IQUADRO
INGEGNERIA INFORMATICA

Corso Java

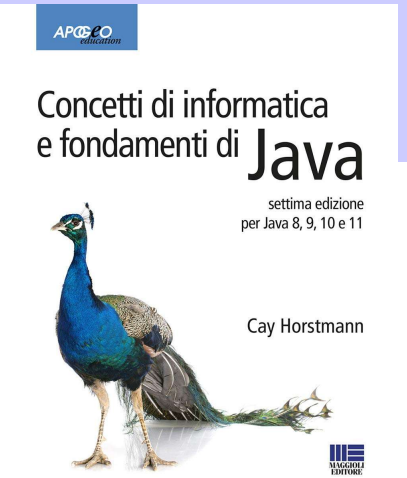
Thierry Bodhuin
bodhuin@gmail.com

- Linguaggio di Programmazione Java
- Ambiente di sviluppo

20 Maggio 2024

Testi consigliati

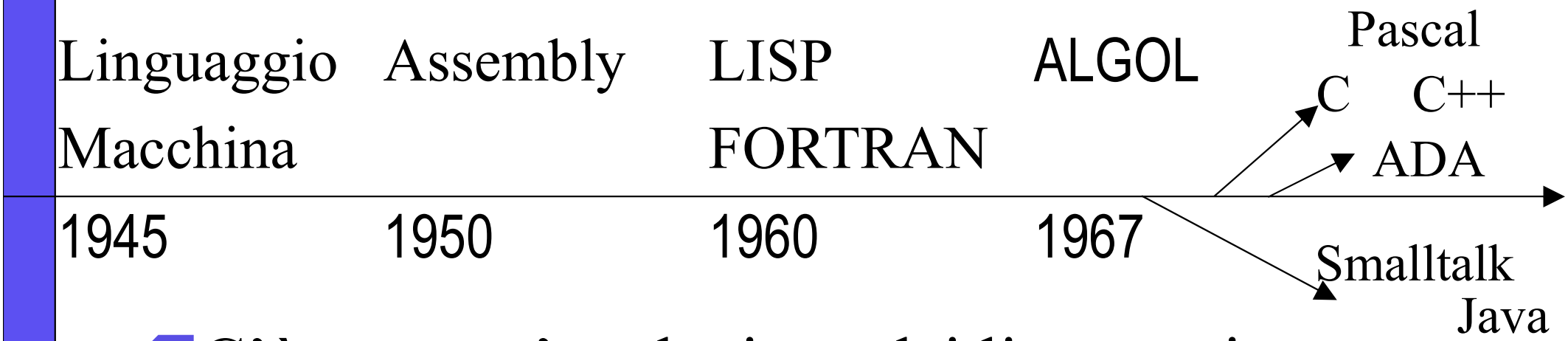
- Cay S. Horstmann “Concetti di Informatica e Fondamenti di Java”
  settima edizione per Java 8,9,10 e 11



- Giovanni Pighizzini, Mauro Ferrari “Dai fondamenti agli oggetti - Corso di programmazione Java” Quarta Edizione, Pearson / Addison-Wesley

- David Arnow, Gerald Weiss “Introduzione alla programmazione con Java - Un approccio object-oriented” Jackson libri

Breve storia dei linguaggi di programmazione…



- C'è stata un'evoluzione dei linguaggi verso:
  - Astrazione,
  - Semplificazione,
  - Similarità con il ragionamento umano.

Linguaggi di programmazione

- I *linguaggi di programmazione* sono classificati in tre livelli:
  - linguaggi macchina,
  - linguaggi assembly,
  - linguaggi ad alto livello.

Dall'assembly al FORTRAN

- Il programmatore non deve necessariamente occuparsi della gestione della memoria.
  - può dicharare una variabile e ottenere dal computer l'assegnazione di una area di memoria alla stessa.
- Il programmatore può usare il linguaggio della matematica e non le istruzioni mnemoniche.

I problemi del FORTRAN

- Programmi difficili da leggere e da correggere a causa delle istruzioni “GOTO”.
- “Pezzi” di codice sono simili tra loro e potrebbero essere scritti solo una volta.
  - Nelle prime versioni non c’è alcuno strumento del linguaggio che aiuti a creare “moduli” o funzioni”.

Alcune soluzioni

- Modularizzazione :
  - creare sotto-programmi il più possibile indipendenti tra loro e isolare dentro tali “moduli” le operazioni più semplici.
    - Costruire da moduli semplici moduli via via più complessi…
- Astrazione:
  - slegare il programmatore dal modello della macchina e avvicinarlo al modello del problema da risolvere.
- Strutturazione:
  - i salti nell’esecuzione del programma debbono essere espliciti, visibili e chiari.
    - NON si deve usare il “GOTO”

Una nuova generazione di linguaggi

- Il linguaggio ALGOL non ha mai preso piede ma … è stato il modello per :

  C, Pascal, MODULA, FORTRAN77.

