

a.s. 2012-13	ISTITUTO TECNICO INDUSTRIALE STATALE “E. Fermi” – Frascati (RM) Indirizzo Informatica “Abacus” Laboratorio di Informatica	Codice documento
		LABINF.01

Java Quick Reference

Parole chiave del linguaggio				
boolean	double	int	short	unsigned
break	else	long	signed	void
case	false	main	static	while
catch	final	new	switch	
char	for	public	throws	
class	if	register	true	
continue	import	return	try	
default				
do				

Commenti	
singola riga	// riga di commento
righe multiple	/* */

Dichiarazione	
variab ili	<i>tipo ident;</i>
costan	final <i>tipo</i>
<i>tipo: int, double, char, String, boolean</i>	

Commenti javadoc	Effetto	Tag javadoc	Scopo
/** commento */	monolinea	@author <i>nome</i> (*)	Nome autore
/** * commento */	multilinea	@param <i>nome descrizione</i>	Descrizione dei parametri
		@return <i>descrizione</i>	Descrizione tipo restituito
/** * * @tag commento tag */	Descrizione tag (elencati di seguito)	@throws <i>nomeclasse descrizione</i>	Descriz. Classe di eccezione
		@see <i>riferimento</i>	Aggiunta See also...
		@since <i>versione</i>	Versione in cui viene introd.

Tab. 1 – Commenti di Javadoc

Istruzioni semplici	
InputStreamReader In = new InputStreamReader (System.in);	Creazione oggetto <i>Tastiera</i>
BufferedReader Tastiera = new BufferedReader (In);	
<i>variabile</i> = Integer.parseInt (Tastiera.readLine());	Legge la <i>variabile</i> intera
<i>variabile</i> = Double.parseDouble (Tastiera.readLine());	Legge la <i>variabile</i> reale
<i>variabile</i> = Tastiera.readLine();	Legge la <i>variabile</i> stringa
System.out.println (<i>variabile</i>);	Stampa la <i>variabile</i>
<i>variabile</i> = <i>espressione</i>	Assegna <i>espressione</i> a <i>variabile</i>

Tab. 2 – Le istruzioni semplici

Istruzioni di I/O mediante finestre grafiche	
<i>variabile</i> = Integer.parseInt (JOptionPane.showInputDialog (<i>stringa</i>));	Legge la <i>variabile</i> intera
<i>variabile</i> = Double.parseDouble (JOptionPane.showInputDialog (<i>stringa</i>));	Legge la <i>variabile</i> reale
<i>variabile</i> = JOptionPane.showInputDialog (<i>stringa</i>);	Legge la <i>variabile</i> stringa
JOptionPane.showMessageDialog (null, <i>stringa</i>);	Stampa la <i>stringa</i>

Tab. 3 – Input/output mediante finestre grafiche

a.s. 2012-13	ISTITUTO TECNICO INDUSTRIALE STATALE “E. Fermi” – Frascati (RM) Indirizzo Informatica “Abacus” Laboratorio di Informatica	Codice documento
		LABINF.01

Tipi di dato	Tipo	Intervallo	Bit utilizzati
logico	boolean	false - true	8
carattere	unsigned char	da 0 a 255	8
	char	da -128 a 127	8
interi	unsigned short	Da 0 a 255	16
	short	da -32768 a 32767	16
	int	da -2.147.483.648 a 2.147.483.647	32
	unsigned long	da 0 a 4.294.967.295	32
	long	da -9223372036854775808 a +9223372036854775807	64
reali	float	da -3.4E ³⁸ a +3.4E ³⁸	32
	double	da -1.7E ³⁰⁸ a 1,79E ³⁰⁸	64

Tab. 4 – I principali tipi di dato

Operatori aritmetici	
+	Addizione
-	sottrazione
*	Moltiplicazione
/	Divisione
%	Modulo

Operatori relazionali	
<	Minore
>	Maggiore
<=	Minore o uguale
>=	Maggiore o uguale
==	Uguale
!=	Diverso

Operatori logici	
&&	AND
	OR
!	NOT

Operatori di assegnazione	
++	Autoincremento
--	Autodecremento
+=	Incremento
-=	Decremento
*=	Moltiplicazione
/=	Divisione
%=	Modulo

Sequenza di escape	
\n	A capo
\t	Tabulazione
\r	Inizio riga
\'	Virgolette
\\	Backslash
\b	Arretra un carattere
\'	Apice
\?	Punto interrogativo
\a	Beep
\0	Fine stringa
\f	Salto pagina

Funzioni classe String	
s1="le";	Assegnazione a s1 di una stringa costante
s3=s1+s2;	Concatena s1 con s2
s3 = s1.concat(s2);	Concatena s1 con s2
pos=s1.indexOf(c);	Posizione della prima occorrenza di c in s1
pos=s1.indexOf(s2);	Posizione della prima occorrenza di s2 in s1
pos = s1.lastIndexOf(c);	Posizione dell'ultima occorrenza di c in s1
pos = s1.lastIndexOf(s2);	Posizione dell'ultima occorrenza di s2 in s1
esito = s1.startsWith(s2);	Verifica se s2 è prefisso di s1
esito = s1.endsWith(s2);	Verifica se s2 è suffisso di s1
esito = s1.equals(s2);	Verifica l'uguaglianza di s1 ed s2
l = s1.compareTo(s2);	l = -1 se s1<s2; l = 0 se s1=s2; l = 1 se s1>s2
s2 = s1.toUpperCase();	Converte s1 in maiuscolo
s2 = s1.toLowerCase();	Converte s1 in minuscolo
l = s1.length();	Dà la lunghezza di s1
c = s1.charAt(2);	Pone in c il carattere in terza posizione di s1
s2=s1.substring(3,10);	Pone in s2 i caratteri di s1 dalla posizione 3 alla 10 esclusa

Tab. 5 – Le funzioni della classe String

a.s. 2012-13	ISTITUTO TECNICO INDUSTRIALE STATALE “E. Fermi” – Frascati (RM) Indirizzo Informatica “Abacus” Laboratorio di Informatica	Codice documento
		LABINF.01

Funzioni classe Math	
A= Math.abs (espressione);	Dà il valore assoluto di <i>espressione</i>
A= Math.pow (b, e);	Dà la potenza b ^e
A= Math.random ();	Dà un valore casuale tra 0 ed 1
A= Math.sqrt (espressione);	Dà la radice quadrata di <i>espressione</i>
A= Math.round (espressione);	Arrotonda <i>espressione</i> all'intero
System.out.println (Math.pi);	Stampa valore di <i>pi</i> greco
System.out.println (Math.e);	Stampa il numero “e” (base logaritmi naturali)

Tab. 6 – Le funzioni della classe Math

Funzioni di conversione	
int a= Integer.parseInt (stringa);	Converte <i>stringa</i> in intero
float a= Float.parseFloat (stringa);	Converte <i>stringa</i> in float
double a= Double.parseDouble (stringa);	Converte <i>stringa</i> in double
String s = String.valueOf (numero);	Converte <i>numero</i> in String

Tab. 7 – Le funzioni di conversione

Strutture di controllo	
Sequenza	{ <i>istruzione</i> ; }
Selezione semplice	if (<i>espressione logica</i>) <i>istruzione</i> ;
Selezione doppia	if (<i>espressione logica</i>) <i>istruzione</i> ; else <i>istruzione</i> ;
Selezione multipla	switch (<i>variabile</i>) { case <i>valore_1</i> : <i>istruzione</i> ; break ; ... case <i>valore_n</i> : <i>istruzione</i> ; break ; default : <i>istruzione</i> ; break ; }

Tab. 8 – Le strutture di controllo

Esecuzione da linea di comando
Compilazione Individua gli errori sintattici e crea il file <i>bytecode</i> (file <i>.class</i>): C:> javac <i>nomeprogramma.java</i>
Esecuzione Esegue il file <i>.class</i> prodotto dalla compilazione: C:> java <i>nomeprogramma</i>
Esecuzione con parametri Esegue il file <i>.class</i> e sostituisce i parametri <i>p1</i> e <i>p2</i> indicati nella linea di comando ad args[0] , args[1] : C:> java <i>nomeprogramma</i> <i>p1</i> <i>p2</i> (si sono indicati solo due parametri per semplicità).

