



Metodologie di Programmazione

- Lezione 4: Conversioni di tipo

Lezione 4:

Sommario



SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

- Conversioni di tipo
- Caratteri e stringhe
- Booleani
- Operatori di confronto
- Consapevolezza del tipo di dato
- Conversione esplicita, cast esplicito e implicito

Da stringhe a tipi primitivi

- Supponiamo di voler utilizzare l'input di un programma per effettuare dei calcoli (es. somme)
- Come convertiamo da **stringa** a **intero**?

```
public class SommaInteri
{
    public static void main(String[] args)
    {
        int a = Integer.parseInt(args[0]);
        int b = Integer.parseInt(args[1]);
        System.out.print("La somma vale: ");
        System.out.println(a+b);
    }
}
```

Da stringhe a tipi primitivi

- Supponiamo di voler utilizzare l'input di un programma per effettuare dei calcoli (es. somme)
- Come convertiamo da **stringa** a **intero**?
- E da **stringa** a **double**?

```
public class SommaVirgolaMobile
{
    public static void main(String[] args)
    {
        double a = Double.parseDouble(args[0]);
        double b = Double.parseDouble(args[1]);
        System.out.print("La somma vale: ");
        System.out.println(a+b);
    }
}
```

Da dati primitivi a stringhe

- Java definisce l'operatore + sul tipo di dato "built-in" String
- Quando usiamo + con almeno un operando String, Java converte automaticamente l'altro operando a String, restituendo una stringa:

```
public class PensieroProfondo
{
    public static void main(String[] args)
    {
        String s = "La risposta alla domanda fondamentale sulla vita, l'universo e tutto quanto e' ";
        int v = 42;

        String risposta = s+v;
        System.out.println(risposta);
    }
}
```

Come si fa in Python?

- Da stringa *s* a intero *k*: *k* = **int**(*s*)
- Da intero *k* a stringa *s*: *s* = **str**(*k*)

- Da stringa *s* a intero *k*: *k* = Integer.parseInt(*s*);
- Da intero *k* a stringa *s*: *s* = Integer.toString(*k*);
- *s* = ""+*k*;

Caratteri e stringhe

- Un **char** è un carattere alfanumerico o un simbolo
- Ci sono 2^{16} possibili valori di caratteri (più eventuali "caratteri supplementari", per esempio per il cinese)
 - Codifica Unicode basata su interi a 16 bit
- Racchiusi da apici (es. `'a'`, `'b'`, `'0'`, `'1'`, ecc.)
- Caratteri di escape:
 - **Tab**: `'\t'`
 - **A capo**: `'\n'`
 - **Backslash**: `'\\'`
 - **Apice**: `'\"'`
 - **Virgoletta**: `'\"'`
- Una **stringa** è una sequenza di caratteri

Booleani

- Il tipo **booleano** ha solo due valori possibili: **true** (vero) e **false** (falso)
- Gli operatori disponibili sono **and** (&&), **or** (||) e **not** (!)
- Le tabelle di verità sono:

a	!a
true	false
false	true

a	b	a && b	a b
false	false	false	false
false	true	false	true
true	false	false	true
true	true	true	true

Esempi di variabili booleane

- `boolean piove = true;`



SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

- `boolean giocoAPallone = !piove;`



- `boolean fuoriForma = true;`



- `boolean vadoInPalestra = piove && fuoriForma;`



- Tutti definiti su **ciascun tipo numerico primitivo**
- Producono un valore booleano
 - Uguaglianza: `==`
 - Diversità: `!=`
 - Minore: `<`
 - Minore uguale: `<=`
 - Maggiore: `>`
 - Maggiore uguale: `>=`

operatore	significato	true	false
==	uguale	42 == 42	42 == 43
!=	diverso	42 != 43	42 != 42
<	minore	42 < 100	42 < 42
<=	minore o uguale	42 <= 42	43 <= 42
>	maggiore	100 > 42	42 > 100
>=	maggiore o uguale	43 >= 42	42 >= 43

Esempi di espressioni

```
public class Espressioni
{
    public static void main(String[] args)
    {
        // esempi di espressioni
        int k = 10;
        int j = k+20;
        int h;

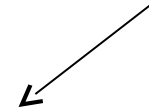
        System.out.print("k == ");
        System.out.println(k);
        System.out.print("j == ");
        System.out.println(j);
        System.out.print("h == ");
        System.out.println(h); // non definito: errore di compilazione!

        String s1 = "una stringa ";
        String s2 = "e un'altra stringa";
        String s3 = s1+s2;
        System.out.println(s3);

        double d = j+5.3;
        System.out.println("d == "+d);

        boolean b = (d == 35.3) && (j == 30);
        System.out.println("b == "+b);
    }
}
```

Questa riga va eliminata!



