Concetti introduttivi

La programmazione

- Programma: sequenza di operazioni semplici (istruzioni e decisioni) eseguite in successione.
 - Un programma indica al computer i passaggi da compiere per svolgere un compito preciso.
 - I programmi danno flessibilità di impiego ai computer
- L'attività di progettazione e implementazione dei programmi è detta programmazione.
- I programmi sono scritti utilizzando linguaggi di programmazione

Linguaggi di Programmazione

- I linguaggi di programmazione sono in genere classificati in
- Linguaggi macchina
 - Istruzioni macchina codificate con sequenze numeriche
 - Dipendenti dalla macchina
- Linguaggi assembly
 - Istruzioni macchina codificate con codici mnemonici
 - Dipendenti dalla macchina
- Linguaggi di alto livello (C, Pascal, Java, ecc.)
 - Istruzioni ad un livello concettuale più elevato
 - Indipendenti dalla macchina

30 40 16 100 156

LOAD REG, loc_b ADD REG, loc_a MOV loc_b, REG

b = a+b;

Linguaggi di alto livello

- I linguaggi di alto livello consentono un maggiore livello di astrazione
 - Permettono di descrivere l'idea che sta dietro l'operazione da compiere

- Sono più vicini ai linguaggi naturali
- Seguono delle convenzioni rigide per facilitarne la traduzione in codice macchina (compilazione)

Compilazione

- Le istruzioni scritte in un linguaggio ad alto livello devono essere tradotte in istruzioni macchina per poter essere "comprese" dalla CPU
 - Il compilatore è il programma che si occupa di tradurre il codice
- L'insieme di istruzioni macchina (linguaggio macchina) dipende dalla CPU
 - Il "back-end" di un compilatore dipende dalla CPU

Linguaggi di alto livello

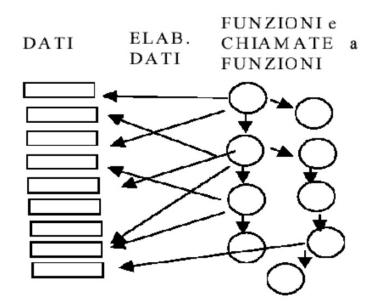
- I linguaggi di alto livello possono essere classificati in vari modi.
- Di interesse per il corso:
 - Linguaggi procedurali o imperativi
 - o C, Pascal, ...
 - Linguaggi orientati agli oggetti
 - o C++, Java, ...
- Altre classi di linguaggi:
 - Linguaggi funzionali
 - Lisp, SML, ...
 - Linguaggi logici o dichiarativi
 - o Prolog, LDL, ...

Limite dei linguaggi procedurali

- Costringe a pensare soluzioni che riflettono il modo di operare del computer piuttosto che la struttura stessa del problema.
 - Per problemi non numerici questo spesso è difficile
 - Il riutilizzo delle soluzioni è più complicato e improbabile
 - La produzione e la manutenzione del software sono costose

Esempio

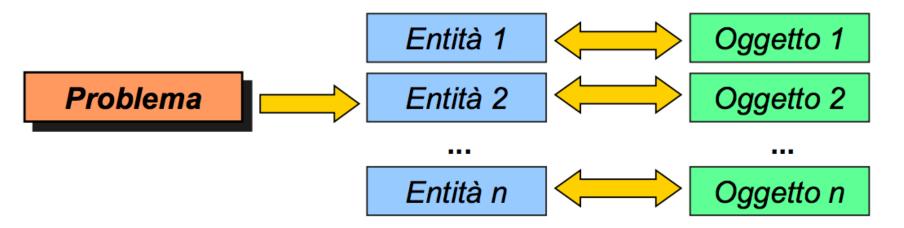
 Scrivere un programma per la gestione di un conto corrente bancario



 Dove sono finiti i concetti di conto corrente, prelievo, versamento, saldo corrente ?

Linguaggi Orientati agli Oggetti

 I linguaggi ad oggetti permettono al programmatore di rappresentare e manipolare non solo dati numerici o stringhe ma anche dati più complessi e aderenti alla realtà (conti bancari, schede personali,...)



Esistono controindicazioni?