- Novità della programmazione strutturata:
  - funzioni che isolano i sotto-programmi;
  - controllo della gestione della memoria mediante variabili “tipizzate”.

Ma i problemi non si sono esauriti…

- I linguaggi ALGOL- like non sono ancora soddisfacenti.
  - Le modalità di organizzazione dei dati (*strutture dati*) restano separate dai metodi necessari a manipolare i dati stessi.
  - Le funzioni/moduli non sono “a tenuta stagna” :
    - possono influenzarsi tra loro in maniera non sempre esplicita al programmatore.
  - Non è quasi mai facile *riutilizzare* una funzione in un altro programma.
- La *manutenzione* del software (correzione ed aggiornamento) diviene rapidamente più difficile e costosa dello sviluppo ex-novo dello stesso software.

L'ultima tendenza …

- Linguaggi orientati agli oggetti:

  SMALLTALK, EIFFEL, C++, JAVA.

  - Ad ogni "entità" del problema da risolvere corrisponde un oggetto capace di memorizzare i dati in maniera organizzata e di manipolarli.
    - Un oggetto è una scatola nera con ingressi e uscite chiaramente ed esplicitamente definiti:
  - Il sogno della modularizzazione sembra così realizzarsi.

Evoluzione delle tecniche di programmazione

Background Java

- Java
  - Creato nel 1991
  - Da James Gosling et al. della Sun Microsystems.
  - Inizialmente chiamato Oak, albero fuori la finestra di James Gosling
    - Il nome e stato cambiato per Java perché ci stava già un linguaggio chiamato Oak e perché Java significa Caffè…
  - Java JDK (Java Development Kit) 1.0 nel 1995.
  - Oggi JDK 22.

Perché ci son voluti 40 anni ?

- Perché ogni linguaggio nuovo si trova un po' più "lontano" dal linguaggio macchina e non è affatto semplice costruire programmi che TRADUCANO *automaticamente* dal linguaggio ad alto livello verso il codice macchina direttamente eseguibile.
  - Tali programmi traduttori sono complessi da creare e richiedono grandi risorse computazionali.
    - Nascono quindi problemi di efficienza che solo computer veloci possono risolvere adeguatamente.

I paradigmi di programmazione

- Forniscono la filosofia con cui si scrivono i programmi e stabiliscono :
  - la metodologia con cui si scrivono i programmi,
  - il concetto di computazione.
- I linguaggi devono *consentire* ma soprattutto *spingere* all'adozione di un particolare paradigma.
  - Funzionale
  - Logica
  - Imperativa
  - Modulare
  - Orientata agli oggetti

Paradigma procedurale

- Enfasi sulla soluzione dei problemi mediante modifica progressiva dei dati
  - Esecuzione sequenziale di istruzioni
  - Stato della memoria
  - Cambiamento di stato tramite esecuzione di istruzioni
- Aderenti al modello della macchina di von Neumann
- Molto efficienti
- Ha mostrato limiti nello sviluppo e mantenimento di software complessi
- **Pascal**, **C**

Influenza del modello di macchina

- Concetto di istruzione
- Concetto di sequenzialità e iterazione
  - Il programma assolve il compito eseguendo le istruzioni in sequenza
- Concetto di variabile e di assegnamento
  - Le celle di memoria hanno un indirizzo e contengono i dati da manipolare
  - Le variabili hanno un nome e un valore
  - L'assegnamento di un valore a una variabile equivale al trasferimento di un dato in una cella

Paradigma funzionale

- Primo tentativo di non rifarsi al modello di macchina di von Neumann
  - Il programmatore può IGNORARE la struttura fisica della macchina e scrivere i propri programmi in maniera assolutamente naturale basata sulla logica e la matematica.
- La computazione avviene tramite funzioni che applicate ai dati riportano nuovi valori
  - Le funzioni possono essere applicate a funzioni in catena e possono essere ricorsive
- **Lisp, …**

Paradigma modulare

- Introduce il concetto di modulo che nasconde i dati all'utente
  - I dati possono essere letti solo tramite un'opportuna interfaccia
- **Modula-2**, **Ada**

IQUADRO INGEGNERIA INFORMATICA

Paradigma a oggetti (OOP)

- Spinge ulteriormente il concetto di modulo che incapsula i dati con le classi
  - Le classi hanno anche una struttura gerarchica ed ereditano caratteristiche e funzionalità
- Introdotto per migliorare l'efficienza del processo di produzione e mantenimento del software

Concetti base della OOP

- Incapsulamento dei dati
  - Il processo di nascondere i dettagli di definizione di oggetti, solo le interfacce con l'esterno sono visibili
- Ereditarietà
  - Gli oggetti sono definiti in una gerarchia ed ereditano dall'immediato parente caratteristiche comuni, che possono essere specializzate
- Astrazione
  - Il meccanismo con cui si specificano le caratteristiche peculiari di un oggetto che lo differenzia da altri
- Polimorfismo
  - Possibilità di eseguire funzioni con lo stesso nome che pure sono state specializzate per una particolare classe

Traduttori

- Servono a generare software.
  - Generano codice in *linguaggio macchina* a partire da codice scritto in un linguaggio di programmazione ad alto livello (ad es. C++, Java).
- Si distinguono in:
  - interpreti,
  - compilatori.

