

Programmazione 1 - Modulo C

Introduzione alla programmazione in Java

Marco Beccuti

Università degli Studi di Torino

Dipartimento di Informatica

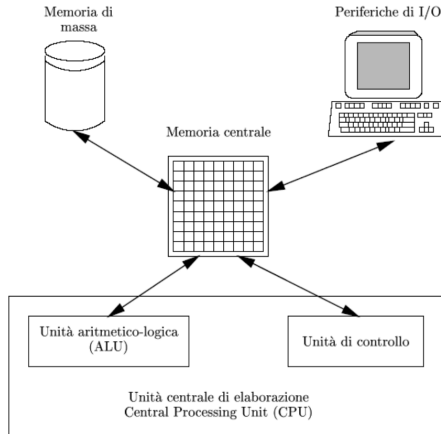
Settembre 2020



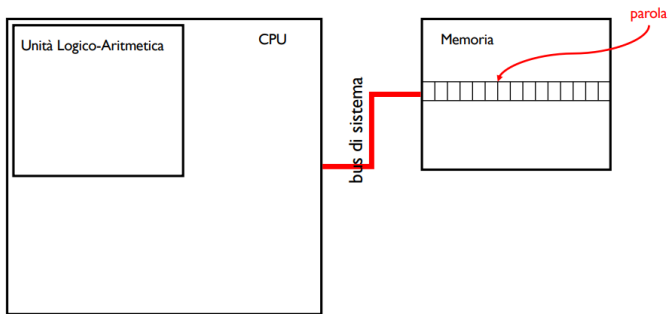
In queste slides

- Primi utilizzi dell'istruzione di assegnamento con modello della memoria;
- Primo esempio di programma Java, con compilazione ed esecuzione.
Creazione di un sorgente java;

Struttura (semplificata) di un calcolatore

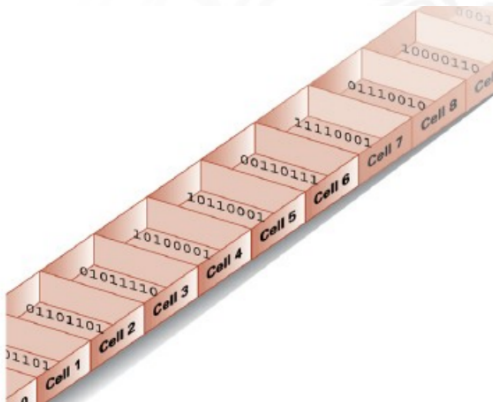


Struttura (semplificata) di un calcolatore

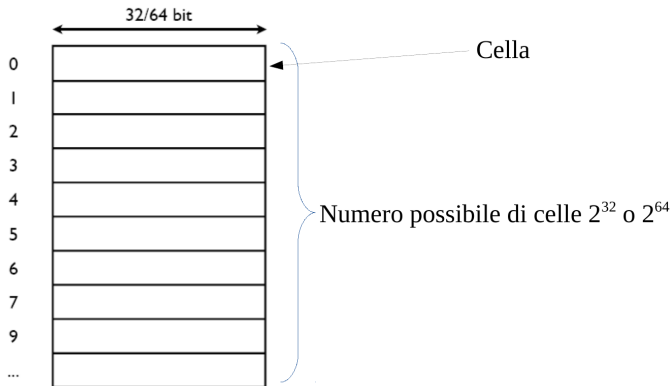


Memoria principale

- viene anche indicata con il termine: RAM (Random Access Memory);
- Fornisce la capacità di **memorizzare** le Informazioni (lettura o scrittura)
- La memoria principale è **volatile**
- Le componenti elementari sono aggregate tra di loro e formano delle strutture complesse dette **PAROLE o CELLE di memoria**.



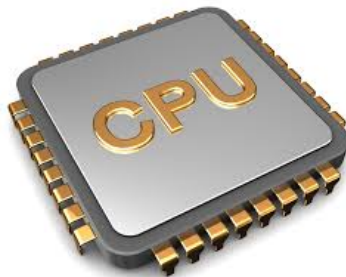
Memoria principale



Ogni dato o istruzione e' una sequenza di bit!!

