Corso di Laurea Ingegneria Informatica

Fondamenti di Informatica 1

Dispensa 8

Tipi riferimento e stringhe

A. Miola

Novembre 2006

Contenuti

- Tipi riferimento
- Stringhe e oggetti String
- Stringhe e tipi primitivi
  - conversione da tipi primitivi a stringhe
  - conversione da stringhe e tipi primitivi
  - il metodo charAt
- Riferimenti e uguaglianza
- Il letterale null
- Classi per istanziare oggetti
- La classe Numero

Prerequisiti

Questo capitolo presuppone la conoscenza degli argomenti già trattati nelle precedenti lezioni di questo corso, con particolare riferimento al capitolo 3 e al capitolo 11 del libro di testo

Tipi riferimento e stringhe

- **I tipi riferimento sono tipi definiti in corrispondenza a classi**
  - **questo capitolo presenta le principali caratteristiche dei tipi riferimento**
  - **in particolare, si riprende il tipo delle stringhe, già visto in precedenza con il tipo riferimento String, e vengono introdotti altri tipi riferimento con la definizione di nuove classi**

Tipi riferimento

- **I tipi riferimento sono tipi definiti in corrispondenza a classi**
  - dal punto di vista dell'utente di un tipo riferimento (classe)
    - il dominio è l'insieme dei possibili oggetti che possono essere istanziati dalla classe
    - le operazioni associate al tipo riferimento sono le operazioni (metodi) definite dalla classe
  - dal punto di vista del linguaggio di programmazione, ai tipi riferimento possono essere applicati i seguenti operatori
    - l'operatore new – per la creazione di un nuovo oggetto
    - l'operatore . – operatore punto per l'invocazione di un metodo o l'accesso a una variabile di un oggetto
- **È sempre possibile associare alle espressioni un tipo e quindi anche un tipo riferimento**

Tipi riferimento – A cosa servono . . .

- A volte è importante introdurre un nuovo tipo riferimento attraverso la definizione di una nuova classe per oggetti istanza
- Ciò può accadere anche quando si devono trattare informazioni che apparentemente potrebbero essere definite e gestite attraverso un tipo primitivo
- La definizione di un nuovo tipo riferimento è motivata da ragioni di qualità del software prodotto, e specificamente di correttezza

. . . Tipi riferimento – A cosa servono

- Ricordando che i problemi che affrontiamo vengono descritti in termini di oggetti, si tratta di definire oggetti che sono caratterizzati dal loro nome, stato e comportamento
- Ciò che interessa ai fini della soluzione di un problema è limitato a questa caratterizzazione
- Per la definizione di ciascun oggetto che interessa applichiamo un processo di astrazione che, nascondendo tutti i dettagli realizzativi, ci permette di interagire con l'oggetto solo attraverso la sua interfaccia

Un esempio – Il contatore . . .

- Supponiamo di dover far uso di variabili intere il cui unico scopo sia quello di contare una serie di eventi o di occorrenze di dati specifici
- Anche se queste variabili sono a tutti gli effetti di tipo intero, sulle quali in teoria potremmo fare tutte le operazioni aritmetiche, esse servono a contare e quindi su di esse l'unica operazione sarà quella di incremento
- Dovremmo quindi applicare un'astrazione sul dato, assicurandoci che quel dato sia trattato esclusivamente con uno specifico e limitato insieme di operazioni
- Si supera così il concetto di variabile intera astraendo da esso e focalizzandosi sulla categoria astratta di variabile contatore

. . . Un esempio – Il contatore . . .

Frammento di codice con una variabile di tipo int

```
. . .
int cont = 0;
if (... succede qualcosa ...)
   cont++;
. . .
/* Si vuole verificare se il contatore
 * ha raggiunto un valore massimo max */
if (cont == max)
   ... fai qualcosa ... ;
. . .
```

. . . Un esempio – Il contatore . . .

