

Come copiare oggetti

Programmazione in rete e laboratorio

Quando si esegue una assegnazione su una variabile oggetto, nella variabile viene copiato il riferimento all'oggetto assegnato.

Se si vuole eseguire una copia dell'oggetto, si può utilizzare il metodo `clone()` ereditato da **Object**.

```
Employee e1 = new Employee (...);    e1 e e2 fanno riferimento  
Employee e2 = e1;
```

```
Employee e1 = new Employee (...);  
Employee e2 = e1.clone();
```

Il metodo `clone` crea un nuovo oggetto inizialmente identico a quello riferito da `e1`.

clone

2

Il metodo `clone` di **Object** è **protected**: può essere usato da ogni sottoclasse di **Object** (ossia da ogni classe), ma non può essere invocato sugli oggetti della classe a meno che non venga ridefinito come **public**.

```
Employee e1 = new Employee (...);  
Employee e2 = e1.clone();
```

dà errore perché il metodo `clone` di **Employee** non è pubblico.

Occorre modificare la classe **Employee**:

```
public class Employee {  
    .....  
    public Object clone() {  
        return super.clone();  
    }  
}
```

clone

3

Questo però non è sufficiente: per poter clonare un oggetto occorre specificare che l'oggetto implementa l'interfaccia **Cloneable**.

```
public class Employee implements Cloneable {  
    .....  
    public Object clone() {  
        return super.clone();  
    }  
}
```

L'interfaccia **Cloneable** non contiene nessun metodo (nemmeno `clone`). Serve solo come etichetta per indicare che è possibile clonare gli oggetti della classe che implementa l'interfaccia.

clone

4

Questo non basta ancora: il metodo `clone` di **Object** può lanciare una eccezione **CloneNotSupportedException**, se la classe non implementa **Cloneable**.

Dato che questa eccezione non è una **RuntimeException**, occorre catturarla esplicitamente o segnalarla con **throws**.

```
public class Employee implements Cloneable {  
    .....  
    public Object clone()  
        throws CloneNotSupportedException {  
        return super.clone();  
    }  
}
```

clone

5

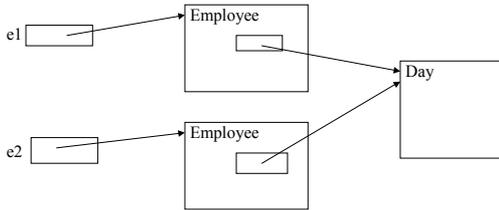
Riassumendo: per poter copiare gli oggetti di una classe occorre

1. Ridefinire il metodo `clone` come **public**.
2. Dichiarare che la classe implementa l'interfaccia **Cloneable**.
3. Gestire l'eccezione **CloneNotSupportedException**.

clone

6

Il metodo *clone* esegue una copia bit a bit dell'oggetto. Se qualche campo contiene un riferimento ad un altro oggetto, *clone* copia solo il riferimento senza clonare questo oggetto.



```
Employee e1 = new Employee (...);  
Employee e2 = e1.clone();
```

clone

7

Il metodo *clone* di *Employee* deve provvedere a copiare l'oggetto *Day*.

```
public class Employee implements Cloneable  
{  
    .....  
    public Object clone()  
    {  
        try  
        {  
            Employee e = (Employee)super.clone();  
            e.hireDay = hireDay.clone();  
            return e;  
        } catch (CloneNotSupportedException e)  
        {  
            return null;  
        }  
    }  
}
```

Naturalmente *Day* deve fornire il metodo *clone*.

clone

8