Tab. 9 – I parametri args[]

Strutture di controllo	
Iterazione predefinita	for (<i>istruzione</i> ; <i>espressione logica</i> ; <i>istruzione</i>) <i>istruzione</i>
Iterazione precondizionata	while (<i>espressione logica</i>) <i>istruzione</i>
Iterazione postcondizionata	do { <i>istruzione</i> } while (<i>espressione logica</i>);

Tab. 10 – Le strutture di controllo iterative

a.s. 2012-13	ISTITUTO TECNICO INDUSTRIALE STATALE “E. Fermi” – Frascati (RM) Indirizzo Informatica “Abacus” Laboratorio di Informatica	Codice documento
		LABINF.01

Schema di base di un'applicazione Java

```

/* Titolo del programma */
/* File: ..... */
/* Data: ..... */
/* Autore: .... */

public class nomeclasse
{
    public static void main (String args[ ])
    {
        blocco istruzioni
    } //fine main
} // fine classe

```

Stream testo e binari

Stream testo standard	
InputStreamReader per lo <i>stream standard</i> input (tastiera)	
OutputStreamWriter per lo <i>stream standard</i> output (video)	
Metodi	
int read()	legge un carattere come intero dallo <i>stream standard</i> (-1 se il file è finito)
void write (char c)	scrive nello <i>stream standard</i> di output il carattere c
void close ()	chiude lo <i>stream</i>

Tab. 11 – I metodi degli stream standard

Stream testo su disco	
FileReader per <i>stream</i> di input su disco	
FileWriter per <i>stream</i> di output su disco	
Metodi	
int read()	legge un carattere come intero dallo <i>stream</i> di input (-1 se il file è finito)
void write (char c)	scrive nello <i>stream</i> di output il carattere c
void close ()	chiude lo <i>stream</i>

Tab. 12 – I metodi degli stream su disco

Stream testo (bufferizzazione)	
BufferedReader per <i>stream</i> di input bufferizzati	
BufferedWriter per <i>stream</i> di output bufferizzati	
Metodi	
String readLine()	legge una stringa dallo <i>stream</i> di input (null se il file è finito)
void write (String s)	scrive nello <i>stream</i> di output la stringa s
void newLine()	inserisce un “a capo” nello <i>stream</i> di output
void close ()	chiude lo <i>stream</i>

Tab. 13 – I metodi degli stream di testo bufferizzati

Stream binari	
FileInputStream per <i>stream</i> binari di input	
FileOutputStream per <i>stream</i> binari di output	
Metodi	
int read()	legge un byte dallo <i>stream</i> di input (-1 se il file è finito)
void write (int b)	scrive un byte nello <i>stream</i> di output
void flush()	svuota il buffer di output
void close ()	chiude lo <i>stream</i>

Tab. 14 – I metodi degli stream binari

a.s. 2012-13	ISTITUTO TECNICO INDUSTRIALE STATALE “E. Fermi” – Frascati (RM) Indirizzo Informatica “Abacus” Laboratorio di Informatica	Codice documento
		LABINF.01

Stream binari (bufferizzazione)	
BufferedInputStream per <i>stream</i> binari di input bufferizzati	
BufferedOutputStream per <i>stream</i> binari di output bufferizzati	
Metodi	
int read(byte b[])	legge un array di byte dallo <i>stream</i> di input (-1 se il file è finito)
void write (byte b[])	scrive un array di byte nello <i>stream</i> di output
void flush()	svuota il buffer di output
void close ()	chiude lo <i>stream</i>

Tab. 15 – I metodi degli stream binari bufferizzati

Stream binari di tipi primitivi	
DataInputStream per <i>stream</i> di input	
DataOutputStream per <i>stream</i> di output	
Metodi	
int readBoolean()	legge un booleano dallo <i>stream</i> di input (-1 se il file è finito)
int readChar()	legge un carattere dallo <i>stream</i> di input (-1 se il file è finito)
int readDouble()	legge un <i>double</i> dallo <i>stream</i> di input (-1 se il file è finito)
int readFloat()	legge un <i>float</i> dallo <i>stream</i> di input (-1 se il file è finito)
int readInt()	legge un intero dallo <i>stream</i> di input (-1 se il file è finito)
String readUTF()	legge una stringa in formato UTF (null se il file è finito)
void writeBoolean()	scrive un booleano nello <i>stream</i> di output
void writeChar()	scrive un carattere nello <i>stream</i> di output
void writeDouble()	scrive un <i>double</i> nello <i>stream</i> di output
void writeFloat()	scrive un <i>float</i> nello <i>stream</i> di output
void writeInt()	scrive un intero nello <i>stream</i> di output
void writeUTF(String s)	scrive una stringa nello <i>stream</i> di output
void write (byte b[])	scrive un array di byte nello <i>stream</i> di output
void flush()	svuota il buffer di output
void close ()	chiude lo <i>stream</i>

Tab. 16 – I metodi degli stream binari di dati primitivi

Stream binari di oggetti	
ObjectInputStream per <i>stream</i> binari di input bufferizzati	
ObjectOutputStream per <i>stream</i> binari di output bufferizzati	
Metodi	
Object readObject()	legge un oggetto dallo <i>stream</i> di input (null se il file è finito)
void writeObject (Object obj)	scrive un oggetto nello <i>stream</i> di output
void flush()	svuota il buffer di output
void close ()	chiude lo <i>stream</i>

Tab. 17 – I metodi degli stream binari di oggetti

a.s. 2012-13	ISTITUTO TECNICO INDUSTRIALE STATALE “E. Fermi” – Frascati (RM) Indirizzo Informatica “Abacus” Laboratorio di Informatica	Codice documento
		LABINF.01

File random	
METODO	EFFETTO
RandomAccessFile <i>ident</i> =new RandomAccessFile (String <i>file</i> , String <i>mode</i>)	Crea uno <i>stream</i> di nome <i>ident</i> , associato al file esterno <i>file</i> . <i>mode</i> =“r”, “rw”, “w”,...
void close()	Chiude lo <i>stream</i> ad accesso random
long getFilePointer()	Restituisce il file pointer nel file corrente
long length()	Restituisce la lunghezza del file
int read()	Legge un byte di dati dal file
int read(byte[] b)	Legge un array di byte dal file
boolean readBoolean()	Legge un valore booleano dal file
char readChar()	Legge un carattere UNICODE dal file
double readDouble()	Legge un valore doppio dal file
float readFloat()	Legge un valore reale dal file
int readInt()	Legge un intero di 32 bit dal file
String readLine()	Legge una stringa dal file fino a fine linea
String readUTF()	Legge una stringa dal file
void seek(long pos)	Posiziona il <i>file pointer</i> a <i>pos</i> byte rispetto all’inizio
int write(byte[] b)	Scrive un array di byte sul file
int write(int b)	Scrive un byte sul file
int writeBoolean(boolean v)	Scrive un valore booleano sul file
int writeBytes(String s)	Scrive una stringa s sul file
void writeChar(int v)	Scrive un carattere di 2 byte sul file
void writeInt(int v)	Scrive un intero di 32 bit sul file
void writeDouble(double v)	Scrive un valore doppio sul file
void writeFloat(float v)	Scrive un valore reale sul file
void writeUTF(String s)	Scrive una stringa sul file