Frammento di codice con una variabile di tipo contatore

```
class Contatore {
  /* Variabile d'istanza - valore del numero */
      private int valore;
  /* Costruttore – Crea un nuovo Contatore con valore iniziale 0 */
      public Contatore() {
                      this.valore = 0; }
  /* Metodi – accesso e modifica valore */
      public int getValore() {
             int n;
             n = this.valore;
             return n; }
      public void setValore(int n) {
             this.valore = n ; }
  /* Metodo – Incrementa Contatore */
      public void incrCont() {
             setValore(getValore() + 1); }
}
```

. . . Un esempio – Il contatore

Frammento di codice per l'uso di un oggetto di tipo contatore

```
. . .

Contatore cont;
cont = new Contatore();
if (... succede qualcosa ...)
   cont.incrCont();
. . .
if (cont.getValore() == max)
   ... fai qualcosa ... ;
. . .
```

Esercizio

- Modificare la classe Contatore inserendo nel metodo setValore un opportuno controllo su un valore massimo ammissibile per lo stato dell'oggetto Contatore e le conseguenti segnalazioni di errore nel caso si tenti di superare tale valore massimo

Visualizzazione e lettura di stringhe

- Per visualizzare sullo schermo il valore di una variabile s di tipo String

```
System.out.println(s);
```

- Per leggere una stringa dalla tastiera e assegnarla ad una variabile s di tipo String

```
s = Lettore.in.leggiLinea();
```

Stringhe e tipi primitivi

- **Sono stati mostrati finora solo metodi per la gestione di stringhe che permettono di calcolare nuove stringhe**
  - **Nei paragrafi dal 12.5 al 12.10 del capitolo 12 del libro di testo vengono mostrati alcuni metodi che correlano le stringhe con i tipi primitivi già visti in precedenza**

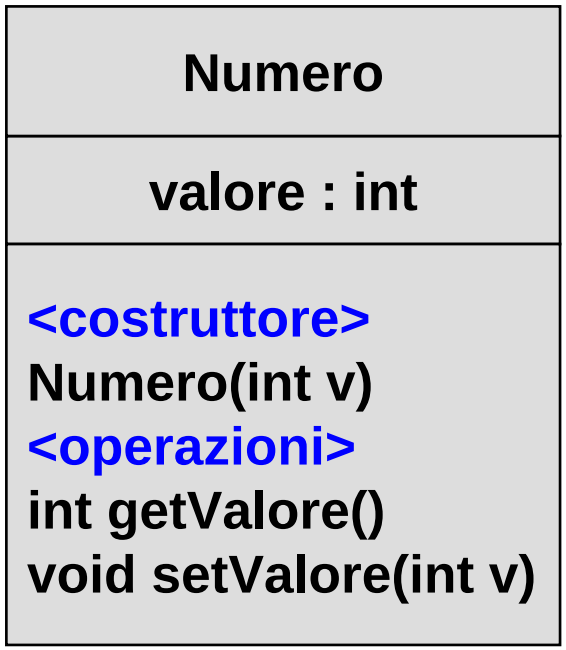
Classi per istanziare oggetti

- **Come si definisce una classe per istanziare oggetti?**
  - progetto (descrizione) di un oggetto
  - dichiarazione di **variabili d'istanza**
    - per rappresentare lo stato (le proprietà)
  - definizione di **costruttori**
    - per implementare le modalità di costruzione dalla classe, in particolare inizializzando le variabili d'istanza
  - definizione di **metodi d'istanza**
    - per implementare le operazioni degli oggetti

Una classe per istanziare oggetti

- **Si vuole definire la classe Numero per istanziare oggetti**
  - un oggetto Numero rappresenta un numero intero
  - lo stato di un oggetto Numero, definito attraverso le variabili d'istanza, consiste nel valore del numero intero
  - per creare un oggetto Numero bisogna specificare il suo valore iniziale con un metodo costruttore
  - un oggetto Numero deve saper eseguire le operazioni
    - int getValore() — restituisce il valore del numero
    - void setValore(int v) — cambia il valore del numero con il valore v

La classe Numero . . .



Questo argomento è trattato nel paragrafo 3.2.6 – Una classe per istanziare oggetti del capitolo 3 del Libro di testo

Cosa abbiamo visto finora

- Tipi riferimento e classi
- Altri metodi della classe String
- Il rapporto tra stringhe e tipi primitivi
- Come verificare l'uguaglianza tra oggetti
- Il significato del letterale null
- Come definire classi per istanziare oggetti
- Un esempio: la classe Numero

Riferimenti al libro di testo

- **Per lo studio di questi argomenti si fa riferimento al libro di testo, e in particolare ai seguenti paragrafi del capitolo 12**
  - **Dal 12.5 al 12.10**
- **L'esempio della classe Numero è trattato nel paragrafo**
  - **3.2.6 – Una classe per istanziare oggetti**

  del capitolo 3