CPU

- Permette di effettuare tutte le operazioni che riguardano il sistema e di modificarne lo stato;
- Può effettuare solo semplici operazioni:
 - ▶ operazioni aritmetiche e booleane;
 - ▶ operazioni di salto condizionato;
 - ▶ accesso in lettura scrittura della memoria;
 - ▶ ...



CPU

É composta da:

- **registri**: speciali locazioni di memoria interne alla CPU, molto veloci, a cui può accedere molto più rapidamente che alla memoria. I registri sono di due tipi:
 - ▶ i **registri speciali** utilizzati per scopi particolari (es. program counter, Registro Istruzione, ...);
 - ▶ i **registri di uso generale**
- **un'unità di controllo**: legge dalla memoria le istruzioni, se occorre legge anche i dati per l'istruzione letta, esegue l'istruzione e memorizza il risultato se c'è, scrivendolo in memoria o in un registro della CPU.
- **un'unità aritmetica e logica** (l'acronimo **ALU**) che si occupa di eseguire le operazioni logiche e aritmetiche.

CPU

- Intel 4040 é stato rilasciato sul mercato nel 1974 come successore dell'Intel 4004;
- Lo sviluppo dell'Intel 4040 é stato proposto da Federico Faggin che ne ha formulato l'architettura;
- Circa 70.000 operazioni al secondo

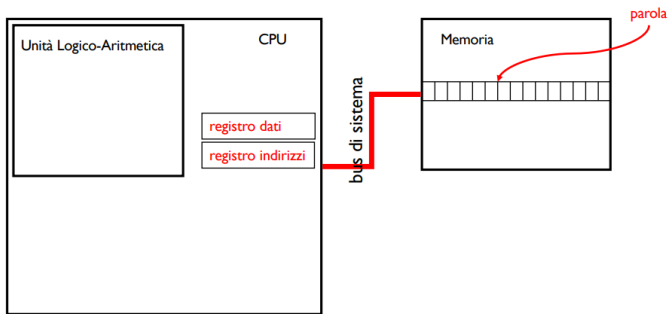


CPU

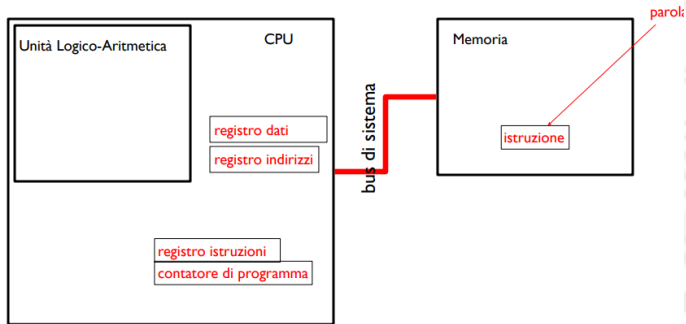
- Core i7;
- 14nanom= $14 * 10^{-9}$ metri;
- Circa 1 miliardo di transistor
- Circa 32 miliardi di operazioni al secondo.



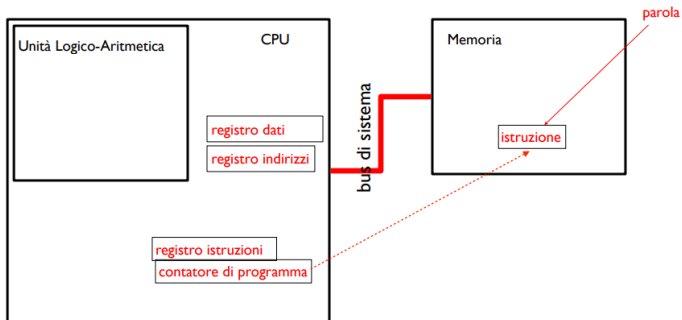
Struttura (semplificata) di un calcolatore