Tab. 18 – I metodi dei file random

Strutture dati e oggetti

Vettori e matrici	
<i>tipo</i> <i>ident</i> = new <i>tipo</i> [<i>lunghezza</i>];	Dichiara e alloca il vettore <i>ident</i> di lunghezza <i>lunghezza</i>
<i>tipo</i> <i>ident</i> [] = { <i>listavalori</i> };	Dichiara e inizializza da programma il vettore <i>ident</i>
<i>tipo</i> <i>ident</i> [<i>righe</i>][<i>colonne</i>];	Dichiara e alloca una matrice <i>ident</i> di lunghezza <i>righe</i> * <i>colonne</i>
<i>tipo</i> <i>ident</i> [][<i>n</i>] = { { <i>lista riga 1</i> }, { <i>lista riga 2</i> }, ... };	Dichiara e inizializza da programma la matrice <i>ident</i> , ponendo su ogni riga la lista dei valori indicata, separati da virgole

Classe Object	
public boolean equals (Object o)	Metodo che stabilisce se due oggetti sono uguali
public int hashCode ()	Restituisce il codice dell’oggetto
public final Class getClass()	Restituisce il nome della classe
public String toString()	Dà la descrizione testuale della classe
protected Object clone()	Crea una copia dell’oggetto
protected void finalize()	Elimina l’oggetto dalla memoria

Tab. 19 – I metodi della classe Object

a.s. 2012-13	ISTITUTO TECNICO INDUSTRIALE STATALE “E. Fermi” – Frascati (RM) Indirizzo Informatica “Abacus” Laboratorio di Informatica	Codice documento
		LABINF.01

Interfacce grafiche

METODO	EFFETTO
Frame <i>ident</i> = new Frame ();	Crea la finestra <i>ident</i>
Frame <i>ident</i> = new Frame (String <i>titolo</i>);	Crea la finestra <i>ident</i> con <i>titolo</i>
f.setLocation (int <i>riga</i> , int <i>colonna</i>);	Posiziona la finestra in (<i>riga</i> , <i>colonna</i>)
f.setSize (int <i>larghezza</i> , int <i>altezza</i>);	Imposta <i>larghezza</i> ed <i>altezza</i> della finestra
f.setVisible (boolean <i>v</i>)	<i>v</i> = false (finestra invisibile), <i>v</i> = true (finestra visibile)
f.setResizable (boolean <i>r</i>)	<i>r</i> = false (non ridimensionabile), <i>r</i> = true (ridimensionabile)
f.setBounds (int <i>riga</i> , int <i>colonna</i> , int <i>larghezza</i> , int <i>altezza</i>)	Posiziona la finestra e imposta le dimensioni
close ()	Chiude la finestra

Tab. 20 – I principali metodi della classe Frame

METODO	EFFETTO
Panel <i>ident</i> = new Panel ();	Crea il pannello <i>ident</i>
setBackground (Color)	Imposta colore di sfondo

COSTANTE	COLORE	COSTANTE	COLORE
Color.black	Nero	Color.magenta	Violetto
Color.blue	Blu	Color.orange	Arancio
Color.cyan	Cyano	Color.pink	Rosa
Color.darkGreen	Verde scuro	Color.red	Rosso
Color.gray	Grigio	Color.white	Bianco
Color.green	Verde	Color.yellow	Giallo
Color.lightGray	Grigio chiaro		

Tab. 21 – Le costanti della classe Color

METODO	EFFETTO
Label <i>ident</i> = new Label ();	Crea l’etichetta <i>ident</i> vuota
Label <i>ident</i> = new Label (String <i>testo</i>);	Crea l’etichetta <i>ident</i> con <i>testo</i>
Label <i>ident</i> = new Label (String <i>testo</i> , int <i>all</i>);	Crea l’etichetta <i>ident</i> con <i>testo</i> allineato a destra (<i>all</i> = Label.RIGHT), centro (<i>all</i> = Label.CENTER) o sinistra (<i>all</i> = Label.LEFT) (oppure <i>all</i> =2,1,0)
l.setForeground (Color . <i>colore</i>);	Imposta il <i>colore</i> del testo
l.setBackground (Color . <i>colore</i>);	Imposta il <i>colore</i> dello sfondo
l.setText (String <i>stringa</i>);	Imposta il testo nell’etichetta
String l.getText ();	Restituisce il testo presente nella label
int l.getAlignment ();	Restituisce l’allineamento corrente
int l.setAlignment (int <i>all</i>);	Imposta allineamento dell’etichetta
setFont (Font <i>font</i>);	Imposta il <i>font</i> dei caratteri dell’etichetta

Tab. 22 – I principali metodi della classe Label

METODO	EFFETTO
Button <i>ident</i> = new Button ();	Crea un pulsante <i>ident</i> senza etichetta
Button <i>ident</i> = new Button (String <i>testo</i>);	Crea un pulsante <i>ident</i> con etichetta <i>testo</i>
b.setLabel (String <i>testo</i>);	Imposta l’etichetta del pulsante <i>ident</i>
String b.getLabel ();	Restituisce l’etichetta presente nel pulsante
b.setEnabled (boolean <i>e</i>);	Abilita (<i>e</i> = true) o disabilita (<i>e</i> = false) il pulsante

Tab. 23 – I metodi della classe Button

a.s. 2012-13	ISTITUTO TECNICO INDUSTRIALE STATALE “E. Fermi” – Frascati (RM) Indirizzo Informatica “Abacus” Laboratorio di Informatica	Codice documento
		LABINF.01

METODO	EFFETTO
TextField <i>ident</i> = new TextField ();	Crea una casella di testo <i>ident</i> vuota
TextField <i>ident</i> = new TextField (String <i>testo</i>);	Crea una casella <i>ident</i> con <i>testo</i>
TextField <i>ident</i> = new TextField (int <i>i</i>);	Crea una casella di testo <i>ident</i> vuota di lunghezza <i>l</i>
TextField <i>ident</i> = new TextField (String <i>testo</i> , int <i>i</i>);	Crea una casella di testo <i>ident</i> con <i>testo</i> di lunghezza <i>i</i>
String <i>t.getText</i> ();	Restituisce il testo presente nella casella di testo <i>t</i>
<i>t.setText</i> (String <i>testo</i>);	Imposta il <i>testo</i> nella casella di testo <i>t</i>
<i>t.setEditable</i> (boolean <i>e</i>)	Imposta la casella <i>t</i> come editabile (<i>e</i> = true) o non editabile (<i>e</i> = false)

Tab. 24 – I metodi della classe TextField

METODO	EFFETTO
Checkbox <i>ident</i> = new Checkbox ();	Crea una casella <i>ident</i> vuota
Checkbox <i>ident</i> = new Checkbox (String <i>testo</i>);	Crea una casella <i>ident</i> con <i>testo</i>
Checkbox <i>ident</i> = new Checkbox (String <i>testo</i> , boolean <i>s</i>);	Crea una casella <i>ident</i> con etichetta <i>testo</i> selezionata (<i>s</i> = true) o deselezionata (<i>s</i> = false)
String <i>c.getLabel</i> ();	Restituisce il testo presente nell'etichetta di <i>c</i>
<i>c.setLabel</i> (String <i>testo</i>);	Imposta il <i>testo</i> nella casella <i>c</i>
boolean <i>c.getState</i> ();	Restituisce lo stato (true o false) della casella <i>c</i>
<i>c.setState</i> (boolean <i>s</i>);	Imposta lo stato selezionato (<i>s</i> = true) o deselezionato (<i>s</i> = false) della casella <i>c</i>

