Programmazione Orientata agli Oggetti in Linguaggio Java

Sintassi e Semantica: UML

versione 2.2

G. Mecca – Università della Basilicata – mecca@unibas.it

Sommario

- Diagrammi UML
  - ⇨Diagrammi delle Classi
  - ⇨Diagrammi di Collaborazione
  - ⇨Diagrammi di Sequenza
  - ⇨Diagrammi degli Oggetti
- Casi d'Uso

Diagrammi UML

- In questo argomento del corso
  - ⇨abbiamo visto vari esempi
  - ⇨per ciascun esempio abbiamo visto vari diagrammi di documentazione
- Diagramma di documentazione
  - ⇨rappresentazione grafica di alcuni aspetti del funzionamento di un'applicazione
  - ⇨ce ne possono essere vari e a vari livelli di approfondimento

Diagrammi UML

- Per la programmazione a oggetti
  - ⇨esiste uno standard consolidato per i diagrammi di documentazione
- Uniform Modeling Language (UML)
  - ⇨sviluppato da Grady Booch, Ivar Jacobson, James Rumbaugh
  - ⇨promosso dall'OMG (Object Management Group) e dalla Rational (ora IBM)
  - ⇨la versione iniziale (1.0) risale al 1997

Diagrammi UML

- Cos'è in concreto UML ?
  - ⇨una "sintassi grafica" per descrivere il codice
  - ⇨orientato alle applicazioni a oggetti
- Le idee alla base di UML
  - ⇨un sistema software è una cosa complessa, con molti aspetti diversi
  - ⇨bisogna utilizzare molti diagrammi per descrivere tutti questi aspetti

Diagrammi UML

ATTENZIONE
alla filosofia di UML

- Il principio fondamentale
  - ⇨un diagramma UML non descrive MAI tutti gli aspetti dell'applicazione
- In altri termini, un diagramma UML
  - ⇨si concentra su alcuni aspetti e può volutamente ometterne altri
  - ⇨affronta la descrizione dell'applicazione da un punto di vista diverso

Diagrammi UML

- La sintassi grafica
  - ⇨UML propone 9 tipi di diagrammi diversi
  - ⇨per ciascun diagramma c'è un insieme di simboli grafici molto ricco
- Ma...
  - ⇨non necessariamente devono essere utilizzati tutti i diagrammi
  - ⇨in un diagramma non è necessario utilizzare tutti i simboli grafici (un diagramma può essere volutamente "superficiale" in alcuni aspetti e concentrarsi su altri)

Diagrammi UML

- Due tipologie di diagrammi
- Diagrammi statici
  - ⇨diagramma delle classi
- Diagrammi dinamici
  - ⇨diagramma di collaborazione
  - ⇨diagramma di sequenza
  - ⇨diagramma degli oggetti

Diagrammi delle Classi

- Diagrammi statici
  - ⇨diagrammi che descrivevano la struttura del codice delle classi e le loro relazioni di dipendenza
  - ⇨il principale è il diagramma delle classi
- Diagramma delle Classi
  - ⇨diagramma statico che descrive l'organizzazione delle classi dell'applicazione

Diagrammi delle Classi

- Classe
  - ⇨rettangolo con nome, proprietà e metodi
- Package
  - ⇨"cartelle" che contengono le classi
- Relazioni tra le classi
  - ⇨vengono rappresentate attraverso frecce
  - ⇨ce ne sono di vari tipi

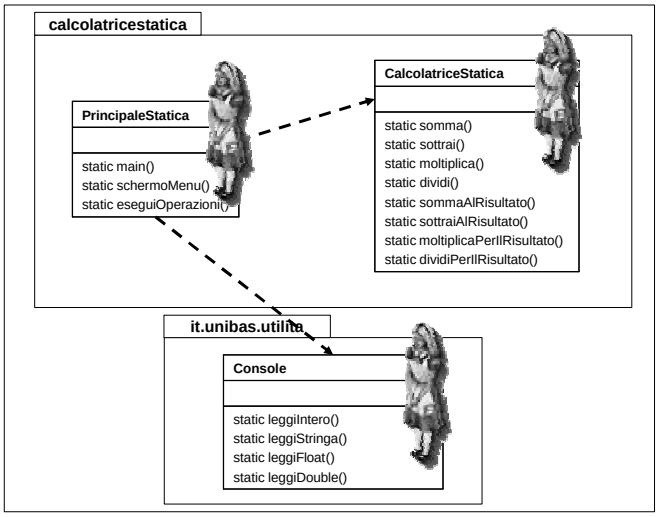
Diagrammi delle Classi

- Principali relazioni
  - ⇨relazione di dipendenza
  - ⇨relazione di associazione
  - ⇨relazione di ereditarietà (o di estensione)
- Relazione di dipendenza
  - ⇨descrive il fatto che nel codice di una classe A sono contenute chiamate al codice di un'altra classe B
  - ⇨freccia tratteggiata che va da A a B