Output:

```
k == 10
j == 30
h == una stringa e un'altra stringa
d == 35.3
b == true
```

Errori tipici

```
public class AssegnazioniCheNonVanno
{
    public static void main(String[] args)
    {
        // ERRORE DI COMPILAZIONE: manca il tipo di k!!!
        k = 50;

        // ERRORE DI COMPILAZIONE: non posso assegnare una string a un intero!
        int k = "50";

        // ERRORE DI COMPILAZIONE: non posso assegnare un intero a una stringa!
        String s = 20;
    }
}
```

Esempio: anno bisestile



SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

```
public class AnnoBisestile
{
    public static void main(String[] args)
    {
        int anno = Integer.parseInt(args[0]);
        boolean bBisestile;
        bBisestile = anno % 4 == 0;
        bBisestile = bBisestile && (anno % 100 != 0);
        bBisestile = bBisestile || (anno % 400 == 0);

        System.out.println("L'anno "+anno+" e' bisestile? "+bBisestile);
    }
}
```

Ogni 4 anni

Ma non ogni 100

Tranne ogni 400!

- **Compila:** `javac AnnoBisestile.java`
- **Esegui:** `java AnnoBisestile 2012`
- **Output:** `L'anno 2012 e' bisestile? true`

Consapevolezza del tipo di dato

- **Regola aurea**: sii sempre consapevole del tipo di dati che il tuo programma sta elaborando
- Quali **valori** può **assumere** ciascuna variabile?
- Tuttavia, lavoriamo tipicamente con molteplici tipi di dati
- Come **convertire** un tipo di dati?



- **Conversione esplicita:**
 - Utilizzando un metodo che prende in ingresso un argomento di un tipo e restituisce un valore di un altro tipo
 - ❖ `Integer.parseInt()`, `Double.parseDouble()`, `Math.round()` ecc.
- **Cast esplicito:**
 - Antepoendo il tipo desiderato tra parentesi
 - ❖ `(int)2.71828` produce un intero di valore 2
 - Se il tipo di partenza è più preciso (es. `double`), le informazioni aggiuntive vengono eliminate nel modo più ragionevole (es. da `double` a `int` viene eliminata la parte frazionaria)
- **Cast implicito:**
 - Se il tipo di partenza è meno preciso, Java converte automaticamente il valore al tipo più preciso
 - ❖ `double d = 2;`

Conversioni di tipo: esempi

Il cast ha precedenza più elevata!

espressione	tipo	valore
(int)2.71828	int	2
Math.round(2.71828)	long	3
(int) Math.round(2.71828)	int	3
(int) Math.round(3.14159)	int	3
Integer.parseInt("42")	int	42
"42" + 99	String	"4299"
42 * 0.4	double	16.8
→ (int)42 * 0.4	double	16.8
42 * (int)0.4	int	0
(int)(42 * 0.4)	int	16

Esempio: un generatore di numeri

```
public class GeneraNumeri
{
    static public void main(String[] args)
    {
        // numero in input ← Commento su singola linea
        int max = Integer.parseInt(args[0]); ← Escluso 1

        // numero (pseudo)casuale compreso tra 0 e 1
        double r = Math.random();

        // intero (pseudo)casuale tra 0 e max-1
        int n = (int) (r*max);

        System.out.println(n);
    }
}
```

Esercizi (1)

1. Scrivere una classe **Moltiplica** che, dati in input 2 numeri interi, ne restituisca a video il prodotto
2. Scrivere una classe **StampaNome** che, dato in input un nome, lo stampi tra due righe di trattini. Ad es.:

```
+-----+
```

```
Roberto
```

```
+-----+
```

3. Scrivere una classe **Somma10** che calcoli la somma dei primi dieci interi positivi

Esercizi (2)

4. Scrivere una classe **Variabili** che, all'interno del metodo main, dichiari una variabile intera i, una variabile di tipo stringa s e una variabile double d. Quindi vengono svolte le seguenti tre operazioni:
- La stringa viene inizializzata al valore del primo argomento fornito in input al main
 - All'intero viene assegnato il valore intero della stringa
 - Al double viene assegnata la metà del valore di i (ad es. se i è pari a 3, d sarà pari a 1.5)
 - I valori di s, i e d vengono stampati a video