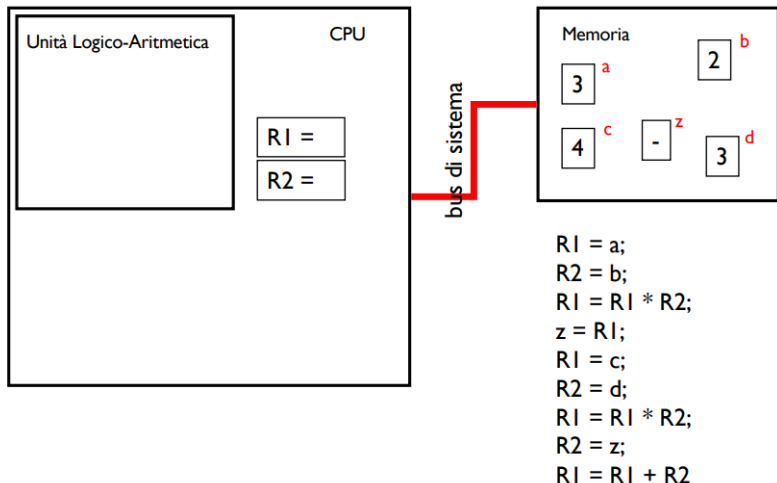
Struttura (semplificata) di un calcolatore



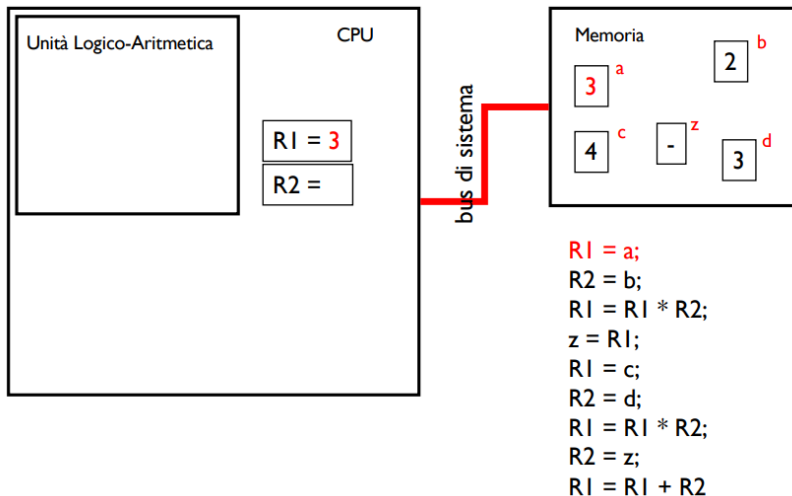
Il ciclo fetch-decode-execute



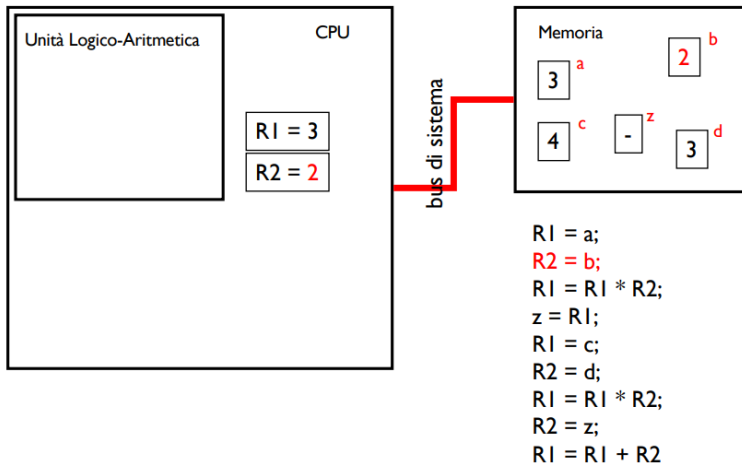
Esempio: valutazione dell'espressione $3 * 2 + 4 * 3$