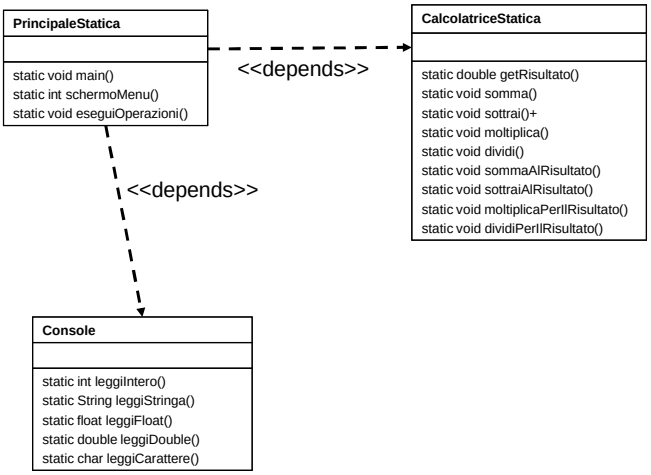
Diagrammi delle Classi

- Varianti della sintassi grafica
  - ⇨la relazione di dipendenza può essere annotata con lo stereotipo <<depends>>
- Stereotipo
  - ⇨stringa tra doppie parentesi acute che descrive la funzione di un simbolo grafico

