuamente confrontabili

Ordinamento dei conti (II)



```
import java.util.Arrays;
import java.util.Random
//...
Random rand = new Random();
SavingsAccount[] s = new SavingsAccount[rand.nextInt(100)];

// inizializzazione di s[i]
for (int i = 0; i < s.length; i++)
    s[i] = new SavingsAccount(rand.nextInt(10));

// Ordinamento dei conti
Arrays.sort(s);
```

Proprietà delle interfacce (I)

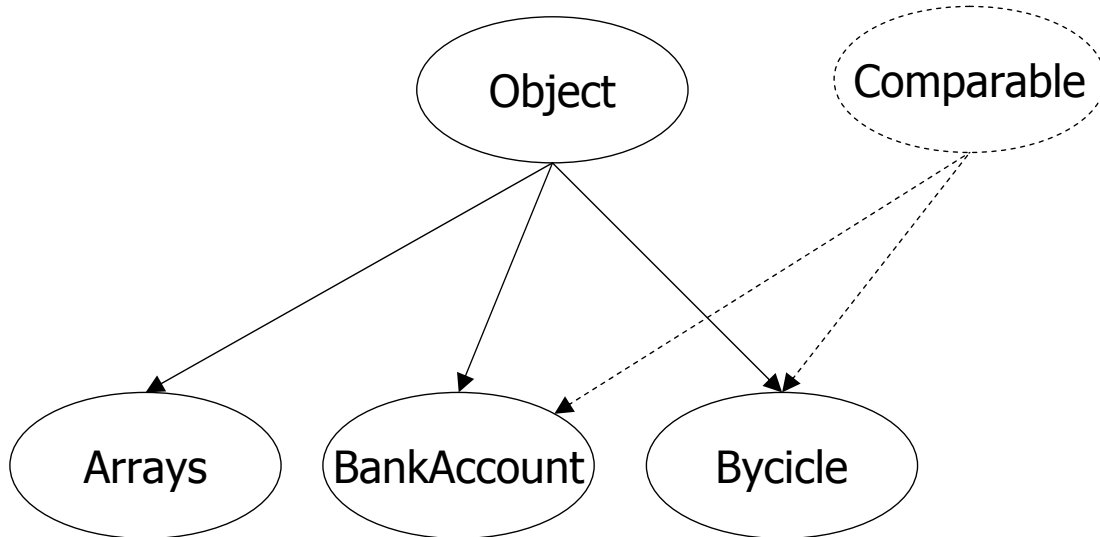


- Una classe può avere una sola superclasse ma può implementare qualsiasi numero di interfacce

```
public class SavingsAccount extends BankAccount
    implements Comparable, Cloneable {
    // ...
}
```

- Un'interfaccia non è una classe per cui non è possibile crearne oggetti istanza;

```
Comparable x; // Riferimento interfaccia: OK
x = new Comparable(); // Errore di compilazione
```



Le interfacce evitano di forzare relazioni tra classi che logicamente non sono richieste o necessarie

Riferimenti e conversioni



- Sono ammessi riferimenti interfaccia che si riferiscono ad oggetti delle classi che implementano l'interfaccia
`Comparable x = new SavingsAccount(10);`
- Quando una classe implementa un'interfaccia è possibile convertire un riferimento di classe in un riferimento di interfaccia
`SavingsAccount s = new SavingsAccount();
Comparable c = s;`
- È possibile passare un riferimento di classe a metodi che si aspettano riferimenti di interfaccia



- Un'interfaccia non può avere variabili istanza ma può avere costanti che verranno ereditate da tutte le classi che implementano l'interfaccia

```
public interface MiaInterfaccia {  
    int pippo(int a);  
    int COST_1 = 1;  
    int COST_2 = 2;  
}
```

- Le variabili di un interfaccia sono automaticamente **public static final**