

Laboratorio di Programmazione II

Corso di Laurea in Bioinformatica
Dipartimento di Informatica - Università di Verona

Sommario

Laboratorio
di Program-
mazione
II

Tipi di dati
primitivi

Strutture di
Controllo

Esercizi Cicli

Cicli
Annidati

Esercizi Cicli
Annidati

- tipi di dati primitivi
- condizioni
- cicli
- cicli annidati

Tipi di dati primitivi

Laboratorio
di Program-
mazione
II

Tipi di dati
primitivi

Strutture di
Controllo

Esercizi Cicli

Cicli
Annidati

Esercizi Cicli
Annidati

Tipi di dati primitivi

Tipi di dati primitivi

- per alcuni tipi di dati Java mette a disposizione una rappresentazione molto efficiente
- Esempi: dati numerici (interi, numeri decimali, etc.), booleani, caratteri
- Tipi di dati primitivi NON sono degli oggetti (ma esistono degli oggetti per rappresentarli)
- Variabili non contengono riferimenti ad oggetti ma direttamente i valori (e.g., numero intero)
- Possono essere manipolati tramite simboli standard (e.g., +, *, /)
- Gli interi vengono rappresentati tramite il tipo primitivo *int*

Esempio tipi di dati primitivi

Laboratorio
di Programmazione
II

Tipi di dati
primitivi

Strutture di
Controllo

Esercizi Cicli

Cicli
Annidati

Esercizi Cicli
Annidati

Example (Esempio interi)

```
int a = 1;
int b = 1;
a = a+b;
a = a+1;
System.out.println(a); //stampa 3
int c = a/2; //divisione tra interi
System.out.println(c); //stampa 1
int d = a%2; //resto della divisione
System.out.println(d); //stampa 1
```

Assegnamento composto

Laboratorio
di Programmazione
II

Tipi di dati
primitivi

Strutture di
Controllo

Esercizi Cicli

Cicli
Annidati

Esercizi Cicli
Annidati

Operatori di assegnamento composto

- Spesso si aggiorna il valore di una variabile come segue: $x = x \text{ op valore}$
- x e' una variabile; op e' un operatore (+, *, /, etc.) e $valore$ e' un valore
- $a = a + 2$ incrementa il valore di a di due unita'
- Esistono speciali operatori per scrivere queste operazioni in maniera' piu' diretta
- $x \text{ op} = \text{valore}$
- $a += 2$ incrementa il valore di a di due unita'
- $op = +, -, *, /, \%$
- Per incrementare (decrementare) di uno: $x ++$ ($x --$)

Operatori composti

Example (Esempio operatori composti)

```
int a = 1;
a += 2; // a = a + 2;
System.out.println(a); //stampa 3
a *= 3; // a = a * 3
System.out.println(a); //stampa 9
a++; // a = a + 1; a += 1;
System.out.println(a); //stampa 10
```

Altri tipi di interi

long, short e byte

- int usa 32 bit per la rappresentazione, hanno una rappresentazione limitata (-2 miliardi a +2 miliardi circa).
- long usa 64 bit
- posso assegnare int a long, non perdo informazioni
- posso usare il *cast* per assegnare long ad int (posso perdere informazioni)
- i letterali long sono seguiti da una L
- short: interi ad 16 bit, byte: interi a 8 bit
- per short e byte valgono regole di assegnazione simili a quelle per interi e long: se non perdo informazioni ok, altrimenti devo usare *cast*.

Tipo primitivo long

Example (Esempio long)

```
long l = 10; //letterale intero
int o = (int) l; //ok, ma necessario cast
l = 10000000000L; //devo usare L
System.out.println(l); //stampa 10000000000
long la = o; //ok
int w = (int) l; //ok, ma perdo info
System.out.println(w); //stampa 1410065408
```

Altri tipi di dati primitivi

float e double

- float modella numeri in virgola mobile (precisione di circa 7 cifre, range di valori molto ampio)
- double doppia precisione rispetto a float (circa 15 cifre)
- I float (e double) usano la notazione scientifica (per stampa e letterali)
- 3.4028235E38 (3.4028235×10^{38})
- i letterali float sono seguiti da una f, i letterali double sono seguiti da una d
- gli operatori sono gli stessi degli int, tranne %; l'operatore / rappresenta la divisione tra reali
- per double e float valgono regole di assegnazione simili a long ed int: se non perdo informazioni ok, altrimenti devo usare *cast*.