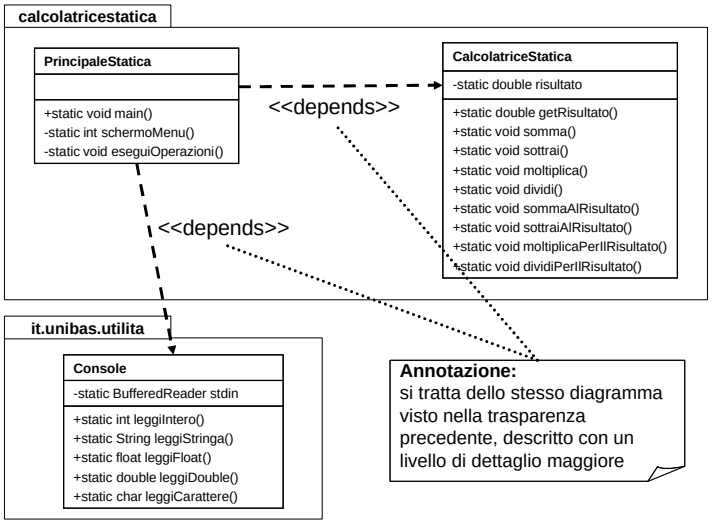
Il Diagramma che Abbiamo Visto



Il Diagramma in UML



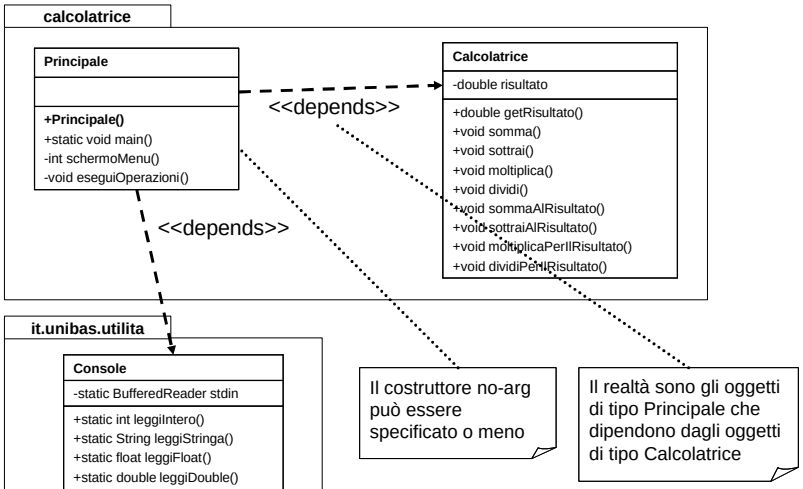
Lo Stesso Diagramma in UML



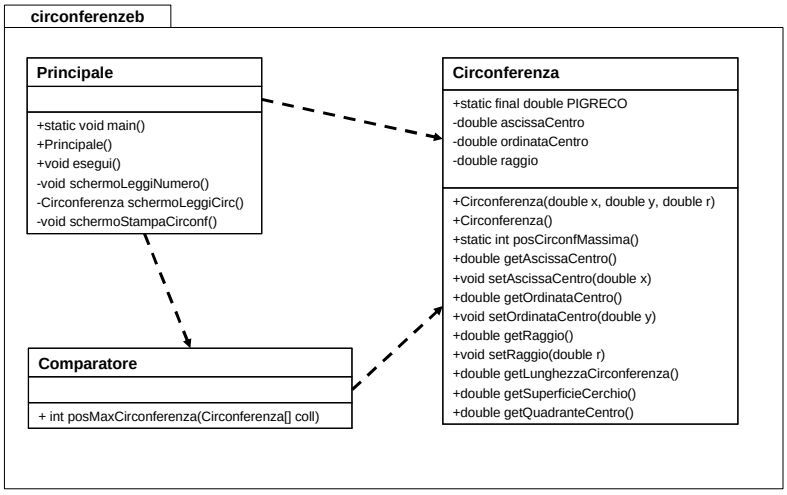
Diagrammi delle Classi

- Nota
  - ⇨nel diagramma delle classi di UML vengono rappresentate esclusivamente le classi
  - ⇨non vengono rappresentati gli oggetti
  - ⇨di conseguenza le frecce vengono rappresentate tra le classi
  - ⇨anche se rappresentano relazioni tra gli oggetti

Esempio: Calcolatrice in UML



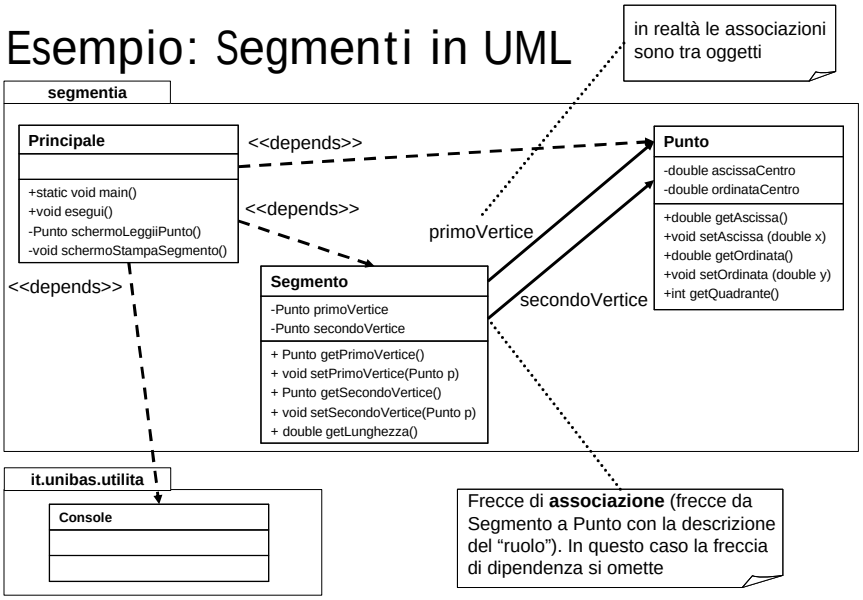
Esempio: Circonferenze in UML



Diagrammi delle Classi

- Relazione di associazione
  - ⇨descrive il fatto che componenti (classi o oggetti) di tipo A hanno proprietà che sono riferimenti ad oggetti di tipo B
  - ⇨ovvero sono in associazione
  - ⇨freccia solida che va da A a B
  - ⇨eventualmente annotata con il "ruolo", ovvero con il nome della proprietà
  - ⇨implica la relazione di dipendenza