Compilatori

- Un compilatore è un programma che prende in input un codice *sorgente* e lo traduce fornendo in output un codice *oggetto*.
  - Per eseguire un programma *sorgente P*, scritto in un linguaggio di programmazione *L*:
    - *P* viene tradotto in un programma *Q* equivalente scritto in linguaggio macchina;
    - il programma *Q* viene eseguito.
- Esempi di linguaggi compilati:
  - C++, Pascal, Cobol, Fortran, …
- Il compilatore è legato all'architettura della macchina.

Interpreti

- Un interprete è un programma che prende in input un codice sorgente e, passo dopo passo, traduce ed esegue ogni singola istruzione.
  - La traduzione avviene dunque simultaneamente all'esecuzione.
  - Per ogni istruzione del programma sorgente *P*:
    - Viene tradotta la *singola* istruzione generando il corrispondente insieme di istruzioni in linguaggio macchina;
    - Si esegue il codice in linguaggio macchina e si passa all'istruzione sorgente successiva.
- Esempi di linguaggi interpretati:
  - VBasic, Lisp, Prolog, Java (inizialmente).

IQUADRO INGEGNERIA INFORMATICA

Compilatori vs. Interpreti

- Interpreti
  - Lenti nell'esecuzione.
  - Spesso si utilizza un linguaggio intermedio.
  - Progetti di dimensione limitata.
  - Facilità d'interazione col codice e velocità di sviluppo.
  - Facili da scrivere.
- Compilatori
  - Veloci nell'esecuzione.
  - Necessitano di poca memoria.
  - Permettono la compilazione separata.
  - Difficili da scrivere.

Background Java

- La motivazione originale per Java
  - La necessità di un linguaggio independent della piattaforma che poteva essere incluso (embedded) in varie apparecchi elettronici come tostapane e frigoriferi.
- Uno dei primi progetti sviluppati con Java
  - Un apparecchio per il controllo remoto di tipi palmare chiamato Star 7.
- Nello stesso tempo, il World Wide Web e l'Internet hanno guadagnato popularità.
  - James Gosling et. al. hanno capito che Java poteva essere usato per la programmazione per l'Internet (Initialmente Java Applet).

Background Java: Tecnologie

- La tecnologia Java è:
  - Un linguaggio di programmazione
  - Un ambiente di sviluppo
  - Un ambiente applicativo
  - Un ambiente di deployment

Tecnologie Java: Un linguaggio di programmazione

- Come Linguaggio di programmazione, Java permette di creare tutti tipi di applicazioni che potreste creare con altri linguaggi di programmazione convenzionali.

Tecnologie Java: Un ambiente di sviluppo

- Come ambiente di sviluppo, la tecnologia Java fornisce un insieme di strumenti :
  - Un compilatore (javac)
  - Un interprete (java)
  - Un generatore di documentazione (javadoc)
  - Un strumento per creare packaging di file .class (jar)
  - Molti altri ...

Tecnologie Java: Un ambiente applicativo & deployment

- Le applicazioni in tecnologia Java sono tipicamente dei programmi per tutti gli usi che funzionano su qualsiasi computer dove è installato il Java Runtime (JRE).
- Ci sono due principali ambiente di deployment:
  - 1. Il JRE (Java Runtime Enviroment) fornito con il Java Standard Edition Development Kit (SDK) che contiene un esteso supporto di classes per tutti packages Java
    - Classes per le base del linguaggio Java (String, Integer,…)
    - Classes per la Rete (InputStream, URL, …)
    - Classes lo sviluppo di interfacce grafiche GUI,…
  - 2. Il Web Browser
    - Tutti principali browser web supportano l'esecuzione di applicazione Java tramite un interpretre e ambiente Runtime (Java Applet)
    - Java Web Start,…

Caratteristiche importante di Java

- La Java Virtual Machine
- Il Garbage Collection
- L’esecuzione in sicurezza del Codice

Caratteristiche Java: La Java Virtual Machine

- Java Virtual Machine (JVM)
  - Una macchina virtuale che funziona emulando il software su una macchina reale
  - Fornisce le specificazioni della piattaforma hardware verso quale il codice Java deve essere compilato
- Java Bytecode
  - Un linguaggio macchina specifico che puo essere usato dalla Java Virtual Machine (JVM)
  - Indipendente di qualsiasi computer hardware, qualsiasi calcolatore con un interprete Java può eseguire un programma Java compilato (qualsiasi sia il computer che ha servito per la fase di compilazione)

Caratteristiche Java: Il Garbage Collection

- Il Garbage collection thread
  - È responsabile della liberazione di qualsiasi memoria che può essere liberata. Ciò accade automaticamente durante il corso della vita del programma Java.
  - Il programmatore è liberato dalla difficoltà di dovere liberare la memoria programmaticamente.