Tab. 25 – I principali metodi della classe Checkbox

METODO	EFFETTO
CheckboxGroup <i>ident</i> =new CheckboxGroup ();	Crea una casella <i>ident</i> vuota
Checkbox <i>ident</i> = new Checkbox (String <i>testo</i>);	Crea una casella <i>ident</i> con <i>testo</i>
Checkbox <i>ident</i> = new Checkbox (String <i>testo</i> , boolean <i>s</i>);	Crea una casella <i>ident</i> con etichetta <i>testo</i> selezionata (<i>s</i> = true) o deselezionata (<i>s</i> = false)
Checkbox <i>ident</i> = new Checkbox (String <i>testo</i> , boolean <i>s</i> , CheckboxGroup <i>g</i>);	Crea una casella <i>ident</i> con etichetta <i>testo</i> selezionata (<i>s</i> = true) o deselezionata (<i>s</i> = false) facente parte del gruppo <i>g</i> di classe CheckboxGroup
String <i>c.getLabel</i> ();	Restituisce il testo presente nell'etichetta di <i>c</i>
<i>c.setLabel</i> (String <i>testo</i>);	Imposta il <i>testo</i> nella casella <i>c</i>
boolean <i>c.getState</i> ();	Restituisce lo stato (true o false) della casella <i>c</i>
<i>c.setState</i> (boolean <i>s</i>);	Imposta lo stato selezionato (<i>s</i> = true) o deselezionato (<i>s</i> = false) della casella <i>c</i>
<i>c.getSelectedCheckbox</i> ()	Dà la checkbox correntemente impostata per il gruppo <i>c</i>
<i>c.setSelectedCheckbox</i> (Checkbox <i>box</i>)	Imposta come selezionata la checkbox <i>box</i> del gruppo <i>c</i>

Tab. 26 – I principali metodi della classe CheckboxGroup

a.s. 2012-13	ISTITUTO TECNICO INDUSTRIALE STATALE “E. Fermi” – Frascati (RM) Indirizzo Informatica “Abacus” Laboratorio di Informatica	Codice documento
		LABINF.01

METODO	EFFETTO
Choice <i>ident</i> = new Choice ();	Crea una casella combinata <i>ident</i> vuota
c.addItem (String <i>testo</i>);	Aggiunge la voce <i>testo</i> alla casella <i>c</i>
String getItem (int <i>index</i>)	Restituisce la voce in posizione <i>index</i>
int getItemCount ()	Restituisce il numero di voci presenti
int getSelectedItem ()	Restituisce la posizione della voce corrente
String getSelectedItem ()	Restituisce la voce corrente
insert (String <i>s</i> , int <i>index</i>)	Inserisce la voce <i>s</i> in posizione <i>index</i>
remove (int <i>index</i>)	Rimuove la voce in posizione <i>index</i>
remove (String <i>s</i>)	Rimuove la prima occorrenza della voce <i>s</i>
removeAll ()	Rimuove tutte le voci dal menu
select (int <i>index</i>)	Seleziona la voce in posizione <i>index</i>
select (String <i>s</i>)	Seleziona la voce <i>s</i>

Tab. 27 – I principali metodi della classe Choice

METODO	EFFETTO
List <i>l</i> = new List ();	Crea una nuova lista <i>ident</i>
List <i>l</i> = new List (int <i>rows</i>);	Crea una lista <i>ident</i> con <i>rows</i> righe visibili
List <i>l</i> = new List (int <i>rows</i> , boolean <i>m</i>)	Crea una lista <i>ident</i> di <i>rows</i> righe in modo multiplo
void add (String <i>item</i>);	Aggiunge <i>item</i> in coda alla lista
void add (String <i>item</i> , int <i>index</i>);	Aggiunge <i>item</i> in posizione <i>index</i>
String getItem (int <i>index</i>);	Dà la stringa in posizione <i>index</i>
int getItemCount ();	Dà il numero di elementi nella lista
int getRows ();	Dà il numero di righe visibili della lista
int getSelectedIndex ();	Dà la posizione dell'elemento selezionato
String getSelectedItem ();	Dà l'elemento selezionato
boolean isIndexSelected ();	Vero se elemento <i>index</i> è selezionato
boolean isMultipleMode ()	Vero se la lista ammette selezioni multiple
void remove (int <i>index</i>)	Elimina elemento in posizione <i>index</i>
void removeAll ()	Elimina tutti gli elementi dalla lista
void replaceItem (String <i>item</i> int <i>index</i>)	Rimpiazza elemento <i>index</i> con <i>item</i>
void deselect (int <i>index</i>)	Deseleziona elemento <i>index</i>
void select (int <i>index</i>)	Seleziona elemento <i>index</i>
void setMultipleMode ()	Imposta il modo multiplo

Tab. 28 – I principali metodi della classe List

a.s. 2012-13	ISTITUTO TECNICO INDUSTRIALE STATALE “E. Fermi” – Frascati (RM) Indirizzo Informatica “Abacus” Laboratorio di Informatica	Codice documento
		LABINF.01

METODO	EFFETTO
Scrollbar <i>ident</i> = new Scrollbar ();	Crea una scrollbar <i>ident</i>
Scrollbar <i>ident</i> =new Scrollbar (int <i>direct</i>);	Crea una scrollbar <i>ident</i> orientata secondo <i>direct</i>
Scrollbar <i>ident</i> =new Scrollbar (int <i>direz</i> , int <i>value</i> , int <i>visible</i> , int <i>min</i> , int <i>max</i>);	Crea una scrollbar <i>ident</i> con le proprietà indicate
int getBlockIncrement ();	Dà l'incremento a blocchi
int getMaximum ();	Dà il valore massimo della barra
int getMinimum ();	Dà il valore minimo della barra
int getOrientation ();	Dà l'orientazione della barra
int getUnitIncrement ();	Dà l'incremento unitario
int getValue ();	Dà il valore corrente della barra
int getVisibleAmount ();	Dà la porzione visibile del cursore
int setBlockIncrement (int <i>b</i>);	Imposta incremento a blocchi
int setMaximum (int <i>max</i>);	Imposta il massimo valore
int setMinimum (int <i>min</i>);	Imposta il minimo valore
int setOrientation (int <i>direct</i>);	Imposta l'orientazione
int setUnitIncrement (int <i>u</i>);	Imposta incremento unitario
void setValue (int <i>value</i>);	Imposta il valore della barra
int setVisibleAmount (int <i>visible</i>);	Imposta la porzione visibile del cursore

Tab. 29 – I principali metodi della classe **Scrollbar**

METODO	EFFETTO
Menu ()	Crea un menu
Menu (String <i>l</i>)	Crea un menu con etichetta <i>l</i>
void add (String <i>label</i>)	Aggiunge <i>label</i> al menu
MenuItem add (MenuItem <i>mi</i>)	Aggiunge item con etichetta <i>mi</i> al menu corrente
MenuItem getItem (int <i>index</i>)	Dà l'oggetto MenuItem in posizione <i>index</i>
int getItemcount ()	Dà il numero di voci nel menu
void insert (MenuItem <i>mi</i> , int <i>index</i>)	Inserisce <i>mi</i> in posizione <i>index</i> del menu
void insert (String <i>label</i> , int <i>index</i>)	Inserisce nuova <i>label</i> in posizione <i>index</i>
void remove (int <i>index</i>)	Rimuove la voce in posizione <i>index</i>
void removeAll ()	Rimuove tutte le voci dal menu