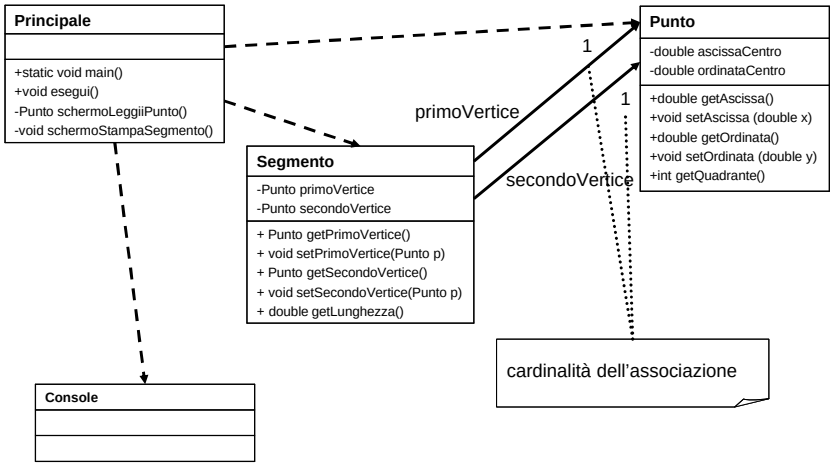
Esempio: Segmenti in UML



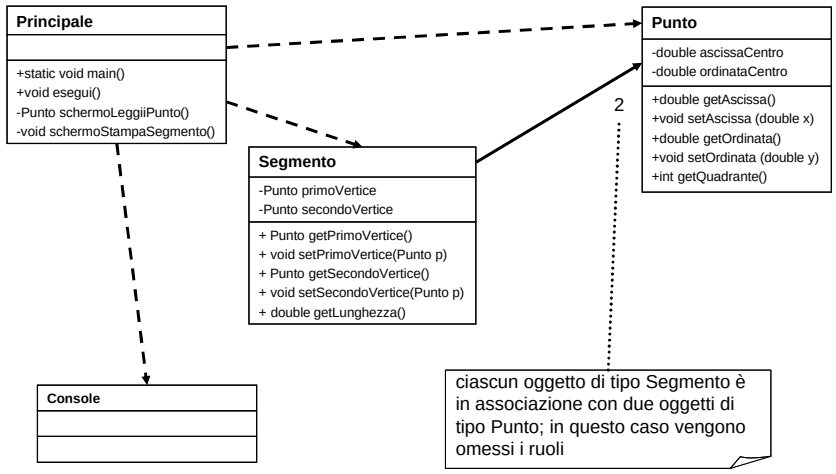
Diagrammi delle Classi

- Cardinalità
  - ⇨sintassi grafica per indicare con quanti oggetti è in associazione un oggetto
  - ⇨si rappresenta con un numero indicato dalla parte finale della freccia di associazione
- Valori tipici
  - ⇨1 (associazione “uno ad uno”)
  - ⇨* (associazione “uno a molti”)
  - ⇨la cardinalità 1 può essere omessa

Esempio: Segmenti in UML



Esempio: Segmenti in UML



Diagrammi delle Classi

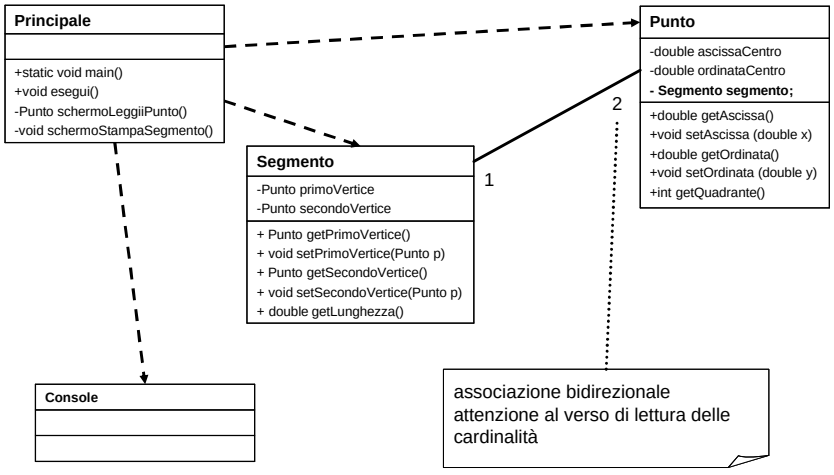
- Associazione bidirezionale
  - ⇨nel caso in cui A mantenga un riferimento a B e B mantenga il riferimento opposto ad A, si tratta di un'associazione bidirezionale
  - ⇨in questo caso viene omesso il verso della freccia e si specificano entrambe le cardinalita

Diagrammi delle Classi

- Esempio
  - ⇨riferimento in Punto a Segmento

```
public class Segmento {
   private Punto primoVertice;
   private Punto secondoVertice;
   ...
}
public class Punto {
   private double ascissa;
   private double ordinata;
   private Segmento segmento;
   ...
}
```


Esempio: Segmenti in UML



Diagrammi delle Classi

- Un esempio di associazione “uno a molti”
  - ⇨dobbiamo costruire una variante dell’applicazione delle circonferenze
- Nell’applicazione circonferenze
  - ⇨la classe Principale utilizza un array di Circonferenze
  - ⇨variabile locale al metodo esegui()
  - ⇨per ipotesi, supponiamo di utilizzare per l’array una proprietà della classe Principale invece che una variabile locale