Caratteristiche Java: L'esecuzione in sicurezza del Codice

- L'esecuzione in sicurezza del Codice è ottenuta in Java attraverso l'implementazione del suo Java Runtime Environment (JRE).
- JRE
  - Esegue codice Java compilato (Java bytecode) per una JVM
    - realizza il caricamento del codice .class (attraverso il class loader)
    - verifica il codice (tramite il bytecode verifier)
    - esegue il codice.

Caratteristiche Java: L'esecuzione in sicurezza del Codice

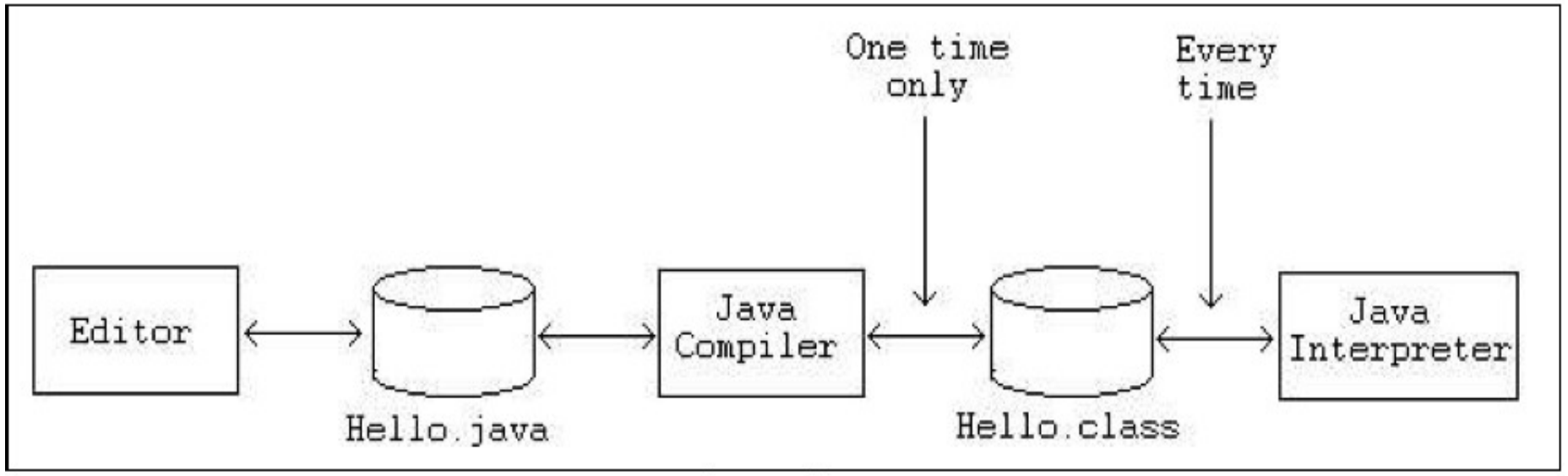
- Class Loader
  - Responsabile del caricamento di tutte le classes necessarie per il programma Java
  - Aggiunge la sicurezza separando i namespaces per le classes del file system locale da quelli che sono stati presi dalla rete
  - Dopo il carico di tutte le classes, la disposizione della memoria per l'eseguibile è allora determinata. Questo aggiunge protezione contro accessi non autorizzati alle zone protette del codice poiché la disposizione di memoria è determinata durante il tempo di esecuzione

Caratteristiche Java: L'esecuzione in sicurezza del Codice

- Bytecode verifier
  - Verifica il formato dei fragmenti di codice e verifica diretti di accesso e accessi illegali verso oggetti (public, protected, private,..)

Le Fase di un Programma Java

- La figura seguente descrive il processo di compilazione e di esecuzione di un programma Java



Le Fase di un Programma Java

Azione	Strumento da usare	Uscita
Scrivere il programma	Qualsiasi editore di testo	File con estensioni .java
Compilare il programma	Compilatore java (javac)	File con estensioni .class
Eseguire il programma	Interpretre java (java)	Output del programma

Ambiente di sviluppo Java

- Creare un programma Java usando un editore di testo e una console (Linux, Solaris, Unix, Windows, MacOS or altri OS)
- Fare la differenza tra errori di sintasse e errori Runtime
- Creare un programma Java con NetBeans

Primo Programma Java: Hello.java

```
1. public class Hello {
2.     /**
3.      * Mio Primo Programma Java
4.      */
5.     public static void main( String[] args ){
6.             // Stampa “Hello world” sullo schermo
7.             System.out.println(“Hello world”);
8.     }
9. }
```

Editore di Testo e Console

- Passo 1: Attivare il'editore di Testo
  - es.: Notepad, Wordpad
- Passo 2: Aprire una console
  - Windows: -> esegui: cmd
- Passo 3: Scrivere il codice sorgente Java nel editore di testo

Editore di Testo e Console

- Passo 4: Salvare il programma Java
  - Creare una cartella: per esempio: C:/mioprimojava
  - Nome del File: Hello.java (Nome della classe + .java)

Editore di Testo e Console

- Passo 5: Compilare il programma
  - Finestra della console
  - Andare nella cartella dove si trova il programma Java
    - Esempio: cd C:/mioprimojava
  - Compilare il programma Java, digitare il commando
    - javac [filename]
    - javac Hello.java
  - Durante la compilazione, javac aggiunge un file sul disco chiamato [filename].class (Hello.class) che contiene il Java bytecode.