Tab. 30 – I principali metodi della classe **Menu**

METODO	EFFETTO
FlowLayout <i>ident</i> = new FlowLayout ();	Crea un LM di tipo FlowLayout
FlowLayout <i>ident</i> = new FlowLayout (int <i>all</i>);	Crea un LM di tipo FlowLayout con allineamento <i>all</i> (FlowLayout.CENTER , FlowLayout.LEFT , FlowLayout.RIGHT)

Tab. 31 – I principali costruttori della classe **FlowLayout**

METODO	EFFETTO
BorderLayout <i>ident</i> = new BorderLayout ();	Crea un LM di tipo BorderLayout
BorderLayout <i>ident</i> = new BorderLayout (int <i>hgap</i> , int <i>vgap</i>);	Crea un LM di tipo BorderLayout con spaziatura orizzontale <i>hgap</i> e verticale <i>vgap</i> tra le componenti

Tab. 32 – I principali costruttori della classe **BorderLayout**

a.s. 2012-13	ISTITUTO TECNICO INDUSTRIALE STATALE “E. Fermi” – Frascati (RM) Indirizzo Informatica “Abacus” Laboratorio di Informatica	Codice documento
		LABINF.01

METODO	EFFETTO
GridLayout <i>ident</i> = new GridLayout ();	Crea un LM di tipo GridLayout con un componente per colonna
GridLayout <i>ident</i> = new GridLayout (int <i>r</i> , int <i>c</i>);	Crea un LM di tipo GridLayout con <i>r</i> righe e <i>c</i> colonne
GridLayout <i>ident</i> = new GridLayout (int <i>r</i> , int <i>c</i> , int <i>hgap</i> , int <i>vgap</i>);	Crea un LM di tipo BorderLayout con <i>r</i> righe e <i>c</i> colonne e spaziatura orizzontale <i>hgap</i> e verticale <i>vgap</i> tra le componenti
Tab. 33 – I principali costruttori della classe GridLayout	

a.s. 2012-13	ISTITUTO TECNICO INDUSTRIALE STATALE “E. Fermi” – Frascati (RM) Indirizzo Informatica “Abacus” Laboratorio di Informatica	Codice documento
		LABINF.01

Classe d’ascolto	Evento generato	Origine (<i>fire</i>)
FocusListener	Focus preso Focus perso	Tutti
MouseListener	Mouse entrato Mouse uscito Tasto mouse premuto Tasto mouse rilasciato Click	Tutti
MouseMotionListener	Mouse premuto e mosso Mouse mosso	Tutti
KeyListener	Tasto premuto Tasto rilasciato Tasto premuto e rilasciato	Tutti
WindowListener	Finestra attivata Finestra disattivata Finestra chiusa Chiudi finestra Riduci a icona Finestra aperta Finestra deiconificata	Window
WindowFocusListener	Fuoco preso Fuoco perso	Window
ActionListener	Azione su oggetto	Button TextField Menu
ItemListener	Click su un oggetto	Checkbox Choice CheckboxGroup Listbox
TextListener	Testo modificato	TextField TextArea
ContainerListener	Oggetto aggiunto Oggetto tolto	Tutti
ComponentListener	Oggetto nascosto Oggetto mostrato Oggetto ridimensionato Oggetto mosso	Tutti

Tab. 34 – Le classi d’ascolto Java

INTERFACCIA WindowListener	EVENTO
void windowActivated(WindowEvent e) {}	Finestra attivata
void windowDeactivated(WindowEvent e) {}	Finestra disattivata
void windowClosed(WindowEvent e) {}	Finestra chiusa
void windowIconified(WindowEvent e) {}	Riduci a icona
void windowOpened(WindowEvent e) {}	Finestra aperta
void windowDeiconified(WindowEvent e) {}	Finestra deiconificata
void windowClosing(WindowEvent e)	Chiudi finestra

Tab. 35 – I gestori di evento dell’interfaccia WindowListener

INTERFACCIA WindowFocusListener	EVENTO
void windowGainedFocus(WindowEvent e) {}	Fuoco preso
void windowLostFocus(WindowEvent e) {}	Fuoco perso

Tab. 36 – I gestori di evento dell’interfaccia WindowFocusListener

a.s. 2012-13	ISTITUTO TECNICO INDUSTRIALE STATALE “E. Fermi” – Frascati (RM) Indirizzo Informatica “Abacus” Laboratorio di Informatica	Codice documento
		LABINF.01

INTERFACCIA ActionListener	EVENTO
void actionPerformed (ActionEvent e) {}	Azione sul pulsante

Tab. 37 – I gestori di evento dell’interfaccia ActionListener

INTERFACCIA MouseListener	EVENTO
public void mouseClicked (MouseEvent e)	Click del mouse
public void mouseEntered (MouseEvent e)	Mouse entrato
public void mouseExited (MouseEvent e)	Mouse uscito
public void mousePressed (MouseEvent e)	Tasto mouse premuto
public void mouseReleased (MouseEvent e)	Tasto mouse rilasciato

Tab. 38 – I gestori di evento dell’interfaccia MouseListener

INTERFACCIA TextListener	EVENTO
Void textValueChanged (TextEvent e)	Testo modificato

Tab. 39 – Il gestore di evento dell’interfaccia TextListener

INTERFACCIA KeyListener	EVENTO
void keyPressed (KeyEvent e)	Testo premuto
void keyReleased (KeyEvent e)	Tasto rilasciato
void keyTyped (KeyEvent e)	Tasto premuto e rilasciato

Tab. 40 – Il gestore di evento dell’interfaccia KeyListener

INTERFACCIA ItemListener	EVENTO
void itemStateChanged (ItemEvent e)	Click su un oggetto

Tab. 41 – Il gestore di evento dell’interfaccia ItemListener

Animazione

Funzioni classe Timer	
METODO	EFFETTO
Timer (int delay, ActionListener listener)	Crea oggetto Timer che avvisa i suoi ascoltatori <i>listeners</i> ogni <i>delay</i> millisecondi
void start ()	Avvia il Timer , che inizia a mandare eventi ai suoi ascoltatori
void stop()	Arresta il Timer e smette di inviare eventi
void setDelay (int delay)	Imposta il <i>delay</i>
int getDelay ()	Restituisce il valore di <i>delay</i>

Tab. 42 – I principali metodi della classe Timer

a.s. 2012-13	ISTITUTO TECNICO INDUSTRIALE STATALE “E. Fermi” – Frascati (RM) Indirizzo Informatica “Abacus” Laboratorio di Informatica	Codice documento
		LABINF.01