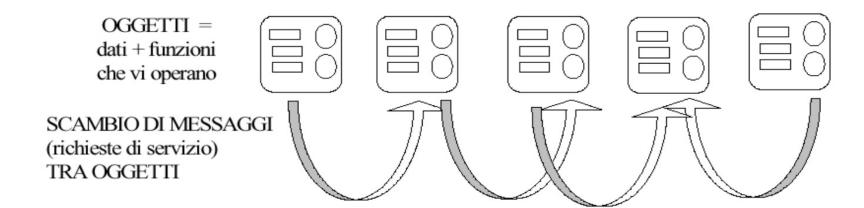
- Il paradigma di programmazione orientata agli oggetti paga la sua semplicità e versatilità in termini di efficienza
- Va molto bene per lo sviluppo di applicazioni, ma non è adatto per lo sviluppo di software di base
 - Sistemi operativi
 - Driver
 - Compilatori

Dominio del problema e dominio della soluzione

- I linguaggi procedurali definiscono un "dominio della soluzione" che "astrae" la macchina sottostante
 - Astrazione procedurale
- Il programmatore deve creare un mapping fra "dominio del problema" e "dominio della soluzione"
 - Tale mapping è spesso innaturale e di difficile comprensione

Linguaggi orientati agli oggetti

- Forniscono astrazioni che consentono di rappresentare direttamente nel dominio della soluzione gli elementi del dominio del problema
 - Oggetti
 - Classi
 - Messaggi



Astrazione

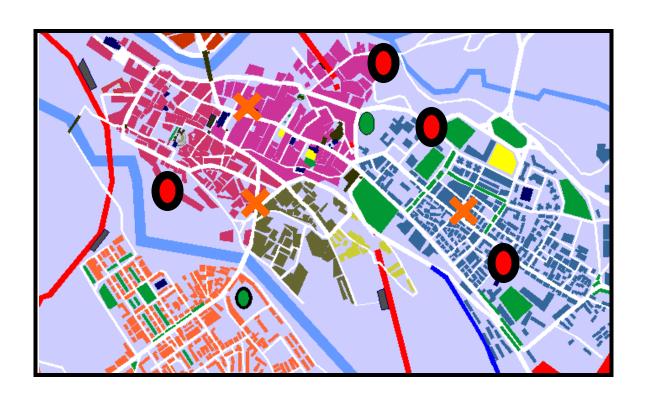
- Astrazione: Una vista di un oggetto che si focalizza sulle informazioni rilevanti ad un particolare scopo e che ignora le informazioni rimanenti
- Information Hiding: Un tecnica per lo sviluppo del software in cui le interfacce dei moduli mostrano il meno possibile del loro funzionamento interno e gli altri moduli sono prevenute dall'usare informazioni del modulo che non sono definite nell'interfaccia

Modelli

 I modelli sono nati prima dei calcolatori e NON devono necessariamente essere realizzati mediante calcolatori







Elementi del modello

- Ogni modello è formato da elementi che rappresentano entità
- Gli elementi del modello presentano un comportamento consistente
- A seconda dei loro comportamenti comuni, gli elementi possono essere raggruppati in categorie diverse
- Il comportamento di un elemento può essere provocato da azioni esterne

Modelli in Java

- Elementi del modello: Oggetti
- La categorie di oggetti vengono chiamate Classi
- Una classe
 - Determina il comportamento degli oggetti appartenenti
 - E' definita da una sezione di codice
- Un oggetto
 - Appartiene ad una classe
 - Costituisce una istanza di tale classe

Esempio

Classe operatore:

- Definisce il comportamento degli operatori (ad esempio, cambiamento di locazione, registrano il tempo di un intervento, ecc.)
- Ogni operatore in servizio è una istanza di tale classe

O Classe chiamata:

- Definisce il comportamento delle chiamate (ad esempio, priorità, orario di arrivo, cliente chiamante ecc.)
- Per ogni chiamata che arriva si crea una istanza

Programmazione OO

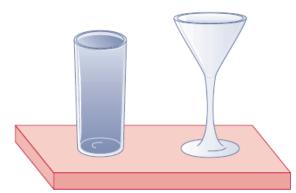
- Focus: gli oggetti
 - e le classi che ne definiscono il comportamento
- Filosofia: In un programma in esecuzione sono gli oggetti che eseguono le operazioni desiderate
- Programmare in Java
 - Scrivere le definizioni delle classi che modellano il problema
 - Usare tali classi per creare oggetti
- o Java è dotato di classi ed oggetti predefiniti
 - Non si deve continuamente reinventare la ruota

Obiettivi

- Formalizzare oggetti del mondo reale in oggetti del dominio applicativo che possano essere utilizzati dall'applicazione, individuando solo gli aspetti che interessano e tralasciando tutto ciò che è superfluo
- Riuscire a fare interagire gli oggetti tra loro, al fine di raggiungere l'obiettivo per il quale è stata creata l'applicazione
- Implementare codice in modo tale che sia il più possibile conforme agli standard di programmazione