Diagrammi delle Classi

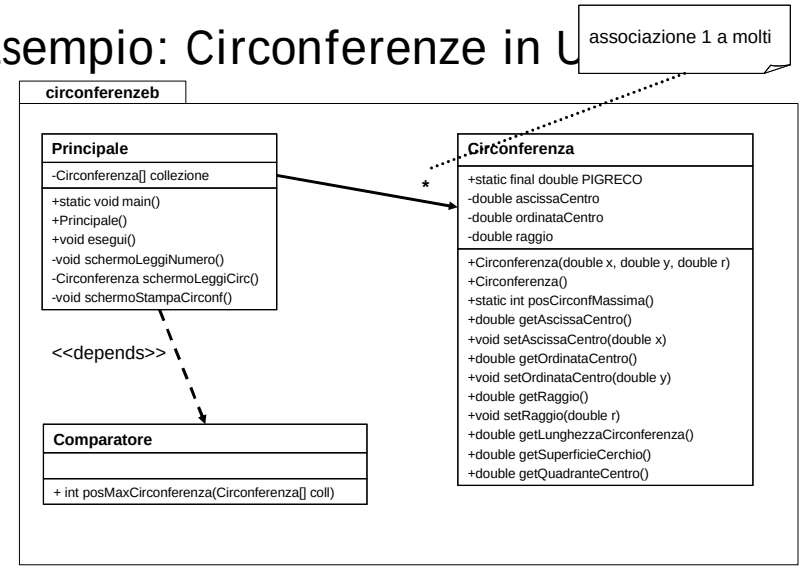
- Nota
  - ⇨questo frammento non corrisponde al reale codice dell’applicazione

```
public class Principale {

    private Circonferenza[] collezione;

    public void esegui() {
        int numeroCirconferenze = this.schermoNumeroCirconferenze();
        if (numeroCirconferenze > 0) {
            this.collezione = new Circonferenza[numeroCirconferenze];
        ....
    }
```

Diagrammi delle Classi

- In questo caso
  - ⇨Principale è in associazione con l'oggetto array di tipo Circonferenze[]
  - ⇨che è in associazione con n oggetti di tipo Circonferenza
- In UML
  - ⇨l'associazione viene indicata direttamente tra Principale e Circonferenza
  - ⇨"nascondendo" l'array intermedio

Esempio: Circonferenze in U



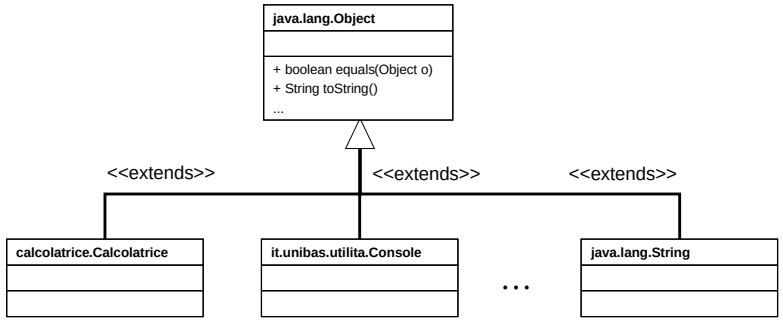
Diagrammi delle Classi

- Attenzione, però
  - ⇨questa soluzione non sarebbe metodologicamente corretta, e per questo non è stata usata
- Infatti
  - ⇨la relazione di associazione è più forte della relazione di dipendenza
  - ⇨introduce maggior accoppiamento tra i metodi di Principale (la proprietà è un dato globale)
  - ⇨introduce maggior accoppiamento tra le due classi (tutti i metodi di Principale sono in linea di principio accoppiati con Circonferenza)

Diagrammi delle Classi

- Relazione di ereditarietà
  - ⇨freccia dalla sottoclasse alla superclasse che termina con una grossa punta a triangolo
  - ⇨la relazione può essere annotata con lo stereotipo <<extends>>
  - ⇨tipicamente le sottoclassi vengono disegnate sotto le superclassi
  - ⇨costituiscono una “gerarchia”