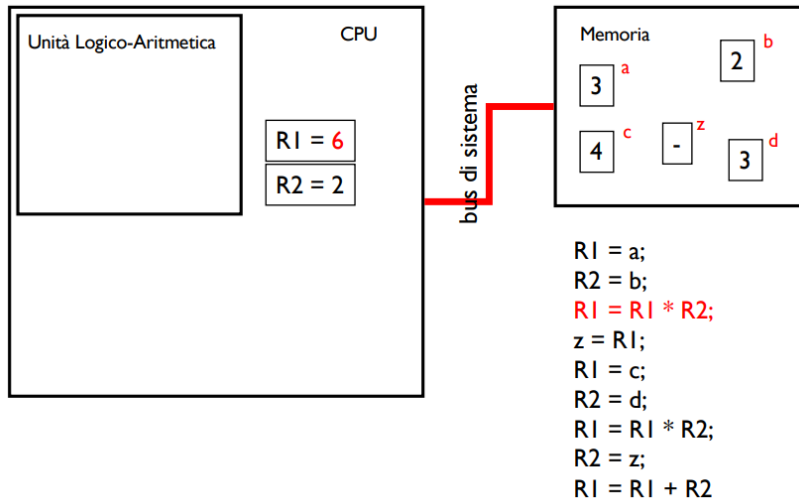
Esempio: valutazione dell'espressione $3 * 2 + 4 * 3$



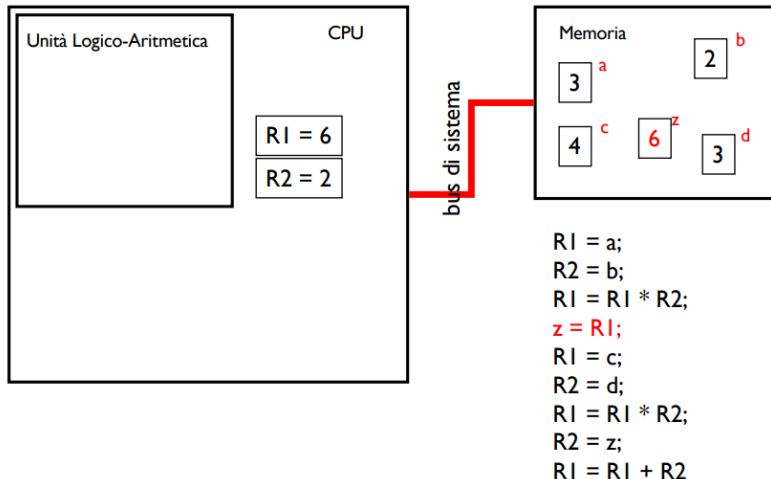
Esempio: valutazione dell'espressione $3 * 2 + 4 * 3$



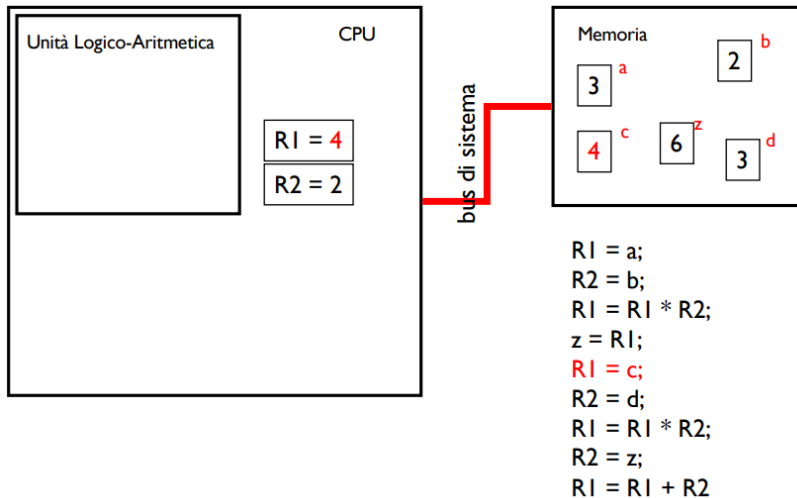
Esempio: valutazione dell'espressione $3 * 2 + 4 * 3$