Distinguere gli oggetti

- Esempio: un Bicchiere
- Ne sappiamo definire le caratteristiche e conosciamo anche quali azioni si possono fare con esso.
- Possiamo definirne la forma, il colore, il materiale di cui è fatto e possiamo dire se è pieno o vuoto.
- Sappiamo anche che si può riempire e svuotare.
- Abbiamo definito un oggetto attraverso
 - le sue caratteristiche
 - le operazioni che può compiere



Il livello di astrazione

 I linguaggi di programmazione si sono evoluti in modo che i codici sorgenti potessero astrarsi sempre più dal modo in cui gli stessi, una volta compilati, sarebbero stati eseguiti.



- Nella OOP non ci si vuole più porre i problemi dal punto di vista del calcolatore, ma si vogliono risolvere facendo interagire oggetti del dominio applicativo come fossero oggetti del mondo reale.
- L'obiettivo è di dare uno strumento al programmatore, per formalizzare soluzioni ai propri problemi, pensando come una persona e senza doversi sforzare a pensare come una macchina.



Il processo di astrazione: le classi

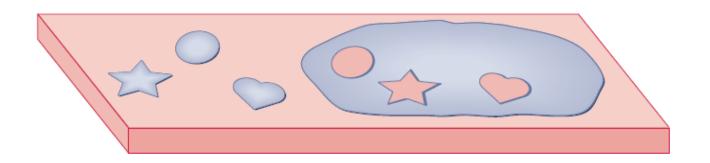
 Per popolare il dominio applicativo utilizzato dall'applicazione è necessario creare gli oggetti, e per fare questo è necessario definire le classi.

 Una classe è lo strumento con cui si identifica e si crea un oggetto.

Una classe è un modello per la creazione di oggetti

 La classe è paragonabile allo stampo

 gli oggetti sono i biscotti ottenuti con quello stampo

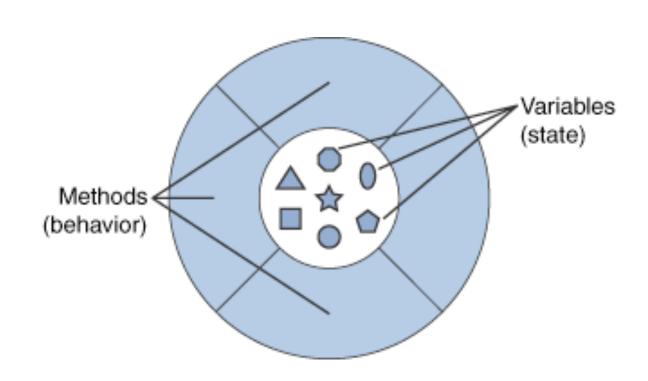


Classi e tipi di dato

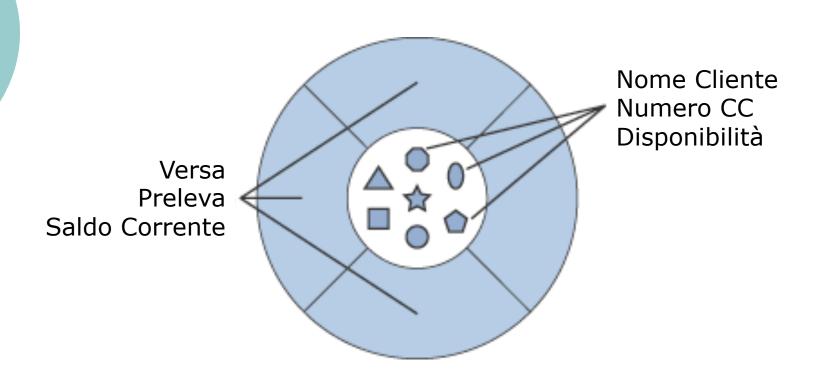
 Una classe è a tutti gli effetti un tipo di dato (come gli interi e le stringhe e ogni altro tipo già definito)

 Nella programmazione orientata agli oggetti, è quindi possibile sia utilizzare tipi di dato esistenti, sia definirne di nuovi tramite le classi

Oggetti, astrazione ed information hiding



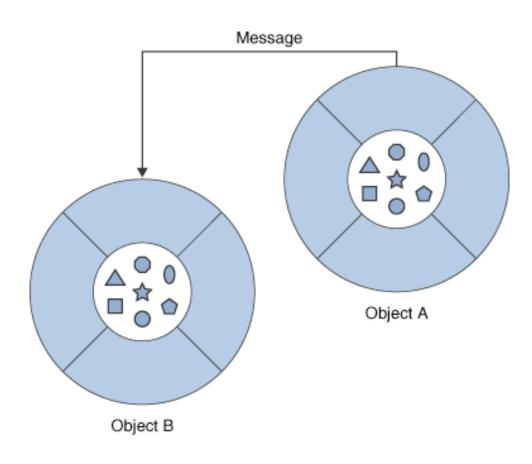
Oggetti, astrazione ed information hiding



