

Anno bisestile:

Un anno è bisestile se è divisibile per 400 oppure è divisibile per 4 ma non per 100.

Operatore per ottenere il resto di una divisione (%):

Esempio:

```
int dividendo = 10;  
int divisore = 3;  
int resto = dividendo % divisore;    (=1)
```

Conversione di una stringa al corrispondente valore intero che rappresenta:

Classe *java.lang.Integer* :

Metodo *parseInt*:

```
public static int parseInt(String n)
```

- parametro *n*: la stringa da decodificare (che rappresenta un numero intero);
- *valore di ritorno*: valore intero (tipo primitivo int) rappresentato dalla stringa *n*;

Metodo *decode*:

```
public static Integer decode(String n)
```

- parametro *n*: la stringa da decodificare (che rappresenta un numero intero);
- *valore di ritorno*: oggetto di classe *Integer* contenente il valore intero (tipo primitivo int) rappresentato dalla stringa *n*;

Metodo *intValue*:

```
public int intValue()
```

- *valore di ritorno*: il valore numerico contenuto in questo oggetto *Integer*.

Esempio:

Metodo più veloce:

```
String numString = "135";  
int num = Integer.parseInt(numString);
```

Alternative:

```
String numString = "135";
```

```
int num = Integer.decode(numString).intValue();
```

oppure, in forma “estesa”:

```
String numString = “135”;  
Integer numInteger = Integer.decode(numString);  
int num = numInteger.intValue();
```

Si noti l’uso dei metodi ‘static’(parseInt e decode): si invocano senza dover prima creare un’istanza della classe.

Metodi per la manipolazione di una stringa:

Classe *java.lang.String* :

Metodo *indexOf(int)*:

public int indexOf(int ch)

- parametro *ch*: il carattere che si intende cercare;
- *valore di ritorno*: l’indice della prima occorrenza del carattere nella stringa, oppure -1 se il carattere non esiste nella stringa;

Esempio:

```
String s = “ciao”;  
int i = s.indexOf(‘i’);    (=1)
```

Metodo *substring(int)*:

public String substring(int n)

- parametro *n*: indice di inizio, a partire dal quale si vuole ottenere la sottostringa;
- *valore di ritorno*: nuova stringa, sottoinsieme di quella originaria, che parte dal carattere di indice *n* e arriva fino alla fine;

Esempio:

```
String s1 = "unhappy";  
String s2 = s1.substring(2)           (="happy")
```

Metodo *substring(int , int)* :

public String substring(int n , int m)

- parametro *n*: indice di inizio;

- parametro *m*: indice di fine;
- *valore di ritorno*: nuova stringa, sottoinsieme di quella originaria, ottenuta prendendo in sequenza i caratteri dall'indice *n* all'indice *m-1* (compresi).

Esempio:

```
String s1 = "hamburger";  
String s2 = s1.substring (4,8);           (= "urge")
```

Per approfondimenti:

Documentazione ufficiale delle classi Java 1.5 :

consultazione on-line:

<http://java.sun.com/j2se/1.5.0/docs/api/>

download per consultazione in locale:

http://java.sun.com/javase/downloads/index_jdk5.jsp
(alla voce "J2SE 5.0 Documentation")