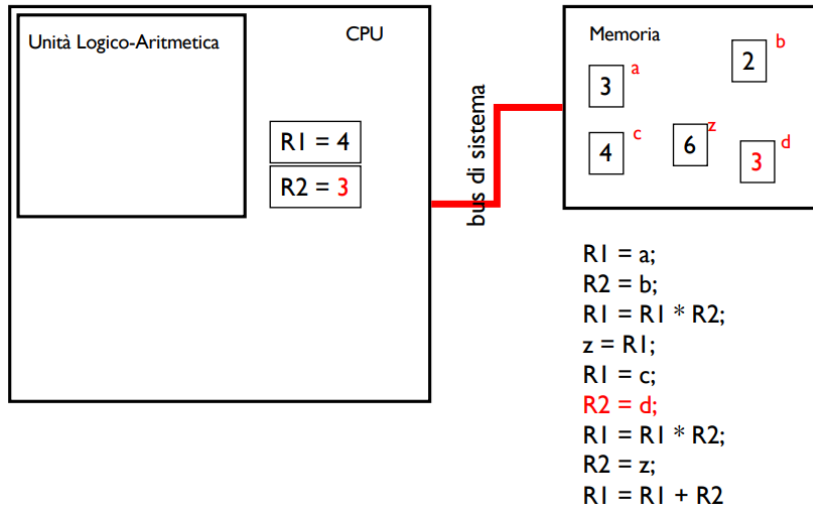
Esempio: valutazione dell'espressione $3 * 2 + 4 * 3$



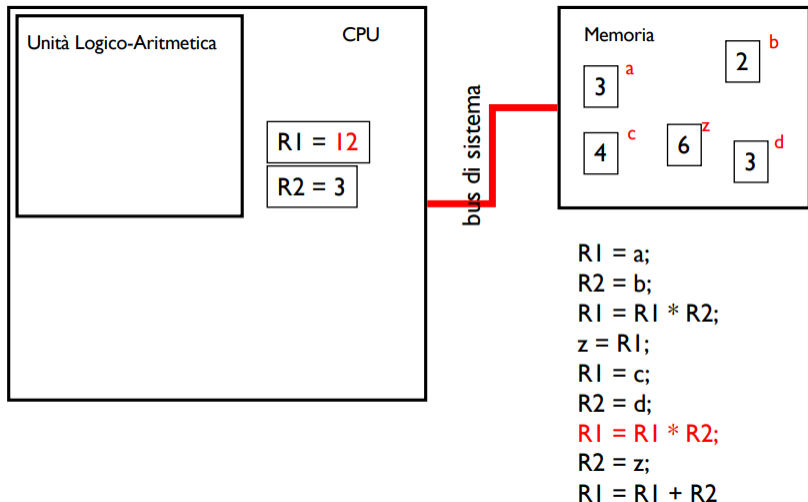
Esempio: valutazione dell'espressione $3 * 2 + 4 * 3$



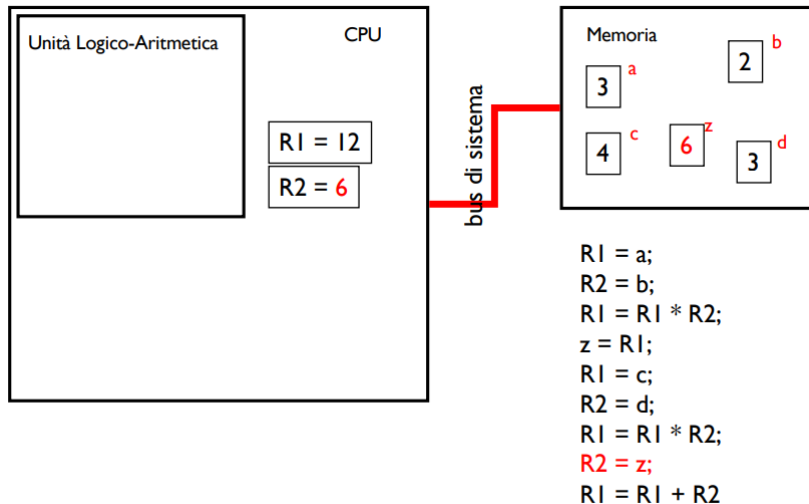
Esempio: valutazione dell'espressione $3 * 2 + 4 * 3$



