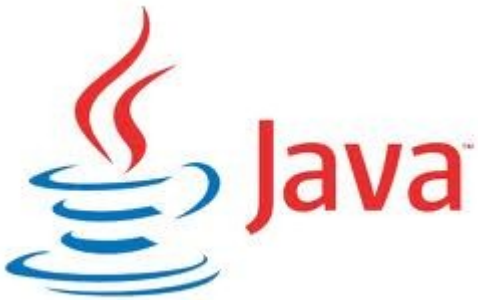
Metodologie di Programmazione

- Lezione 1: Introduzione

Obiettivi del corso

- Fornire i concetti fondamentali della programmazione orientata agli oggetti
- Fornire strumenti e metodologie di base per la progettazione del software
- Usando:



Da dove venite...

- Un tipo di programmazione imperativa in cui il programma è costituito da una o più procedure (funzioni)
- Tipi di base, costanti, espressioni, variabili, operatori, costrutti di selezione (if) e di iterazione (for, while), array, funzioni, ricorsione, strutture, stringhe, input/output, ecc.

Dove stiamo andando

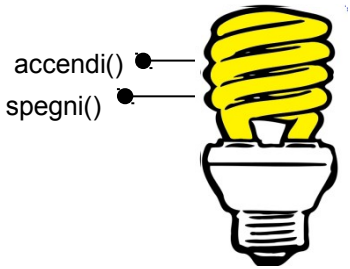
- La programmazione orientata agli oggetti fornisce nuovi strumenti per rappresentare elementi nello spazio del problema (qualsiasi esso sia!)
  - Gli elementi sono chiamati oggetti
  - Puoi descrivere il problema in termini del problema stesso, non in termini del computer su cui gira il programma

Tutto è un oggetto

- Un oggetto è un po' come un piccolo computer
  - Ha uno stato
  - Puoi "farci delle cose" (= ha delle operazioni che puoi chiedergli di eseguire)
- Esempio:
  - Modellare una lampadina in una stanza
    - Stato: accesa vs. spenta
    - Operazioni: accendi, spegni
  - Modellare un personaggio in un ambiente simulato:

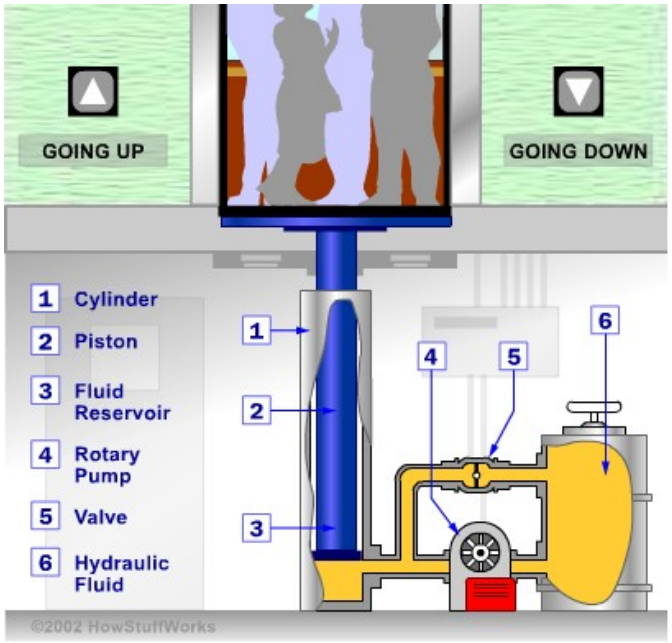
Un programma è una collezione di oggetti

- Per effettuare una richiesta a un oggetto, si invia un messaggio a quell'oggetto
- Un messaggio è una richiesta di chiamata a una funzione (metodo) che appartiene a un particolare oggetto



