Metodologie di Programmazione

Lezione 18: introduzione a interfacce ed eccezioni

Lezione 18: Sommario

UNITELMA SAPIENZA
SAPIENZA UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

- La parola chiave final per classi e metodi
- Introduzione a interfacce ed eccezioni
- Esercizi

Metodi e classi final

- Ricordate? con la parola chiave abstract **obblighiamo** i programmatori a implementare certi metodi

- La parola chiave final ci permette di fare il contrario: **impedire** ad altri programmatori di:
  - Creare sottoclassi (se specificato di fronte a class)
  - Reimplementare (=sovrascrivere) certi metodi (di fronte all'intestazione del metodo)

Metodi final: un esempio (1)

- Supponete di creare un conto corrente "sicuro"
- Accessibile solo tramite la giusta password

```
public class ContoCorrenteSicuro extends ContoCorrente
{
      public       boolean controllaPassword(String password)
      {
            // verifica la password
            // ...
      }
}
```

- Tuttavia, estendendo la classe, possiamo "gabbare" l'utente che la userà e accedere noi al (

```
public class ContoCorrenteSicuroFidatiDiMe extends ContoCorrenteSicuro
{
      public       boolean controllaPassword(String password)
      {
            return true;
      }
}
```

Metodi final: un esempio (2)

- Possiamo evitare questo problema specificando il metodo final

```
public class ContoCorrenteSicuro extends ContoCorrente
{
      public final boolean controllaPassword(String password)
      {
            // verifica la password
            // ...
      }
}
```

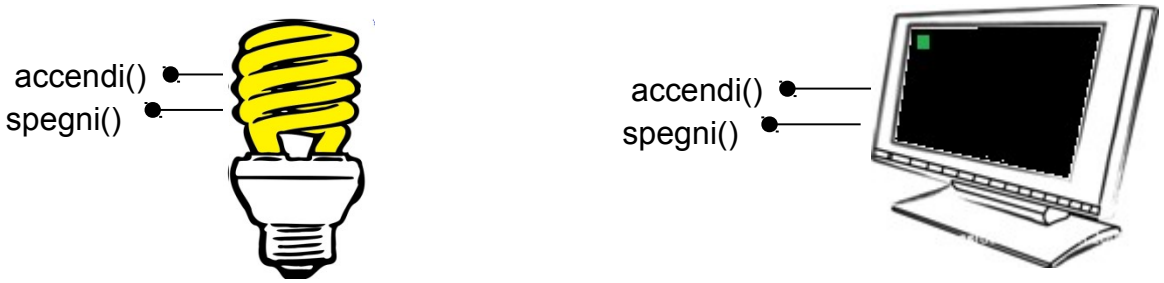
Nessuno può sovrascrivere questo metodo!

Classi final: alcuni esempi

- Alcune classi della libreria Java standard sono immutabili:
  - La classe String
  - Tutti i wrapper delle classi primitive (Integer, Character, Long, ecc.)
  - java.awt.Color

Un “assaggino” di Interfacce (1)

- Quali sono le operazioni essenziali che svolge un oggetto?
- Per esprimere tali operazioni “comuni” (che si applicano a classi differenti) si utilizzano le interfacce



- Si definisce un’interfaccia con la parola chiave interface

```
public interface Accendibile
{
    void accendi();
    void spegni();
    boolean acceso();
}
```

I metodi di un’interfaccia sono automaticamente pubblici

Un “assaggino” di Interfacce (2)

- Una volta definita l’interfaccia

```
public interface Accendibile
{
    void accendi();
    void spegni();
    boolean acceso();
}
```

le classi che intendono implementare quell’interfaccia usano la parola chiave implements:

```
public class Lampadina implements Accendibile
{
    private boolean bAcceso;

    public void accendi()
    {
        bAcceso = true;
    }

    public void spegni()
    {
        bAcceso = false;
    }

    public boolean acceso() { return bAcceso; }
}
```

Un “assaggino” di Eccezioni (1)

- Un’eccezione è una classe le cui istanze vengono utilizzate per segnalare una condizione che **impedisce** la normale prosecuzione del programma
- Ad esempio:
  - Accesso oltre i limiti di un array (ArrayIndexOutOfBoundsException)
  - Divisione per zero (ArithmeticException)
  - Errore nel formato del numero (NumberFormatException)
  - Errore di input/output (IOException)
  - La costante enumerativa passata a valueOf non esiste (IllegalArgumentException)
- Potete definire anche le vostre eccezioni personalizzate

```
public class NonToccareLaMiaRobaException extends Exception
{
}
```

Un “assaggino” di Eccezioni (2)

- Potete definire anche le vostre eccezioni personalizzate

```
public class NonToccareLaMiaRobaException extends Exception
{
}
```

e “lanciarle” dall’interno dei metodi di una classe con la parola chiave throw

```
public class Armadietto
{
    private Sportivo proprietario;

    public void apriArmadietto(Sportivo g) throws NonToccareLaMiaRobaException
    {
        if (!proprietario.equals(g)) throw new NonToccareLaMiaRobaException();
    }
}
```

I metodi che lanciano eccezioni devono dichiararlo nell’intestazione (parola chiave throws)