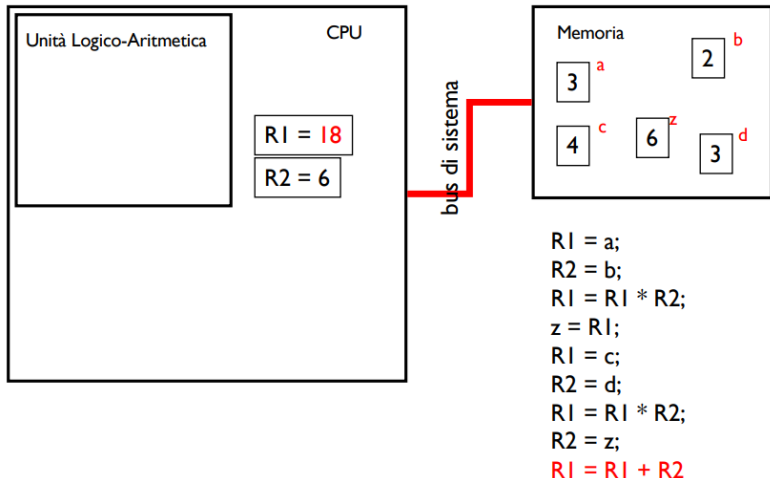
Esempio: valutazione dell'espressione $3 * 2 + 4 * 3$



Esempio: valutazione dell'espressione $3 * 2 + 4 * 3$



Esempio: valutazione dell'espressione $3 * 2 + 4 * 3$



Struttura (semplificata) di un programma

```
class <nome del programma> {  
    public static void main (String[] args){  
        <dichiarazioni di variabili>  
        <istruzioni>  
    }  
}
```

- Prima di essere utilizzata in un programma, una variabile deve essere **dichiarata**;
- una dichiarazione di variabile può seguire, nel testo del programma, delle istruzioni.

Breve Introduzione a Java

- É un linguaggio di programmazione ad alto livello, **orientato agli oggetti e a tipizzazione statica**
- progettato per essere il piú possibile indipendente dalla piattaforma hardware di esecuzione (tramite compilazione in **bytecode** prima e interpretazione poi da parte di una **Java Virtual Machine - JVM**)
- é il **bytecode** ad essere portabile: possiamo compilare una applicazione Java su Windows ed eseguirla su un S.O. Linux.

Breve Introduzione a Java

- **bytecode** é un linguaggio intermedio tra il linguaggio di programmazione ad alto livello e il linguaggio macchina;
- É così chiamato perché spesso le operazioni hanno un codice che occupa un solo **byte**.

			Byte Offset	
i = j + k;	1	ILOAD j // i = j + k	0	0x15 0x02
if (i == 3)	2	ILOAD k	2	0x15 0x03
k = 0;	3	IADD	4	0x60
else	4	ISTORE i	5	0x36 0x01
j = j - 1;	5	ILOAD i // if (i < 3)	7	0x15 0x01
	6	BIPUSH 3	9	0x10 0x03
	7	IF_ICMPEQ L1	11	0x9F 0x00 0x0D
	8	ILOAD j // j = j - 1	14	0x15 0x02
	9	BIPUSH 1	16	0x10 0x01
	10	ISUB	18	0x64
	11	ISTORE j	19	0x36 0x02
	12	GOTO L2	21	0xA7 0x00 0x07
13 L1:	13	BIPUSH 0 // k = 0	24	0x10 0x00
14	14	ISTORE k	26	0x36 0x03
15 L2:	15		28	

https://en.wikipedia.org/wiki/Java_bytecode_instruction_listings

Breve Introduzione a Java

- La JVM specifica per un S.O ed un'architettura hardware é implementata dal **Java Runtime Environment (JRE)**;
- **Java Development Kit (JDK)** include la JRE , ma fornisce inoltre un insieme di software per sviluppare le applicazioni basate su Java;
- JDK é necessaria per sviluppare nuove applicazioni Java: contiene il **compilatore Java**

Breve Introduzione a Java

- Dove scaricare JDK:

<https://www.oracle.com/java/technologies/javase-downloads.html>

- Le informazioni sull'installazione per i vari sistemi operati sono reperibili nella stessa pagina.