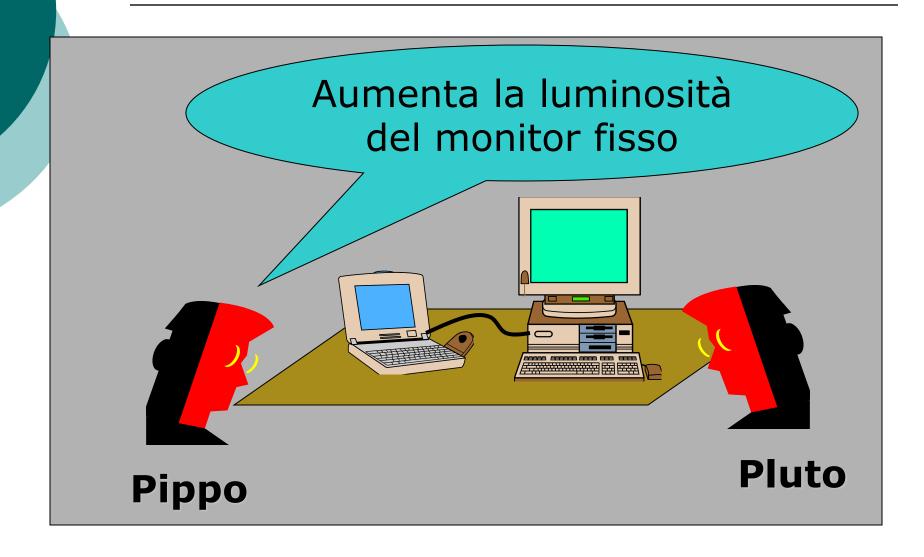
Messaggi

- Gli oggetti sono gli elementi attivi di un programma. Come fanno gli oggetti a compiere le azioni desiderate ?
- Gli oggetti sono attivati dalla ricezioni di un messaggio
- Una classe determina i messaggi a cui un oggetto può rispondere
- I messaggi sono inviati da altri oggetti

Messaggi



Invio di un messaggio



Invio di un messaggio

Aumenta la luminosità del monitor fisso

Messaggio

Comportamento:

- Modifica
- La proprietà luminosità
- In aumento

Destinatario (Receiver)

Referenza ad un oggetto



Messaggi

- Per l'invio di un messaggio è necessario specificare:
 - Ricevente
 - Messaggio
 - Eventuali informazioni aggiuntive
- Non tutti i messaggi sono comprensibili da un determinato oggetto:
 - Es: abbassare il volume
 - Un messaggio deve invocare un comportamento dell'oggetto

Nomi e referenze

- Le classi hanno un nome
 - Ogni classe Java deve avere un nome
 - Ogni classe ha un solo nome
 - Es: Impiegato, Molecola, ContoCorrente
 - Convenzione: comincia con una lettera maiuscola
- Regole Java per i nomi (identificatori)
 - Lettere, cifre e caratteri speciali (es: "_")
 - Devono cominciare con una lettera
 - Il linguaggio è case sensitive

Nomi e referenze

- Gli oggetti NON hanno nome
 - In Java gli oggetti sono identificati da riferimenti
 - Un riferimento (reference) è una frase che si riferisce ad un oggetto
 - I riferimenti sono espressioni
 - E' possibile avere più riferimenti ad uno stesso oggetto

Classi ed oggetti predefiniti

Modellano componenti e comportamenti del sistema Modellano l'interfaccia grafica di interazione con l'utente Modellano "oggetti" di uso comune, ad esempio Data e Calendario

- Un esempio: il monitor
 - Ci si riferisce a lui mediante il riferimento:
 System.out

```
public class Program1 {
     public static void main (String[] arg) {
         System.out.println("Benvenuti al corso");
     }
}
```