Diagrammi delle Classi



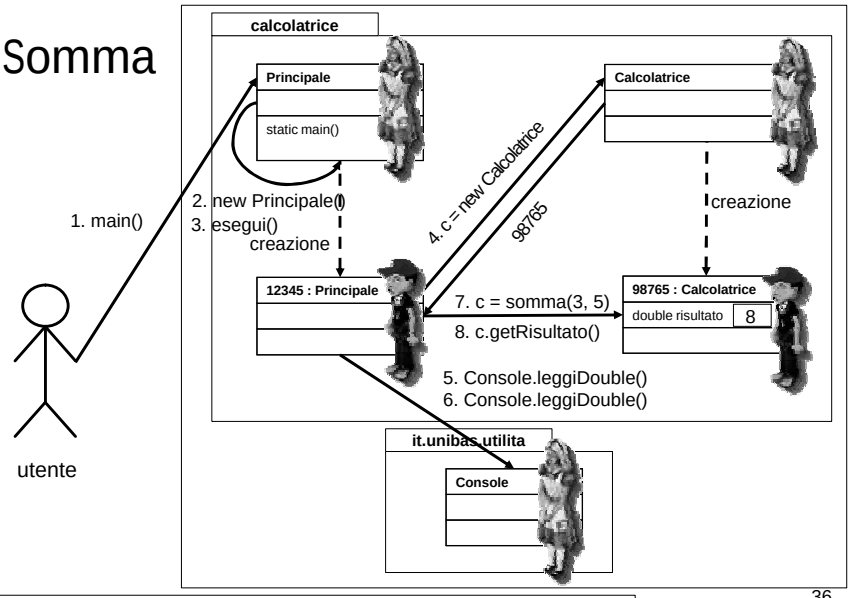
Diagrammi di Collaborazione

- Diagrammi dinamici
  - ⇨diagrammi che illustrano il funzionamento dell'applicazione e non solo la struttura del codice
  - ⇨tipicamente fotografano lo stato dell'applicazione in condizioni particolari di esecuzione
- Esempio
  - ⇨diagrammi di collaborazione

Diagrammi di Collaborazione

- Diagramma di collaborazione
  - ⇨i componenti sono rappresentati come rettangoli
  - ⇨le chiamate dei metodi sono indicate con frecce dal chiamante al chiamato
  - ⇨le frecce sono numerate per illustrare il flusso di esecuzione
  - ⇨anche questi sono stati sostanzialmente già visti negli esempi

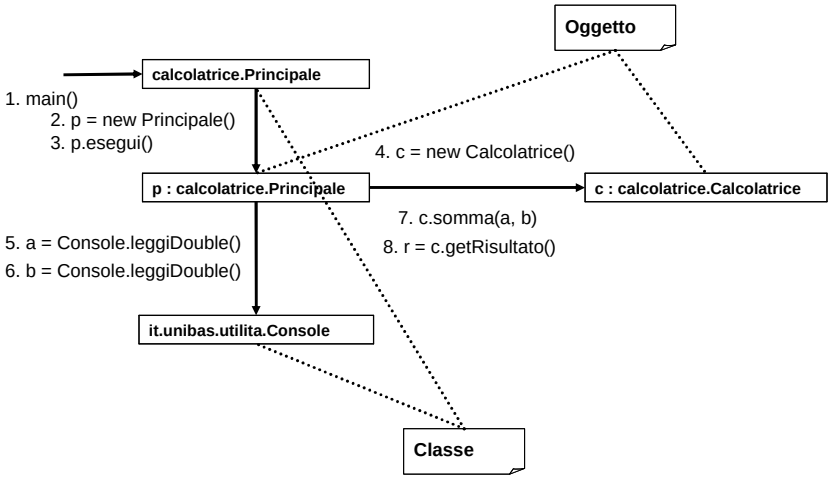
Somma



Diagrammi di Collaborazione

- Nella sintassi UML
  - ⇨rispetto agli esempi visti, la differenza principale è il modo in cui viene indicata la creazione degli oggetti
  - ⇨non traspare la sequenza completa di creazione
  - ⇨freccia dal componente che chiama il costruttore direttamente all'oggetto creato