Text editor

- Esisto diversi text editor che posso essere usati per scrivere un programma Java
- In linux potete usare **TextEditor, Code::Blocks, ...**
- In Windows potete usare **Notepad++** (scaricabile gratuitamente da: <https://notepad-plus-plus.org/>)
- In MacOS potete usare **Sublime Text, ...**

Ambienti di sviluppo integrati (IDE)

- Esistono software che, in fase di programmazione, supportano i programmatori nello sviluppo e debugging del codice sorgente di un programma;
- **NetBean IDE** (scaricabile gratuitamente da: <https://netbeans.org/>)
- **Eclipse IDE** (scaricabile gratuitamente da: <https://www.eclipse.org/downloads/>)

Struttura (semplificata) di un programma

```
class Saluto {  
    public static void main (String[] args){  
        System.out.println ("Ciao.");  
    }  
}
```

- Questo programma é il contenuto di un file di testo chiamato **Saluto.java**;
- Si puo' compilare con il comando:

```
javac Saluto.java
```

- Dalla compilazione si ottiene il file bytecode **Saluto.class**
- Può essere eseguito con il comando:

```
java Saluto
```

Dichiarazioni di variabili: sintassi

- Le variabili sono dichiarate assieme al loro **tipo**, per esempio:

int n;

int i,j; equivalente a *int i; int j;*

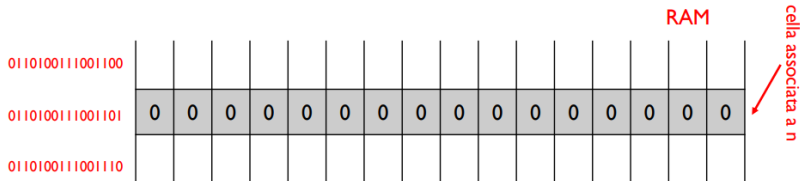
float pi;

- Dal punto di vista sintattico, le variabili sono **identificatori**, sequenze di caratteri alfanumerici che non iniziano con una cifra.

Alcuni tipi primitivi	Valori
boolean	true, false
char	caratteri UNICODE
int	interi 32 bit
float	numeri in virgola mobile, 32 bit
double	numeri in virgola mobile, 64 bit

Dichiarazioni di variabili: semantica

```
class Esempio {  
    public static void main (String[] args){  
        int n;  
        double pi;  
        ...    }  
    }
```



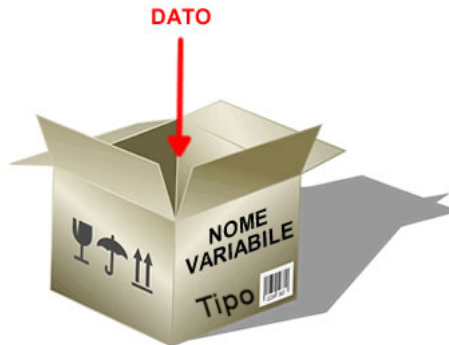
Dichiarazioni di variabili: semantica

- Al momento della dichiarazione, ad una variabile viene associata una **cella di memoria** (o un gruppo di celle);
- I tipi primitivi permettono al compilatore di calcolare quante celle di memoria allocare ad una variabile;
- Il valore di una variabile dichiarata ma non esplicitamente inizializzata **deve essere considerato indefinito**

Morale: non usare variabili senza dichiararle ed inizializzarle.

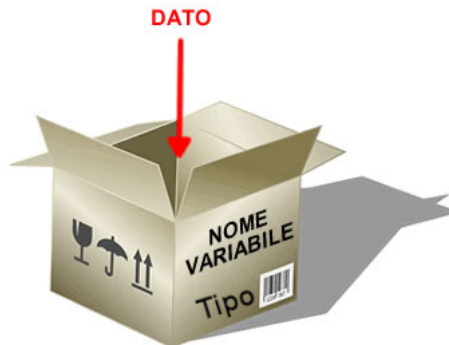
Variabile

- Una variabile è caratterizzata da:
 - ▶ un **nome** (che è un identificatore);
 - ▶ una **dimensione** (determinata dal tipo con il quale è stata dichiarata);
 - ▶ un **valore** (del tipo della variabile).