Allocazione dinamica

Funzioni classe Vector	
METODO	EFFETTO
Vector <tipo> <i>ident</i> =new Vector <tipo>()	Crea un vettore vuoto di nome <i>ident</i>
Vector <tipo>=new Vector (int <i>n</i>)	Crea un vettore ideint vuoto di <i>n</i> elementi
Vector <tipo>=new Vector (int <i>n</i> , int <i>inc</i>)	Crea un vettore di <i>n</i> elementi ed aggiunge <i>inc</i> elementi ogni volta che viene riempito.
void addElement (Object <i>obj</i>)	Aggiunge l'elemento <i>obj</i> in coda e incrementa di uno la dimensione del vettore
void add (int <i>index</i> , Object <i>obj</i>)	Aggiunge l'elemento <i>obj</i> in posizione <i>index</i>
void clear ()	Rimuove tutti gli elementi dal vettore
boolean contains (Object <i>obj</i>)	Test di esistenza dell'elemento <i>obj</i> nel vettore
Object elementAt (int <i>index</i>)	Restituisce l'oggetto presente in posizione <i>index</i>
int indexOf (Object <i>obj</i>)	Restituisce la posizione dell'elemento <i>obj</i>
Object firstElement ()	Restituisce l'oggetto presente in prima posizione
Object lastElement ()	Restituisce l'oggetto presente in ultima posizione
boolean isEmpty ()	Test vettore vuoto
void insertelementAt (Object <i>obj</i> , int <i>index</i>)	Inserisce elemento <i>obj</i> in posizione <i>index</i>
void removeElementAt (int <i>index</i>)	Rimuove elemento in posizione <i>index</i>
Object set (int <i>index</i> , Object <i>obj</i>)	Sostituisce con <i>obj</i> l'elemento in posizione <i>index</i>
int size ()	Restituisce il numero di componenti presenti
String toString ()	Converte in stringa ciascun elemento del vettore

Tab. 43 – I principali metodi della classe Vector

Funzioni classe Arrays	
METODO	EFFETTO
static List asList (Object <i>v[]</i>)	Restituisce gli elementi dell'array <i>v[]</i> in una lista
static int binarySearch (tipo <i>t[]</i>], tipo <i>k</i>)	Cerca nell'array <i>t[]</i> il valore <i>k</i>
static boolean equals (tipo <i>t1[]</i>], tipo <i>t2[]</i>])	Testa se i due array sono uguali
static void fill (tipo <i>t[]</i>], tipo <i>v</i>)	Assegna a ciascuna componente di <i>t[]</i> il valore <i>v</i>
static void sort (tipo <i>t[]</i>])	Ordina l'array <i>t[]</i>]

Tab. 44 – I principali metodi della classe Arrays

Funzioni classe StringBuffer	
METODO	EFFETTO
StringBuffer ()	Crea uno stringbuffer vuoto
StringBuffer (int <i>length</i>)	Crea uno stringbuffer di lunghezza <i>length</i>
StringBuffer (String <i>str</i>)	Crea uno stringbuffer e lo inizializza con <i>str</i>
StringBuffer append (tipo <i>t</i>)	Converte <i>t</i> in stringa e l'aggiunge allo stringbuffer
int length ()	Dà il numero di caratteri dello stringbuffer
char charAt (int <i>index</i>)	Dà il carattere in posizione <i>index</i> dello stringbuffer
StringBuffer deleteCharAt (int <i>index</i>)	Elimina il carattere in posizione <i>index</i>
StringBuffer delete (int <i>start</i> , int <i>end</i>)	Elimina i caratteri da posizione <i>start</i> a <i>end</i>
int indexOf (String <i>str</i>)	Dà la posizione di <i>str</i> all'interno dello stringbuffer
StringBuffer insert (int <i>index</i> , tipo <i>t</i>)	Inserisce in posizione <i>index</i> di stringbuffer il dato <i>t</i>
StringBuffer replace (int <i>start</i> , int <i>end</i> , String <i>str</i>)	Sostituisce con <i>str</i> una sottostringa di stringbuffer
void setCharAt (int <i>index</i> , char <i>ch</i>)	Pone <i>ch</i> in posizione <i>index</i> dello stringbuffer
String substring (int <i>start</i>)	Dà la sottostringa di stringbuffer a partire da <i>start</i>
String substring (int <i>start</i> , int <i>end</i>)	Dà la sottostringa di stringbuffer tra <i>start</i> e <i>end</i>
String toString ()	Converte lo stringbuffer in String

Tab. 45 – I principali metodi della classe StringBuffer

a.s. 2012-13	ISTITUTO TECNICO INDUSTRIALE STATALE “E. Fermi” – Frascati (RM) Indirizzo Informatica “Abacus” Laboratorio di Informatica	Codice documento
		LABINF.01

Funzioni classe StringTokenizer	
METODO	EFFETTO
StringTokenizer (<i>String str</i>)	Crea <i>str</i> come stringtokenizer vuoto
StringTokenizer (<i>String str</i> , <i>String delim</i>)	Crea <i>str</i> stringtokenizer con delimitatore <i>delim</i>
int countTokens ()	numero di token ancora da analizzare
boolean hasMoreTokens ()	true se esistono ancora token da analizzare
String nextToken ()	restituisce il token successivo

Tab. 46 – I principali metodi della classe StringTokenizer

Funzioni classe LinkedList	
METODO	EFFETTO
LinkedList ()	Crea una lista vuota
void add (<i>Object elem</i>)	Inserisce <i>elem</i> in coda alla lista
void addLast (<i>Object elem</i>)	Inserisce <i>elem</i> in coda alla lista
void add (<i>int index</i> , <i>Object elem</i>)	Inserisce <i>elem</i> in posizione <i>index</i> nella lista
void addFirst (<i>Object elem</i>)	Inserisce <i>elem</i> in testa alla lista
void clear ()	Rimuove tutti gli elementi dalla lista
boolean contains (<i>Object elem</i>)	Dà true se <i>elem</i> esiste nella lista
Object element ()	Dà il primo elemento della lista senza rimuoverlo
Object get (<i>int index</i>)	Restituisce l'elemento in posizione <i>index</i>
Object getFirst ()	Restituisce l'elemento in prima posizione
Object getLast ()	Restituisce l'elemento in ultima posizione
int IndexOf (<i>Object elem</i>)	Dà la posizione della prima occorrenza di <i>elem</i>
Object remove ()	Dà il primo elemento della lista e lo rimuove
Object removeFirst ()	Dà il primo elemento della lista e lo rimuove
Object remove (<i>int index</i>)	Dà l'elemento in posizione <i>index</i> e lo rimuove
Object removeLast ()	Dà l'ultimo elemento della lista e lo rimuove
Object set (<i>int index</i> , <i>Object elem</i>)	Sostituisce con <i>elem</i> , l'elemento in posizione <i>index</i>
int size ()	Dà il numero di elementi presenti

Tab. 47 – I principali metodi della classe LinkedList

METODO	EFFETTO
Stack s = new Stack();	Costruttore
boolean empty ()	Testa se lo stack è vuoto
Object peek ()	Restituisce l'oggetto in cima senza estrarlo
Object pop ()	Restituisce l'oggetto in cima e lo estrae
Object push (<i>Object elemento</i>)	Inserisce l'oggetto <i>elemento</i> in cima allo stack
int search (<i>Object o</i>)	Restituisce la posizione di <i>o</i> nello stack
Object pop ()	Dà il primo elemento della lista e lo rimuove
int size ()	Restituisce il numero di elementi presenti

Tab. 48 – I metodi della classe Stack per la gestione di una pila

METODO	EFFETTO
Queue s = new Queue();	Costruttore
Object peek ()	Restituisce il prossimo da estrarre senza estrarlo
Object remove ()	Restituisce il prossimo da estrarre e lo estrae
Boolean add (<i>Object o</i>)	Inserisce <i>o</i> e dà errore se coda piena
int size ()	Restituisce il numero di elementi in coda