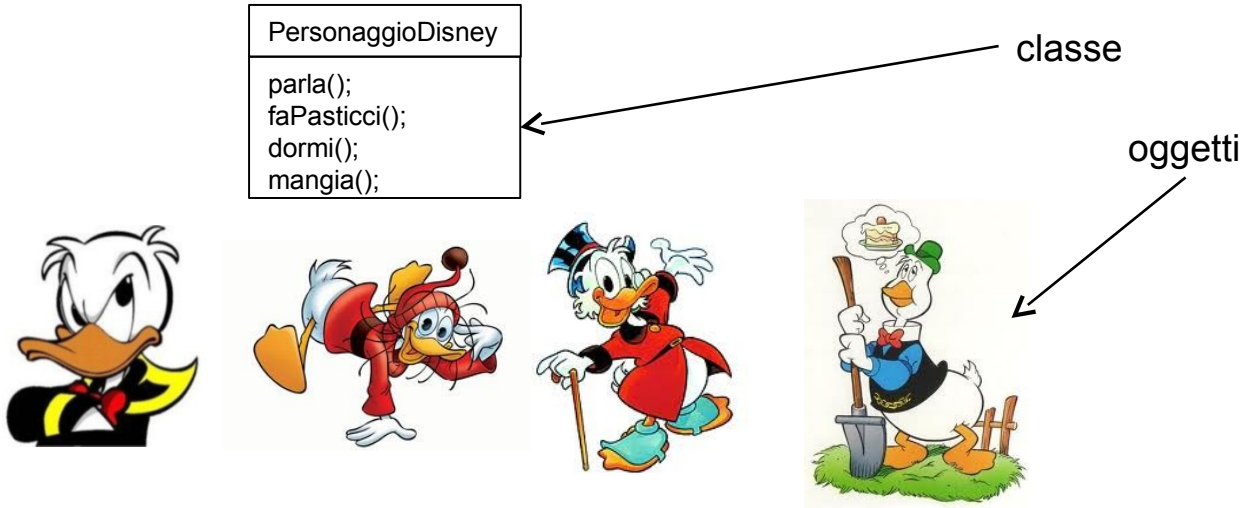
La “memoria” degli oggetti e l’information hiding

- Un nuovo tipo di oggetto può essere creato utilizzando oggetti esistenti
- Un programma può nascondere la sua complessità mediante la semplicità degli oggetti