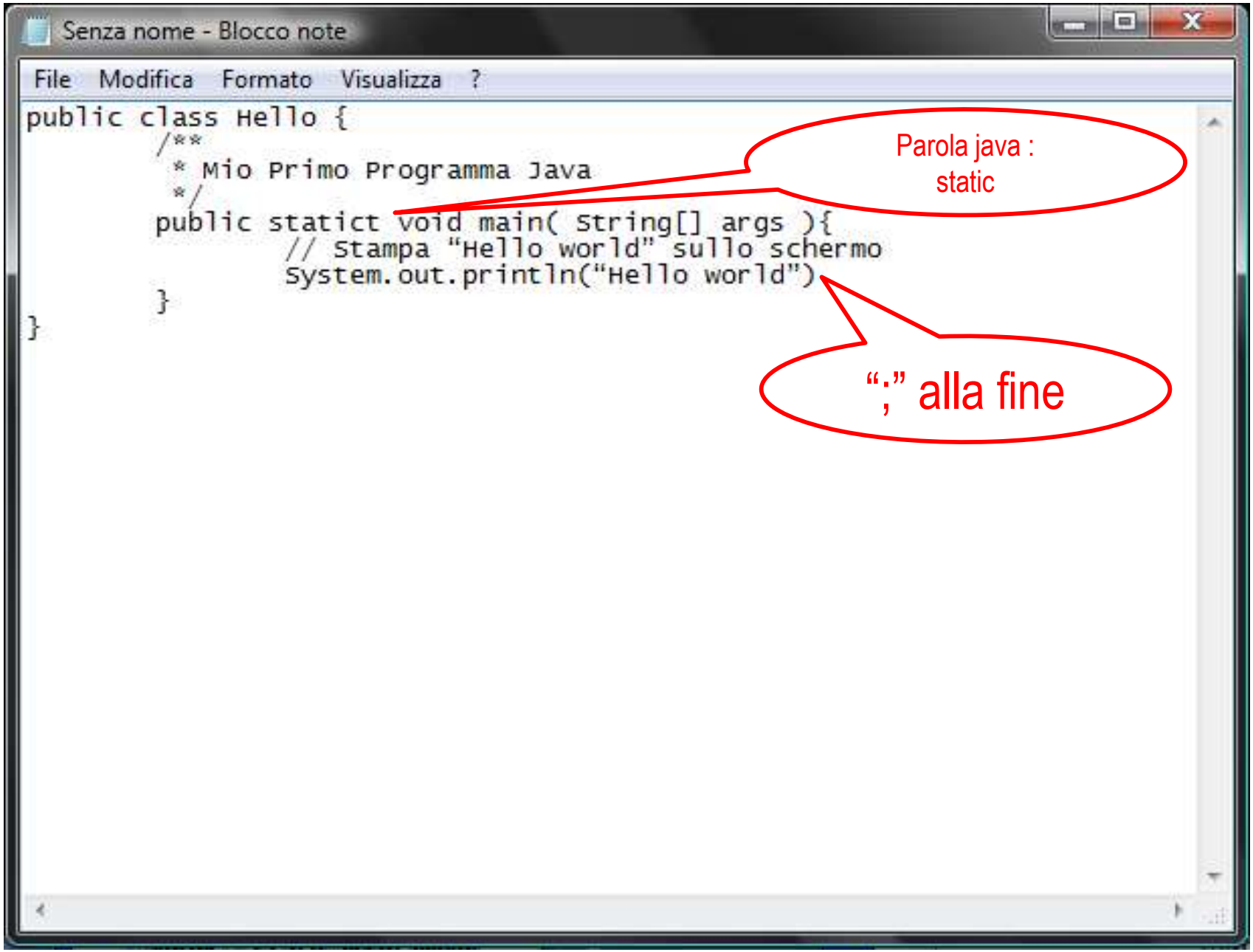
Editore di Testo e Console

- Passo 6: Eseguire il programma
  - Per Eseguire il programma, digitare il commando:
    - Java [filename senza l'estensione]
  - Nel nostro caso
    - Java Hello
  - Normalmente si ottiene sullo schermo la scritta:
    - "Hello World!"