PrintStream e System.out

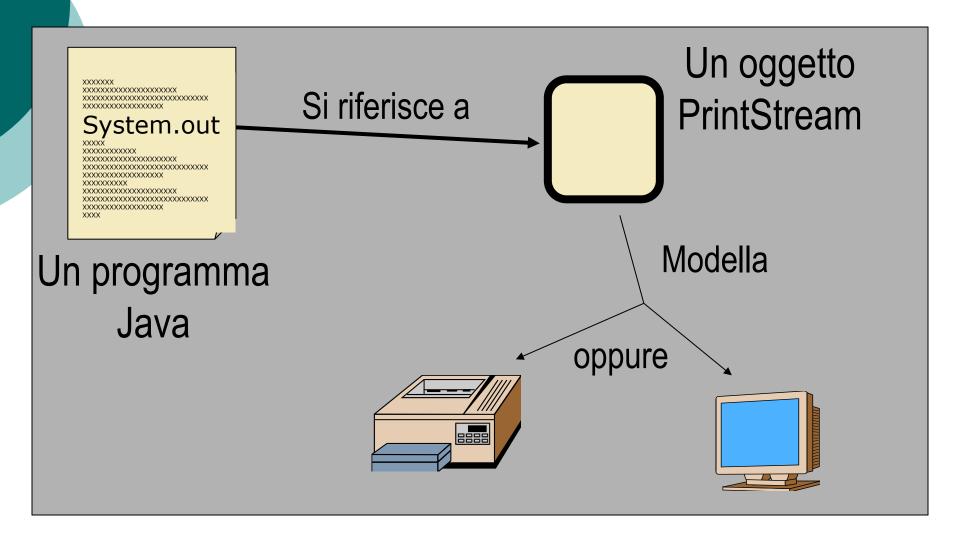
La classe PrintStream

- Modella monitor e stampanti
- Comportamento: visualizzare sequenze di caratteri

System.out

- Un riferimento ad un oggetto predefinito
- Istanza della classe PrintStream

PrintStream e System.out



Messaggi in Java

- Forma generale
 Comportamento-desiderato (altre-informazioni)
- Esempio:
- println("Benvenuti al corso")
 - Comportamento: println (stampa una linea)
 - Informazione: "Benvenuti al corso" (contenuto della linea)

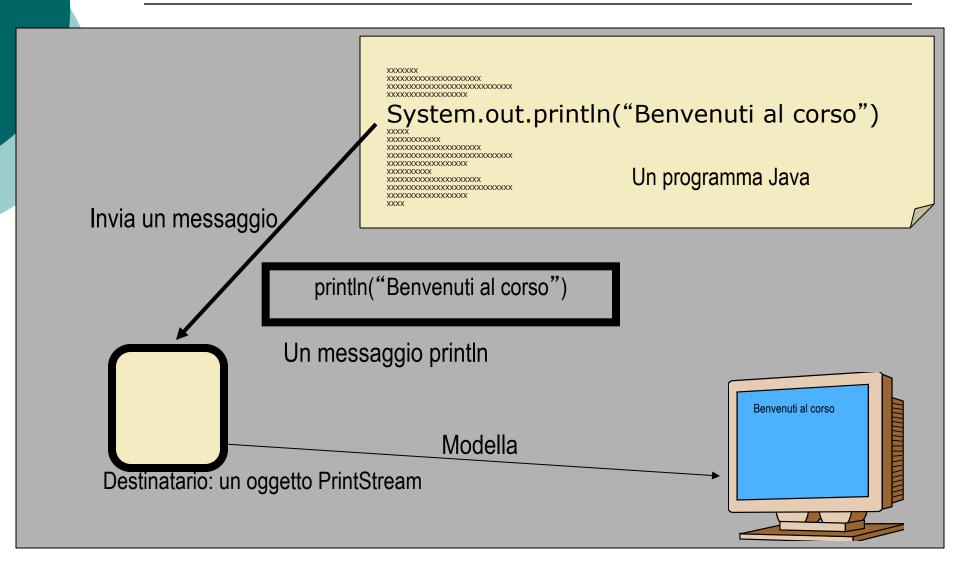
Invio di un messaggio

- Forma generale:
 Riferimento-al-destinatario.messaggio
- Esempio:
- System.out.println ("Benvenuti al corso")

Riferimento Messaggio

 L'oggetto a cui si riferisce il riferimento System.out è il destinatario del messaggio println("Benvenuti al corso")

Invio di un messaggio



Istruzioni

- Le istruzioni Java
 - Provocano un'azione (es: inviare un messaggio)
 - Devono essere chiuse da punto e virgola ";"
- Esempio
- System.out.println("Benvenuti al corso");
 - L'invio di un messaggio deve essere sempre espresso da una istruzione

Forma di un programma

o Almeno per le prime lezioni:

```
public class ProgramName {
      public static void main (String[] arg) {
            istruzione;
            istruzione;
            istruzione;
```