Esecuzione di una Somma in UML



Diagrammi di Collaborazione

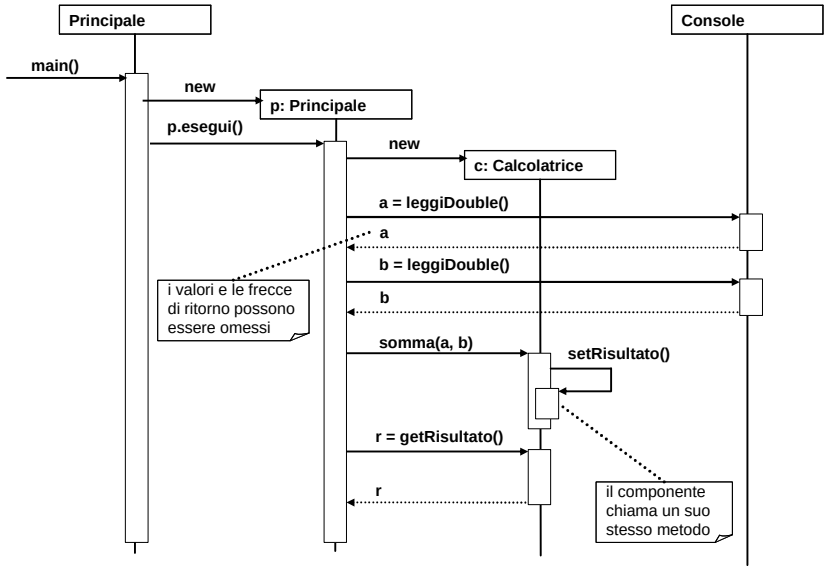
- Attenzione
  - ⇨differenza tra diagrammi statici e diagrammi dinamici
  - ⇨i diagrammi statici (ed in particolare il diagramma delle classi) descrivono l'organizzazione del codice
  - ⇨i diagrammi dinamici descrivono l'esecuzione dell'applicazione in una specifica condizione di utilizzo

Diagrammi di Sequenza

- Diagrammi di sequenza
  - ⇨una sintassi grafica alternativa per i diagrammi di collaborazione
- Disposizione verticale
  - ⇨a ciascun componente corrisponde una linea verticale "di esecuzione" su cui sono indicati gli intervalli di esecuzione dei metodi
  - ⇨le chiamate dei metodi sono indicate come frecce tra una linea verticale e l'altra

Diagrammi di Sequenza

- La creazione degli oggetti
  - ⇨anche nel caso dei diagrammi di sequenza è illustrata in modo semplificato
  - ⇨freccia dal chiamante all'oggetto creato
- Esempio
  - ⇨nel caso del diagramma di collaborazione precedente 4 componenti
  - ⇨per ciascuno un rettangolo e una "linea di esecuzione"



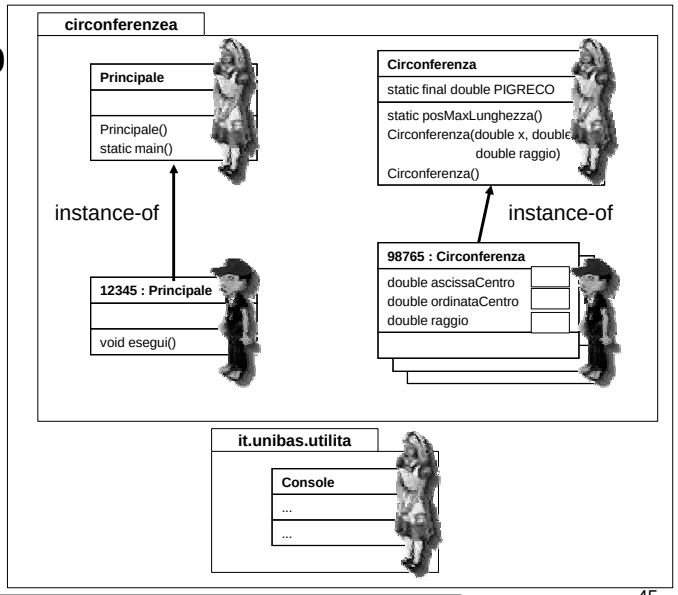
Diagrammi di Sequenza

- Vantaggi
  - ⇨rispetto al diagramma di collaborazione è più facile seguire il flusso di esecuzione
  - ⇨basta seguire l'ordine delle frecce dall'alto in basso (non sono necessari numeri di sequenza)
- Svantaggi
  - ⇨lo sfruttamento dello spazio è meno flessibile
  - ⇨richiede uno sviluppo prevalent. verticale

Diagrammi degli Oggetti

- Un ultimo tipo di diagramma dinamico
  - ⇨diagramma degli oggetti
  - ⇨descrive le relazioni tra le classi e gli oggetti in un certo momento del programma
  - ⇨quali e quanti oggetti ci sono e di quali classi sono istanze
  - ⇨diagramma relativamente poco usato in UML, che non utilizzeremo oltre

Esempio



Casi d'Uso

- Un altro strumento importante di UML
  - ⇨i casi d'uso
  - ⇨servono a descrivere il funzionamento di un'applicazione
  - ⇨in particolare, ne descrivono i requisiti funzionali, ovvero "cosa il sistema deve consentire di fare agli utenti"
  - ⇨in aggiunta esistono anche requisiti non funzionali (>>)