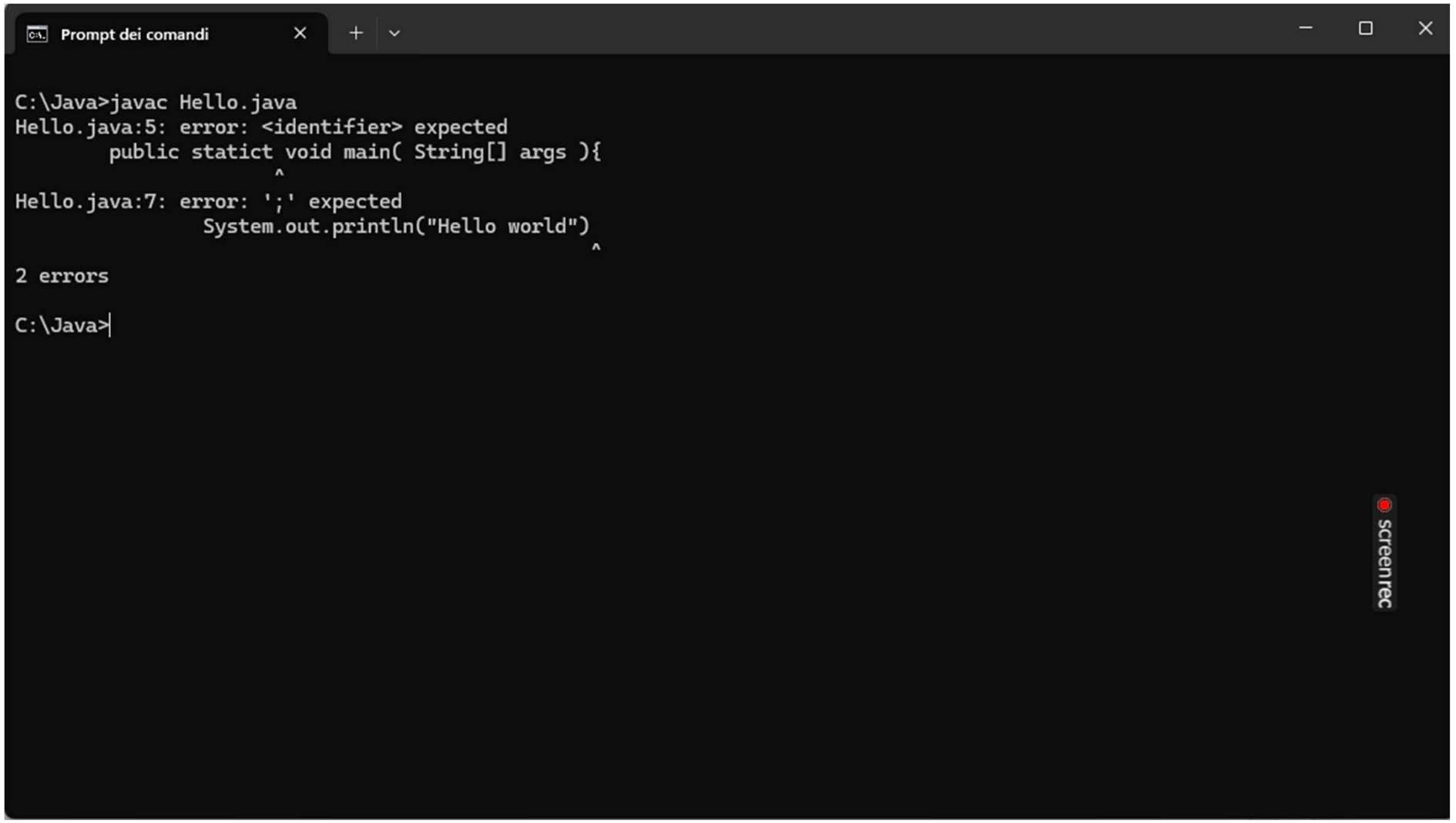
Errori: Errori di Sintassi

- Errori di sintassi
  - Sono spesso errori di scrittura
- Errori di sintassi comuni:
  - Un commando Java mal scritto (classe)
  - Dimenticare il “;” alla fine dell’istruzione (statement)

Errori: Errori di Sintassi



Errori: Errori di Sintassi



Errori: Errori Runtime

- Errori Runtime
  - Errori che non saranno visibili fino all'esecuzione del programma
  - Anche quando un programma compila correttamente, il programma può visualizzare risultati incorretti.

