

Classi wrapper:

- Nella versione 1.5 di Java è stata introdotta una funzionalità davvero comoda che viene chiamata boxing (inscatolamento) che permette ai valori dei tipi primitivi di essere convertiti in oggetti, e viceversa. In particolare:
 - l'autoboxing è un casting automatico che permette ai valori dei tipi primitivi di essere convertiti in oggetti
 - l'unboxing effettua il casting inverso
 - Gli oggetti che rappresentano i tipi primitivi sono detti classi wrapper (in italiano "involucro") che sono classi che fanno da contenitore a un tipo di dato primitivo, astruendo proprio il concetto di tipo.
N.B. Tutte le classi wrapper sono classi final per cui non possono essere estese.
-

Classi "Wrapper" per tipi Primitivi

- Se si vogliono trattare anche i dati primitivi come oggetti si possono utilizzare le classi "wrapper"
- Un oggetto di una classe "wrapper" incorpora un dato primitivo e fornisce metodi per operare su di esso
- Le classi wrapper dei tipi primitivi sono definite nel package java.lang

Corrispondenza

Tipo Primitivo	ClasseWrapper
boolean	Boolean
char	Character
byte	Byte
short	Short
int	Integer
long	Long
float	Float
double	Double
void	Void

- In Java, ogni tipo primitivo possiede una corrispondente classe wrapper: Byte, Short, Integer, Long, Float, Double, Boolean, Character.
- Ciascuna di queste classi permette di manipolare i valori di tipo primitivo come se fossero valori di oggetti.
- Spesso abbiamo a che fare con tipi primitivi (int, double, boolean, ...) che sono tipi semplici e, quindi, non possiedono metodi.
- I wrapper, invece, essendo degli oggetti, sono dotati di metodi ed attributi.

- Prima dell'introduzione dell'autoboxing, programmando in Java ci si poteva trovare nella necessità di convertire un tipo primitivo nella sua corrispondente classe wrapper.

```
Integer x = new Integer(10);  
Double y = new Double(5.5f);  
Boolean z = Boolean.parseBoolean("true");
```

Occorreva prima creare un nuovo oggetto di una classe wrapper.

Le stesse operazioni precedenti possono essere ora eseguite mediante il seguente codice.

```
Integer x = 10;  
Double y = 5.5f;  
Boolean z = true;
```

L'autoboxing permette al developer di non preoccuparsi delle operazioni di conversione. Da notare che è grazie all'autoboxing che possiamo inserire i tipi primitivi all'interno delle Collection in maniera totalmente trasparente e senza preoccuparci di convertire i dati.

Classi wrapper	tipo primitivo
Boolean	è la classe wrapper per il tipo boolean.
Byte	è la classe wrapper per il tipo byte.
Character	è la classe wrapper per il tipo char.
Double	è la classe wrapper per il tipo double.
Float	è la classe wrapper per il tipo float.
Integer	è la classe wrapper per il tipo int.
Long	è la classe wrapper per il tipo long.
Short	è la classe wrapper per il tipo short.

Metodi statici della famiglia parse:

- `Byte.parseByte(String s)`
 - `Double.parseDouble(String s)`
 - `Float.parseFloat(String s)`
 - `Integer.parseInt(String s)`
 - `Long.parseLong(String s)`
 - `Short.parseShort(String s)`
-

Classe Character

La classe involucro Character si trova in java.lang

Include metodi statici per verificare le proprietà di un carattere

Esempio

```
String s = "Stringa Generica." char c = s.charAt (0); ... Character.isUpperCase(c) ... // true ...  
Character.isDigit(c) ... // false ...
```

Predicati vari

- boolean isLetter (char c)
- boolean isDigit (char c)
- boolean isLetterOrDigit (char c)
- boolean isSpaceChar (char c)
- boolean isLowerCase (char c)
- boolean isUpperCase (char c)

Trasformazioni varie

- char toUpperCase (char c)
- char toLowerCase (char c)