Variabile

- Su una variabile sono disponibili le seguenti operazioni:
 - ▶ **dichiarazione** introduce un nome ed un tipo (e, di conseguenza, una dimensione) per la variabile;
 - ▶ **scrittura** associa un valore alla variabile;
 - ▶ un **lettura** recupera il valore della variabile.



Istruzione di assegnamento

- Formato generale:

$\langle \text{variabile} \rangle = \langle \text{espressione} \rangle;$

dove $\langle \text{espressione} \rangle$ ha lo stesso tipo di $\langle \text{variabile} \rangle$

- Esecuzione di un'istruzione di assegnamento $x = E$

- ▶ si valuta l'espressione E ; se la valutazione termina si ottiene un valore v ;
- ▶ il valore v viene scritto nella cella di memoria (eventualmente più di una) associata alla variabile x al momento della sua dichiarazione.

- Osservazione:

- ▶ Sulla **sinistra** di un assegnamento, una variabile indica una cella di memoria;
- ▶ sulla **destra** di un assegnamento, una variabile rappresenta il valore scritto (al momento della valutazione) nella cella associata.

Espressioni di tipo primitivo

- Espressioni di tipo numerico

se E, E' sono espressioni di tipo int, anche:

- ▶ $E + E'$ (somma)
- ▶ $E * E'$ (prodotto)
- ▶ E / E' (divisione)
- ▶ $E \% E'$ (resto della divisione)

sono espressioni di tipo int.

Espressioni di tipo primitivo

- Espressioni di tipo boolean

se E, E' sono espressioni di tipo int, allora:

- ▶ $E == E'$
- ▶ $E > E'$
- ▶ $E < E'$
- ▶ $E! = E'$
- ▶ $E \leq E'$
- ▶ $E \geq E'$

sono espressioni di tipo boolean.

- Le espressioni booleane possono essere composte in and ($\&\&$) o in or ($\|\|$)

Tavole di verità

Operatore not (!)

P	NOT P
1	0
0	1

← TAVOLA DI VERITÀ

Tavole di verità

Operatore AND (&&)

P	Q	P AND Q
1	1	1
1	0	0
0	1	0
0	0	0

Tavole di verità

Operatore OR (\vee)

P	Q	P OR Q
1	1	1
1	0	1
0	1	1
0	0	0

Semantica delle istruzioni di assegnamento: esempio

Per eseguire l'istruzione

$$x = (-3 + 5) * 4;$$

avendo dichiarato int x:

(1) Valutare l'espressione $(-3 + 5) * 4$:

valutare l'espressione $-3 + 5$

valutare l'espressione -3 ,

valutare l'espressione 5 , ottenendo valore 3

ottenendo valore -3

valutare l'espressione 5 , ottenendo valore 5

ottenendo valore 2

valutare l'espressione 4 , ottenendo valore 4

ottenendo valore 8

(2) scrivere il valore 8 nella cella di memoria associata a x al momento della sua dichiarazione.

Semantica delle istruzioni di assegnamento: esempio

Per valutare l'espressione

$$x = (-3 + 5) * 4;$$

avendo dichiarato int x:



(1) Valutare l'espressione $(-3 + 5) * 4$:

valutare l'espressione $-3 + 5$

valutare l'espressione -3 ,

valutare l'espressione 5 , ottenendo valore 3

ottenendo valore -3

valutare l'espressione 5 , ottenendo valore 5

ottenendo valore 2

valutare l'espressione 4 , ottenendo valore 4

ottenendo valore 8

(2) scrivere il valore 8 nella cella di memoria associata a x al momento della sua dichiarazione.

Semantica delle istruzioni di assegnamento: esempio

$$\begin{array}{rcl} 3 \rightarrow 3 & & \\ \hline -3 \rightarrow -3 & 5 \rightarrow 5 & \\ \hline (-3 + 5) \rightarrow 2 & 4 \rightarrow 4 & \\ \hline (-3 + 5) * 4 \rightarrow 8 & & \end{array}$$



Dichiarazione di variabili con inizializzazione

- La dichiarazione

```
int x = E;
```

dichiara la variabile x associandole il valore dell'espressione E .

- È un'abbreviazione di

```
int x;  
x = E;
```