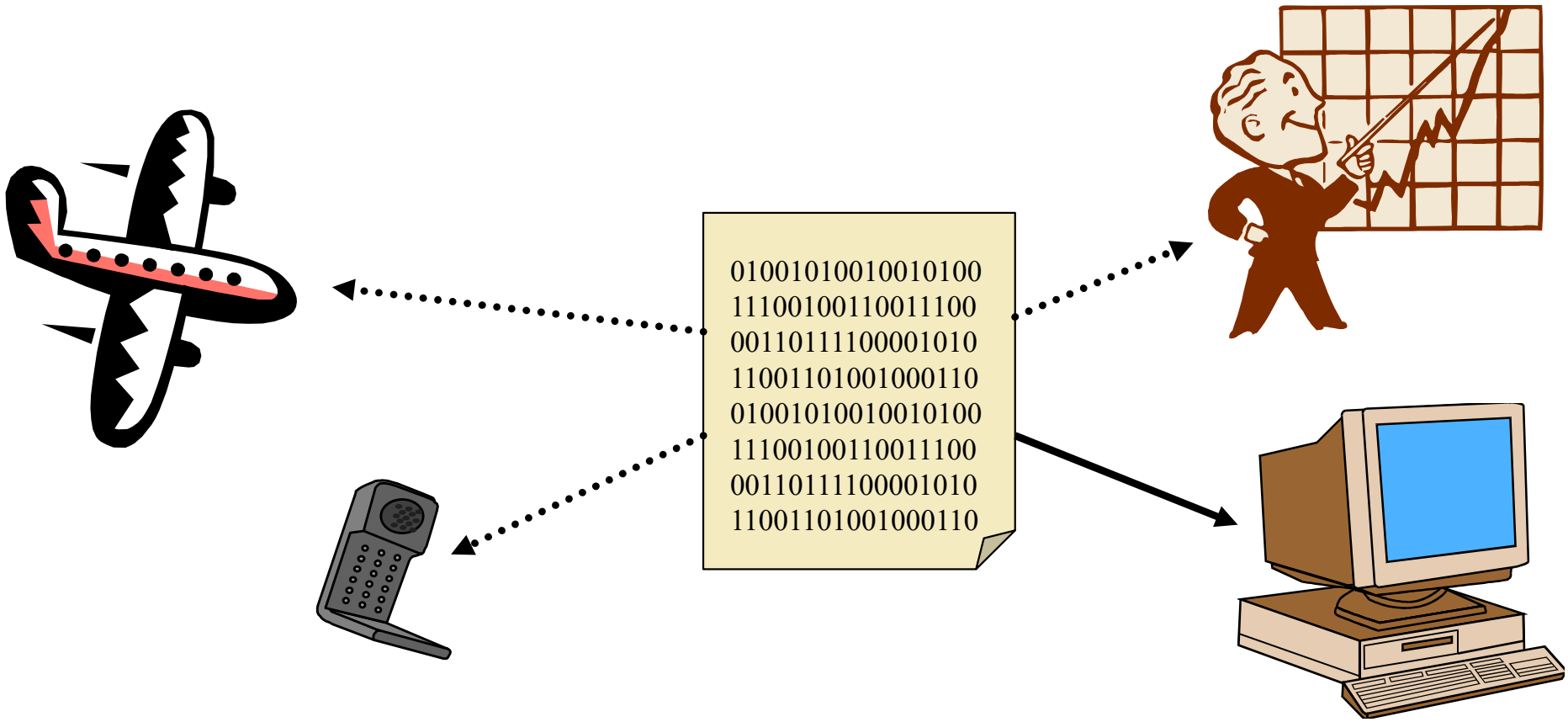
IQUADRO
INGEGNERIA INFORMATICA

Programmare ad oggetti

- La programmazione come attività di creazione di modelli.
- I concetti di classe, oggetto e scambio di messaggi.
- Un primo esempio di programma Java.
- Meccanismi di scrittura ed esecuzione di programmi Java.