Casi d'Uso

- Caso d'Uso
  - ⇨storia di utilizzo del sistema da parte di un utente
  - ⇨descritto come sequenza di passi
  - ⇨con una struttura definita
- Struttura del caso d'uso
  - ⇨uno scenario principale ("sc. di successo")
  - ⇨zero o più scenari secondari che corrispondono a situazioni alternative di uso

Casi d'Uso

- Esempio
  - ⇨la calcolatrice
- Un unico caso d'uso
  - ⇨"Utente effettua operazioni"
- Lo scenario principale
  - ⇨quello in cui l'utente effettua varie operazioni
- Uno scenario alternativo
  - ⇨l'utente tenta di effettuare una divisione per 0

Casi d'Uso

- Scenario principale
  1. l'utente seleziona una operazione (+, -, *, /), eventualmente con memoria
  2. l'utente fornisce il primo operando
  3. se l'operazione non è con memoria, l'utente fornisce il secondo operando
  4. il sistema calcola il risultato e lo visualizza sullo schermo
  5. l'utente prosegue o interrompe l'applicazione

Casi d'Uso

- Scenario alternativo
  - ⇨punto di diramazione: 3
  - ⇨condizione: l'utente ha richiesto una divisione per 0

  3a. il sistema chiede di fornire di nuovo il divisore, finchè non viene fornito un valore corretto

  3b. l'esecuzione riprende dal passo 4

Casi d'Uso

- Altro esempio
  - ⇨le circonferenze
- Un unico caso d'uso
  - ⇨"Utente analizza circonferenze"
- Scenario principale
  - ⇨l'utente fornisce i dati delle circonferenze, e il sistema calcola i risultati e li visualizza
- Scenario alternativo
  - ⇨l'utente fornisce una collezione vuota

Casi d'Uso

- Scenario principale
  1. l'utente fornisce al sistema il numero di circonferenze da esaminare
  2. l'utente fornisce i dati di ciascuna circonferenza (coord. del centro e raggio)
  3. il sistema elabora i dati e visualizza per ogni circonf. i dati, la lunghezza, la superficie, il quadrante del centro
  4. il sistema visualizza la posizione della circonferenza più lunga

Casi d'Uso

- Scenario alternativo
  - ⇨punto di diramazione: 1
  - ⇨condizione: l'utente fornisce il numero 0

  2a. l'applicazione termina
- Attenzione
  - ⇨questa è una scelta di progetto
  - ⇨ci sono altre scelte possibili

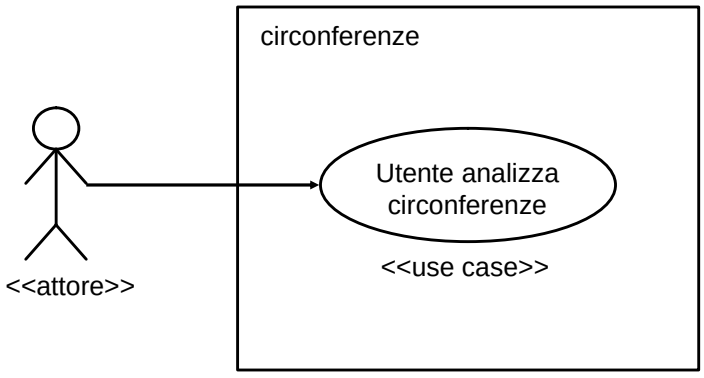
Casi d'Uso

- Nota
  - ⇨i casi d'uso sono storie di utilizzo del sistema, e quindi sono sostanzialmente testi
  - ⇨UML, però, prevede tra i suoi diagrammi anche un diagramma dei casi d'uso
  - ⇨il diagramma ha una sintassi grafica molto semplificata e fornisce solo un elenco dei casi d'uso forniti dai sistemi

Casi d'Uso

- Diagramma dei casi d'uso
  - ⇨si tratta di un diagramma statico
  - ⇨il sistema viene rappresentato come una scatola nera
  - ⇨ciascun caso d'uso viene rappresentato come un ovale
  - ⇨gli attori che partecipano al caso d'uso sono rappresentati come utenti stilizzati da cui partono frecce verso il caso d'uso

Casi d'Uso



Riassumendo

- Diagrammi UML
  - ⇨Diagrammi delle Classi
  - ⇨Diagrammi di Collaborazione
  - ⇨Diagrammi di Sequenza
  - ⇨Diagrammi degli Oggetti
- Casi d'Uso

Termini della Licenza