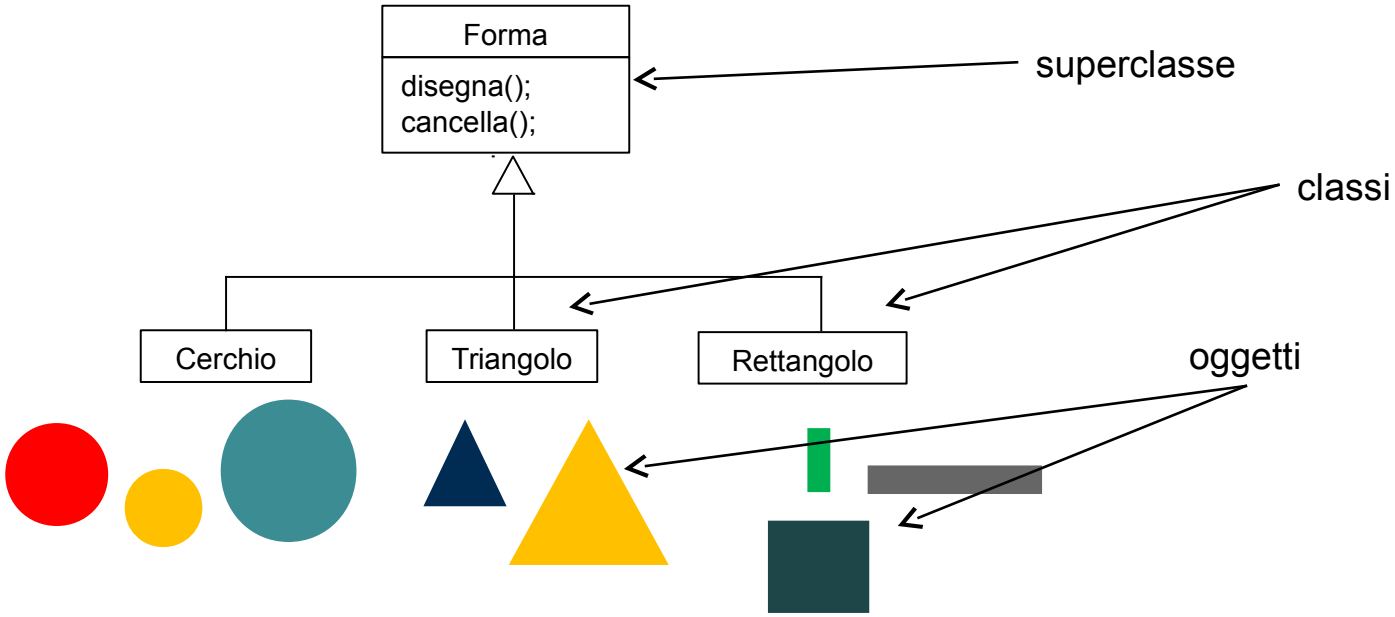
Ogni oggetto ha un suo tipo: la classe

- Ogni oggetto è istanza di una classe
- La classe è identificata dai messaggi (metodi) che essa possiede
- Tutti gli oggetti di uno stesso tipo possono ricevere gli stessi messaggi



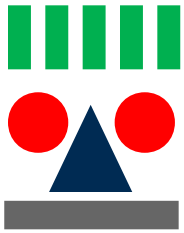
Ereditarietà

- Vogliamo evitare di ricreare nuove classi di oggetti quando esse hanno funzionalità simili



Polimorfismo

- E' possibile utilizzare una classe base, senza dover conoscere necessariamente la classe specifica di un oggetto
- Permette di scrivere codice che non dipende dalla classe specifica
- Posso aggiungere nuove sottoclassi anche in seguito!
- Ad esempio, una **ImmagineVettoriale** è una collezione di oggetti di tipo **Forma** in determinate posizioni:



Parliamo di Java

- Un linguaggio di programmazione potente, orientato agli oggetti
  - Creato da James Gosling e altri informatici di Sun Microsystems (ora Oracle)
- Recente (1995)
  - Precursori: Smalltalk (fine '70), C++ (inizio '80)
- Costruito per essere "sicuro", cross-platform e internazionale

Caratteristiche di Java (I)

- **Indipendenza dalla piattaforma:**
  - Portabile: "WORA" (write once, run anywhere)
  - Al contrario di linguaggi come il C o il C++ non viene compilato su una macchina o piattaforma, ma nel bytecode di una macchina virtuale
  - Architecture-neutral
- **Sicurezza:**
  - Non permette manomissioni
  - Le tecniche di autenticazione sono basate su codifiche con chiavi pubbliche
- **Robustezza:**
  - Situazioni tipiche d'errore vengono eliminate il più possibile a tempo di compilazione
  - Laddove non possibile, gestite a tempo di esecuzione con appositi controlli

Caratteristiche di Java (2)

- **Multithreaded:**
  - E' possibile scrivere programmi che gestiscono attività eseguite in contemporanea (thread)
  - Facilita la costruzione di applicazioni interattive
- **Interpretato:**
  - Il byte code è tradotto "al volo" in istruzioni macchina native e non viene memorizzato da nessuna parte
  - Rende più veloce e snello il processo di sviluppo
- **Alte prestazioni:**
  - Con l'uso dei compilatori Just-In-Time (JIT), le prestazioni sono le stesse se non addirittura SUPERIORI del codice nativo

Caratteristiche di Java (3)

- **Distribuito:**
  - Progettato per ambienti distribuiti come Internet
- **Dinamico**
  - Si adatta a un ambiente in evoluzione
  - Porta con sé parecchie informazioni a tempo di esecuzione per verificare e risolvere gli accessi agli oggetti
- **Enterprise, Web e Mobile**
  - E' utilizzato a livello enterprise, web e mobile per applicazioni robuste, solide, sicure e distribuite

Statistiche

- Dal sito http://www.java.com/en/about/

- 97% of Enterprise Desktops Run Java
- 89% of Desktops (or Computers) in the U.S. Run Java
- 9 Million Java Developers Worldwide
- #1 Choice for Developers
- #1 Development Platform
- 3 Billion Mobile Phones Run Java
- 100% of Blu-ray Disc Players Ship with Java
- 5 Billion Java Cards in Use
- 125 million TV devices run Java
- 5 of the Top 5 Original Equipment Manufacturers Ship Java ME