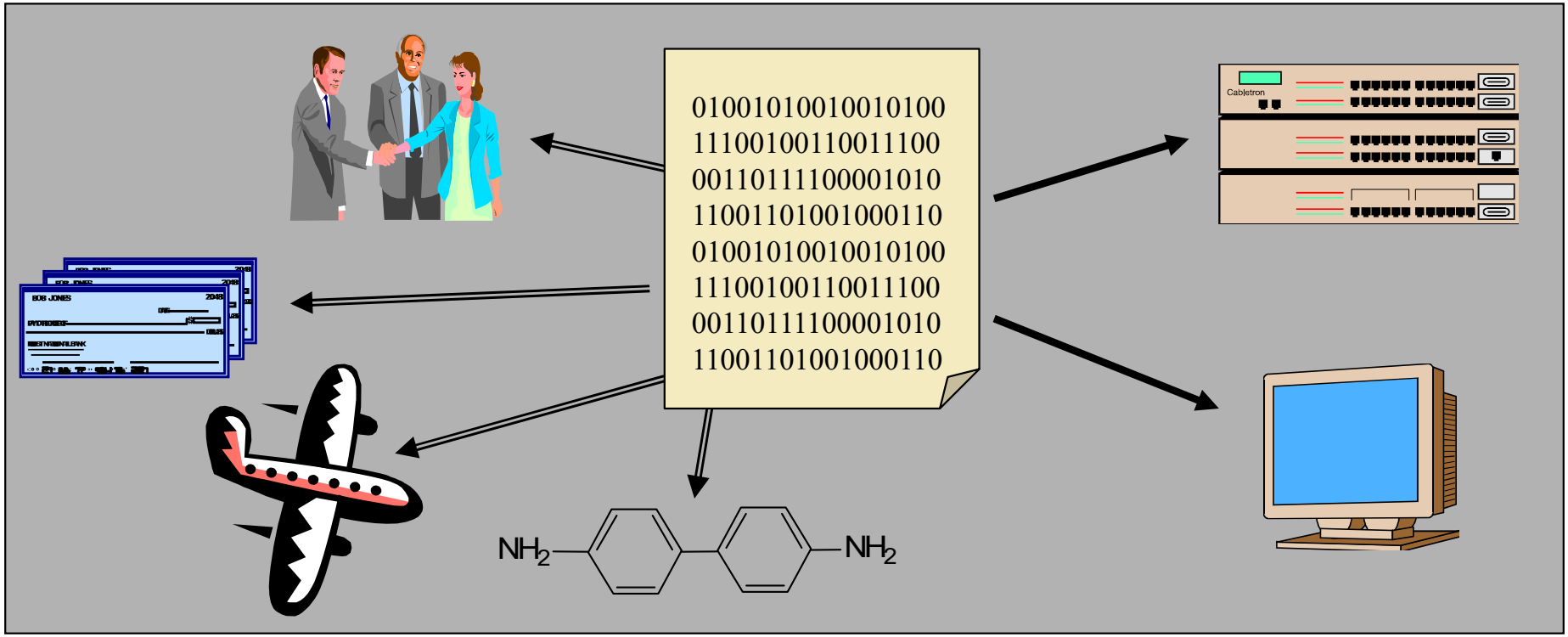
Programmi e mondo reale

- Il codice di un programma interagisce con calcolatore ed i suoi componenti così come con entità e processi esterni



I programmi come modelli

- Il codice di un programma modella altre cose
  - Interne: tastiera, dischi, reti, schermo
  - Esterne: piani di volo, molecole, transazioni bancarie

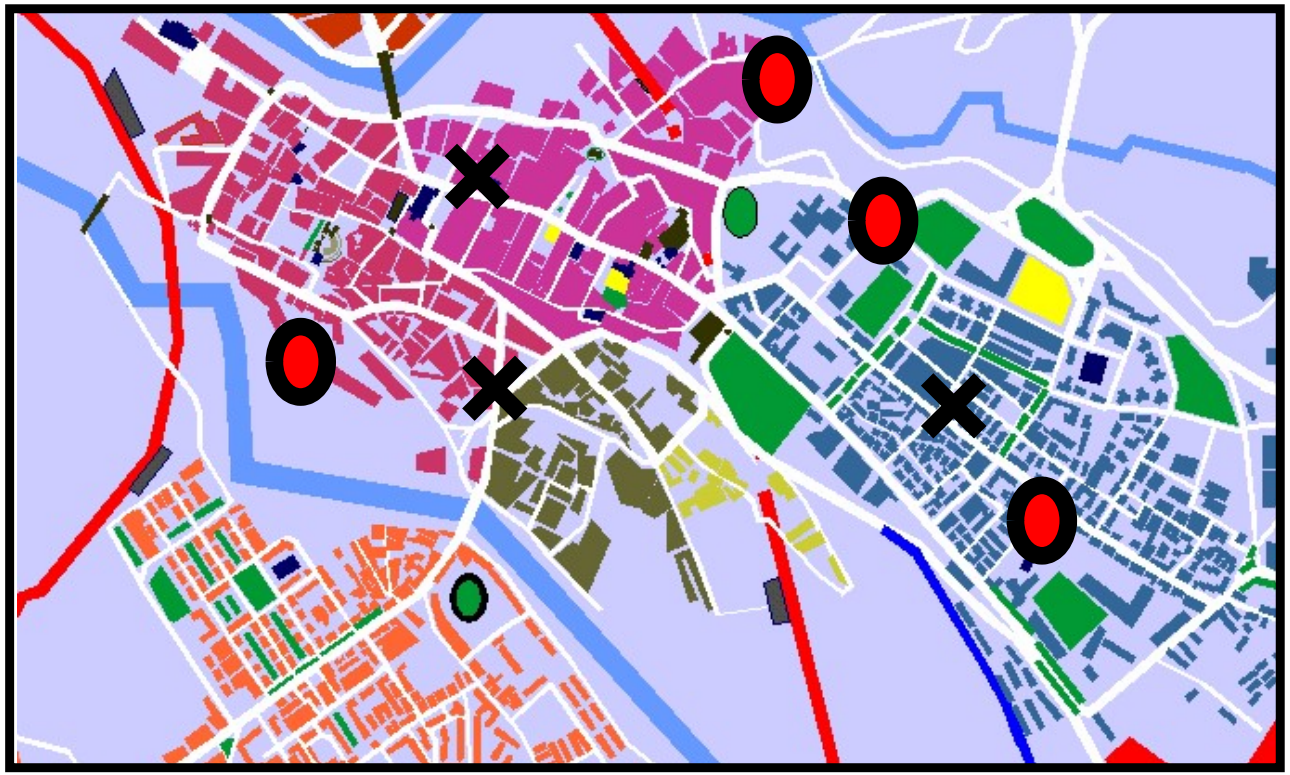


Modelli

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

Chiamate

Operatori



Elementi del modello

- Ogni modello è formato da **elementi** che rappresentano altre 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