Tab. 49 – I metodi della classe Queue per la gestione di una coda

a.s. 2012-13	ISTITUTO TECNICO INDUSTRIALE STATALE “E. Fermi” – Frascati (RM) Indirizzo Informatica “Abacus” Laboratorio di Informatica	Codice documento
		LABINF.01

File e sistema operativo

Funzioni classe File	
METODO	EFFETTO
File f = new File (<i>percorso</i>);	Crea un file <i>f</i> in <i>percorso</i>
File f = new File (<i>percorso</i> , <i>nomefile</i>);	Crea in <i>percorso</i> , un file <i>f</i> avente nome <i>nomefile</i>
File f = new File (<i>directory</i> , <i>nomefile</i>);	Crea nella cartella <i>directory</i> un file <i>f</i> di nome <i>nomefile</i>
Boolean createNewFile ();	Crea un oggetto file nella cartella
public boolean exists ();	Verifica l'esistenza di un file
public boolean isFile ();	Verifica se l'oggetto è un file
public boolean isDirectory ();	Verifica se l'oggetto è una directory
public long length ();	Restituisce la lunghezza del file
public boolean renameTo (<i>dest</i>)	Cambia il nome del file.
public boolean delete ();	Cancella il file dal disco
public String getName ();	Restituisce il nome del file
public String getPath ();	Restituisce il <i>pathname</i> completo del file
String[] list ();	Dà un array di stringhe con i nomi dei file in directory
boolean mkdir ();	Crea una directory e restituisce l'esito

Tab. 50 – I principali metodi della classe File

Funzioni classe Runtime	
METODO	EFFETTO
Process static Runtime getRuntime ();	Dà l'oggetto Runtime associato all'applicazione
int availableProcessors ();	Dà il numero di processori disponibili per la JVM
long freeMemory ();	Dà la memoria disponibile per la JVM
long totalMemory ();	Dà la memoria totale disponibile per la JVM
public Process exec (<i>String command</i>);	Esegue il comando <i>command</i>
public Process exec (<i>String [] cmdArray</i>);	Esegue i comandi contenuti in <i>cmdArray</i>
public Process exec (<i>String command</i> , <i>String [] envp</i>);	Esegue <i>command</i> con gli argomenti nell'ambiente <i>envp</i> .
public Process exec (<i>String [] cmdArray</i> , <i>String [] envp</i> , <i>File dir</i>);	Esegue <i>command</i> con gli argomenti <i>cmdArray</i> nella <i>directory</i> indicata.

Tab. 51 – I principali metodi della classe Runtime

Funzioni classe Process	
METODO	EFFETTO
abstract int exitValue ();	Dà il valore di uscita del sottoprocesso
abstract InputStream getErrorStream ();	Dà l'error stream del processo
abstract InputStream getInputStream ();	Dà l'input stream del processo
abstract OutputStream getOutputStream	Dà l'output stream del processo
abstract int waitFor ();	Causa il blocco del thread corrente fino a che termini il processo rappresentato da questo oggetto
abstract void destroy ();	Uccide il sottoprocesso

Tab. 52 – I principali metodi della classe Process

a.s. 2012-13	ISTITUTO TECNICO INDUSTRIALE STATALE “E. Fermi” – Frascati (RM) Indirizzo Informatica “Abacus” Laboratorio di Informatica	Codice documento
		LABINF.01

Funzioni classe System			
METODO		EFFETTO	
String getProperty (String key)		Restituisce la proprietà richiesta del sistema	
key	Effetto	key	Effetto
"os.arch"	Architettura del S.O.	"file.separator"	Separatore “\” o “/”
"os.name"	Nome del S.O.	"java.class.path"	Posizione file di Java
"os.version"	Versione del S.O.	"java.home"	Posizione di JRE
"path.separator"	Separatore usato nel path	"java.vendor"	Nome fornitore di Java
"user.dir"	Directory di lavoro utente	"java.vendor.url"	URL fornitore di Java
"user.home"	Directory home utente	"java.version"	Num. versione di Java
"user.name"	Nome utente	"line.separator"	Separatore linee in text file
static long currentTimeMill()		Dà il tempo corrente in millisecondi dal 1/01/1970	
static void exit (int status)		Termina l’esecuzione della JVM e restituisce <i>status</i>	

Tab. 53 – I principali metodi della classe System

Thread

Funzioni classe Thread	
METODO	EFFETTO
Thread currentThread()	Restituisce il <i>thread</i> del processo corrente
void setName (String tname)	Assegna al <i>thread</i> il nome <i>tname</i>
String getName()	Restituisce il nome del <i>thread</i>
void setPriority (int priority)	Assegna al <i>thread</i> il valore di priorità <i>priority</i>
int getPriority()	Restituisce la priorità corrente
void sleep (int msec)	Pone in attesa il <i>thread</i> per <i>msec</i> millisecondi
void start()	Avvia il <i>thread</i>
void run()	Specifica l’elaborazione che deve eseguire il thread
void yield()	Cede il controllo ad un altro thread rimettendosi in coda
void join()	Attende la terminazione del <i>thread</i>

Tab. 54 – I principali metodi della classe Thread

Gestione database

Funzioni classe Connection	
METODO	EFFETTO
void close()	Rilascia la connessione con il DB e JDBC
Statement createStatement()	Crea un oggetto Statement per inviare un comando SQL al database
PreparedStatement prepareStatement (String sql)	Crea un oggetto PreparedStatement per mandare un comando <i>sql</i> parametrizzato al database

Tab. 55– I principali metodi della classe Connection

a.s. 2012-13	ISTITUTO TECNICO INDUSTRIALE STATALE “E. Fermi” – Frascati (RM) Indirizzo Informatica “Abacus” Laboratorio di Informatica	Codice documento
		LABINF.01

Funzioni classe DriverManager	
METODO	EFFETTO
static Connection getConnection (String url)	Tenta connessione con un database di <i>url</i> specificata
static Connection getConnection (String url, String user, String pass)	Tenta connessione con un database di <i>url</i> specificata, mediante username <i>user</i> e password <i>pass</i>

Tab. 56– I principali metodi della classe DriverManager

Funzioni classe Statement	
METODO	EFFETTO
void close()	Rilascia l’oggetto Statement utilizzato dal DB e da JDBC
int executeUpdate (String sql)	Esegue il comando <i>sql</i> che può contenere INSERT, UPDATE o DELETE o comandi DDL
ResultSet executeQuery (String sql)	Esegue il comando <i>sql</i> e restituisce nell’oggetto ResultSet la tabella trovata

Tab. 57– I principali metodi della classe Statement

Funzioni classe ResultSet	
METODO	EFFETTO
void close()	Rilascia l’oggetto ResultSet utilizzato dal DB e da JDBC.
String getString (int columnIndex)	Restituisce come stringa il valore presente nella colonna di posizione <i>columnIndex</i> del ResultSet corrente.
String getString (int columnLabel)	Restituisce come stringa il valore presente nella colonna <i>columnLabel</i> del ResultSet corrente.
int getInt (int columnIndex)	Restituisce come intero il valore presente nella colonna di posizione <i>columnIndex</i> del ResultSet corrente.
float getFloat (int columnIndex)	Restituisce come intero il valore presente nella colonna di posizione <i>columnIndex</i> del ResultSet corrente.
double getDouble (int columnIndex)	Restituisce come double il valore presente nella colonna di posizione <i>columnIndex</i> del ResultSet corrente.
boolean next()	Muove il cursore sulla riga successiva
boolean previous()	Muove il cursore sulla riga precedente
boolean first()	Muove il cursore sulla prima riga
boolean last()	Muove il cursore sull’ultima riga
boolean isLast()	Dà true se il cursore è sull’ultima riga
int getRow()	Dà il numero corrente di riga