Tipi primitivi float e double

Example (Esempio float e double)

```
double dd = 3.5E38d; //maggiore del massimo float;
System.out.println(dd); //stampa 3.5E38
float ff = (float) dd;
System.out.println(ff); //stampa infinity
dd = 3;
System.out.println(dd/2); //stampa 1.5
```

I tipi boolean e char

boolean e char

- boolean rappresenta valori booleani
- letterali sono true e false
- char rappresenta caratteri secondo lo standard Unicode a 16-bit
- i letterali sono caratteri ('a') e simboli speciali ('@')

Example (Esempio float e double)

```
boolean t = true;
System.out.println(t); //stampa true
char c = '@';
System.out.println(c); //stampa @
```

Oggetti per tipi primitivi

Laboratorio
di Programmazione
II

Tipi di dati
primitivi

Strutture di
Controllo

Esercizi Cicli

Cicli
Annidati

Esercizi Cicli
Annidati

Classi associate ai tipi primitivi

- Ogni tipo primitivo ha associata una classe
- e.g., int associato a Integer, double a Double, etc.
- alcuni metodi delle classi corrispondenti sono molto utili per la gestione dei tipi primitivi

Example (Esempio di uso classi associate)

```
int i = Integer.parseInt("1976");
System.out.println(i+1); //stampa 1977
double d = Double.parseDouble("3.14");
int max = Integer.MAX_VALUE;
System.out.println(max); //stampa il massimo intero
```

Vedere il file [EsempioTipiPrimitivi.java](#)

Strutture di Controllo

Laboratorio
di Program-
mazione
II

Tipi di dati
primitivi

**Strutture di
Controllo**

Esercizi Cicli

Cicli
Annidati

Esercizi Cicli
Annidati

Selezione Semplice

Laboratorio
di Programmazione
II

Tipi di dati
primitivi

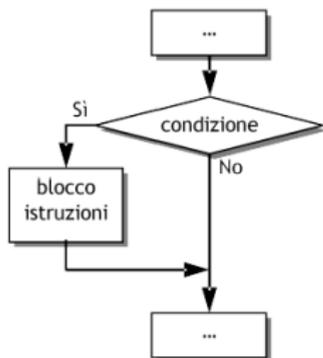
Strutture di
Controllo

Esercizi Cicli

Cicli
Annidati

Esercizi Cicli
Annidati

Selezione Semplice



Pseudocodice

```
....  
SE condizione  
    ALLORA  
        blocco istruzioni  
FINESE  
....
```

Java

```
....  
if (condizione )  
{  
    blocco istruzioni  
}  
....
```

Selezione a due vie

Laboratorio
di Programmazione
II

Tipi di dati
primitivi

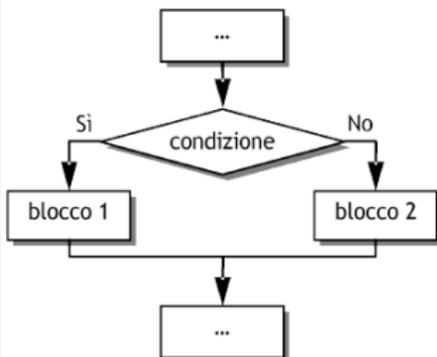
Strutture di
Controllo

Esercizi Cicli

Cicli
Annidati

Esercizi Cicli
Annidati

Selezione a due vie



Pseudocodice

```
....  
SE condizione  
    ALLORA  
        blocco 1  
    ALTRIMENTI  
        blocco 2  
FINESE  
....
```

Java

```
....  
if (condizione) {  
    blocco 1  
}  
else {  
    blocco 2  
}  
....
```

Ciclo a condizione iniziale

Laboratorio
di Programmazione
II

Tipi di dati
primitivi

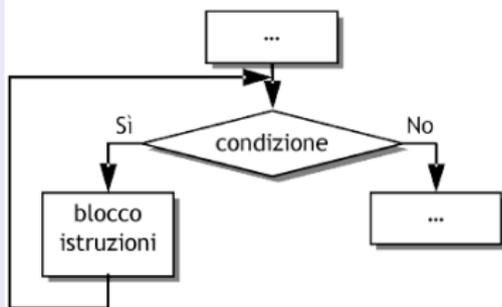
Strutture di
Controllo

Esercizi Cicli

Cicli
Annidati

Esercizi Cicli
Annidati

Ciclo a condizione iniziale



Pseudocodice

```
....  
QUANDO condizione ESEGUI  
    blocco istruzioni  
RIPETI  
....
```

Java

```
....  
while (condizione){  
    blocco istruzioni  
}  
....
```

Ciclo a condizione finale

Laboratorio
di Programmazione
II

Tipi di dati
primitivi

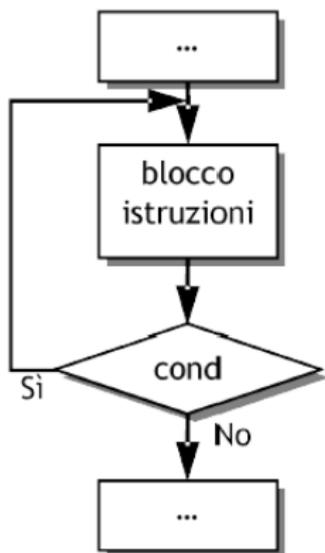
Strutture di
Controllo

Esercizi Cicli

Cicli
Annidati

Esercizi Cicli
Annidati

Ciclo a condizione finale



Pseudocodice

```
....  
ESEGUI  
    blocco istruzioni  
QUANDO condizione  
....
```

Java

```
....  
do {  
    blocco istruzioni  
}while (condizione)  
....
```

Ciclo FOR

Laboratorio
di Programmazione
II

Tipi di dati
primitivi

Strutture di
Controllo

Esercizi Cicli

Cicli
Annidati

Esercizi Cicli
Annidati

Ciclo a FOR

```
for (inizializzazione; condizione; incremento)  
    istruzione
```

Semanticamente equivalente a:

```
{  
    inizializzazione;  
    while (condizione) {  
        istruzione  
        incremento;  
    }  
}
```

Ciclo FOR esempio

Laboratorio
di Programmazione
II

Tipi di dati
primitivi

Strutture di
Controllo

Esercizi Cicli

Cicli
Annidati

Esercizi Cicli
Annidati

Ciclo a FOR esempio

```
for (int i = 0; i<=100; i++){  
    System.out.println("*")  
}
```

Semanticamente equivalente a:

```
{  
    int i=0;  
    while (i<=100) {  
        System.out.println("*");  
        i++;  
    }  
}
```

Schema ciclo while lettura dati

Laboratorio
di Program-
mazione
II

Tipi di dati
primitivi

Strutture di
Controllo

Esercizi Cicli

Cicli
Annidati

Esercizi Cicli
Annidati

Ciclo while lettura dati

```
while( dato disponibile){  
  processa dato  
  leggi nuovo dato  
}
```

Example (Numero interi letti da input)

```
...  
Scanner sc = new Scanner(System.in);  
int numInt = 0;  
while (sc.hasNextInt()){  
    numInt++;  
    sc.nextInt();  
}  
System.out.println("numero di interi immessi "+numInt);  
...
```

Cicli I

- 1 Scrivere un programma che calcoli il numero di interi positivi inseriti da tastiera.
Soluzione: [NumeroInteriPositivi.java](#)
- 2 Scrivere un programma che calcoli la media dei numeri reali inseriti da tastiera.
Soluzione: [MediaReali.java](#)
- 3 Scrivere un programma che stampi il MCD di due numeri interi dati in input
Soluzione: [MassimoComunDivisore.java](#)
 - considerare come possibile MCD il minimo dei due numeri e poi decrementare MCD fino a che non divide esattamente entrambi i numeri in input.

Cicli II

- 4 Scrivere un programma che calcoli il massimo dei numeri reali inseriti da tastiera (assumere che venga inserito almeno un reale).

Soluzione: MassimoReali.java

- 5 Scrivere un programma che stampi il massimo dei numeri reali inseriti da tastiera caso generale

Soluzione: MassimoGenerale.java

- utilizzare una variabile booleana per decidere se è stato inserito almeno un dato valido

Cicli annidati

- Il corpo di un ciclo può contenere a sua volta un ciclo, chiamato ciclo annidato.
- È possibile annidare un qualunque numero di cicli.

Example (stampa della tavola pitagorica)

```
public class TavolaPitagorica {
    static final int NMAX = 10;

    public static void main (String[] args) {
        int riga, colonna;

        for (riga = 1; riga <= NMAX; riga++) {
            for (colonna = 1; colonna <= NMAX; colonna++)
                System.out.print(riga * colonna + " ");
            System.out.println();
        }
    }
}
```

Esercizi cicli annidati I

Laboratorio
di Program-
mazione
II

Tipi di dati
primitivi

Strutture di
Controllo

Esercizi Cicli

Cicli
Annidati

Esercizi Cicli
Annidati

Media

- 1 Scrivere un programma che produca la seguente stampa:

```
*  
**  
***
```

dove l'altezza deve essere un parametro passato da input

Soluzione: StampaAsterischi.java

- 2 Scrivere un programma che produca la seguente stampa (l'altezza della piramide è un dato passato da input):

```
*  
***  
*****
```

trovare la relazione che lega il numero della riga ed il numero di spazi ed asterischi da stampare per ciascuna riga

Soluzione: StampaPiramideAsterischi.java