Tab. 58– I principali metodi della classe ResultSet

a.s. 2012-13	ISTITUTO TECNICO INDUSTRIALE STATALE “E. Fermi” – Frascati (RM) Indirizzo Informatica “Abacus” Laboratorio di Informatica	Codice documento
		LABINF.01

Funzioni classe PreparedStatement	
METODO	EFFETTO
boolean execute()	Esegue un qualunque comando SQL
ResultSet executeQuery()	Esegue la query presente nell'oggetto ResultStatement e restituisce il ResultSet generato dalla query
int executeUpdate()	Esegue il comando SQL (DDL o DML) presente nell'oggetto PreparedStatement
void setDouble (int paramIndex, double x)	Pone il double <i>x</i> nel parametro indicato
void setFloat (int paramIndex, float x)	Pone il float <i>x</i> nel parametro indicato
void setInt (int paramIndex, int x)	Pone l'intero int <i>x</i> nel parametro indicato
void setString (int paramIndex, String x)	Pone la stringa <i>x</i> nel parametro indicato

Tab. 59– I principali metodi della classe PreparedStatement

a.s. 2012-13	ISTITUTO TECNICO INDUSTRIALE STATALE “E. Fermi” – Frascati (RM) Indirizzo Informatica “Abacus” Laboratorio di Informatica	Codice documento
		LABINF.01

Schemi di connessione a data base con comandi non parametrici

```

/* Esempio di connessione a database Access per comandi DDL e DML */
import java.sql.*;
public class nomeClasse
{
    public static void main(String args[])
    {
        String sql = stringa comando DDL o DML;
        try{ Class.forName("sun.jdbc.odbc.JdbcOdbcDriver"); }
        catch (ClassNotFoundException cnf)
        { System.out.println("Classe non trovata");
          System.out.println(cnf.getMessage()); }
        String url = "jdbc:odbc:nomeDriver";
        try{ Connection conn = DriverManager.getConnection(url, "", "");
          Statement stmt = conn.createStatement();
          stmt.executeUpdate(sql);
          stmt.close(); conn.close(); }
        catch(SQLException e) { System.out.println("Tabella esistente"); }
    }
}

```

```

/* Esempio di connessione a database Access per esecuzione query */
import java.sql.*;
public class nomeClasse
{
    public static void main(String args[])
    {
        dichiarazione variabili di output var1, var2, ..., varN;
        ResultSet rs;
        String query = stringa contenente la query;
        Statement stmt;
        try { Class.forName("sun.jdbc.odbc.JdbcOdbcDriver"); }
        catch (ClassNotFoundException cnf)
        { System.out.println("Classe non trovata");
          System.out.println(cnf.getMessage()); }
        String url = "jdbc:odbc:nomeDriver";
        try { Connection conn = DriverManager.getConnection(url, "", "");
          stmt = conn.createStatement();
          rs=stmt.executeQuery(query);
          while (rs.next())
          { var1 = rs.getXXX(posizione1);
            .....
            varN = rs.getXXX(posizioneN);
            Stampa tupla;
          }
          stmt.close();
          conn.close();
        }
        catch(SQLException e) { System.out.println("Inesistente"); }
    } // end main
} // end class

```

a.s. 2012-13	ISTITUTO TECNICO INDUSTRIALE STATALE “E. Fermi” – Frascati (RM) Indirizzo Informatica “Abacus” Laboratorio di Informatica	Codice documento
		LABINF.01

Schemi di connessione a data base con comandi parametrici

/ Esempio di connessione a database Access per comandi parametrici DDL e DML */*

```
import java.sql.*;
import java.io.*;
public class nomeClasse
{
    public static void main(String args[])
    {
        dichiarazione variabili di input;
        String sql = Stringa comando DDL o DML parametrico;
        PreparedStatement stmt;
        InputStreamReader In = new InputStreamReader (System.in);
        BufferedReader tastiera = new BufferedReader(In);
        try { Class.forName("sun.jdbc.odbc.JdbcOdbcDriver"); }
        catch (ClassNotFoundException cnf)
        {
            System.out.println("Classe non trovata");
            System.out.println(cnf.getMessage());
        }
        String url = "jdbc:odbc:nomeDriver";
        try { lettura variabili di input; }
        catch (Exception e) { System.out.println("Errore in lettura"); }
        try {
            Connection conn = DriverManager.getConnection(url, "", "");
            stmt = conn.prepareStatement(sql);
            caricamento dati in stmt mediante i metodi setXXX();
            stmt.executeUpdate();
            stmt.close();
            conn.close();
        }
        catch(SQLException e) { System.out.println("Esistente"); }
    }
}
```

/ Esempio di connessione a database Access per esecuzione query parametrica */*

```
import java.sql.*;
import java.io.*;
public class nomeClasse
{
    public static void main(String args[])
    {
        dichiarazione variabili di input;
        dichiarazione variabili di output var1, ..., varN;
        ResultSet rs;
        String query = stringa contenente la query parametrica;
        PreparedStatement stmt;
        InputStreamReader In = new InputStreamReader (System.in);
        BufferedReader tastiera = new BufferedReader(In);
        try { Class.forName("sun.jdbc.odbc.JdbcOdbcDriver"); }
        catch (ClassNotFoundException cnf)
        {
            System.out.println("Classe non trovata");
            System.out.println(cnf.getMessage());
        }
        String url = "jdbc:odbc:nomeDriver";
```

a.s. 2012-13	ISTITUTO TECNICO INDUSTRIALE STATALE “E. Fermi” – Frascati (RM) Indirizzo Informatica “Abacus” Laboratorio di Informatica	Codice documento
		LABINF.01

```

try { lettura variabili di input; }
catch (Exception e) { System.out.println("Errore in lettura"); }
try { Connection conn = DriverManager.getConnection(url, "", "");
      stmt = conn.prepareStatement(query);
      caricamento dati in stmt mediante i metodi setInt(), setString() ecc;
      rs=stmt.executeQuery();
      rs.next();
      varI = rs.getXXX(posizioneI);
      .....
      varN = rs.getXXX(posizioneN);
      stmt.close();
      conn.close();
    }
catch (Exception e) { Sstem.out.println("Inesistente"); }
}

```

Client – Server

Funzioni classe Socket	
METODO	EFFETTO
Socket()	Crea un socket non connessa
Socket (InetAddress address, int port)	Crea un socket e lo onnette alla porta <i>port</i> dell'indirizzo <i>address</i>
Socket (InetAddress address, int port, InetAddress localAddr, int localPort)	Crea un socket sulla macchina locale di indirizzo <i>localAddr</i> e porta <i>localPort</i> e loconnette alla porta <i>port</i> dell'indirizzo <i>address</i>
Socket (String host, int port)	Crea un socket e loconnette con l'host di indirizzo <i>host</i> , sulla porta <i>port</i> .
int getLocalPort()	Restituisce la porta a cui è connessa il socket
InetAddress getInetAddress()	Dà l'indirizzo a cui il socket è connesso
public Socket(String host, int port)	crea un <i>socket</i> di tipo <i>stream</i> e lo connette alla porta <i>port</i> dell'host <i>host</i>
public InputStream getInputStream()	ritorna uno <i>stream</i> di input per leggere dal socket
public OutputStream getOutputStream()	ritorna uno <i>stream</i> di output per scrivere sul <i>socket</i>
public synchronized void setSoTimeout(int timeout)	imposta un <i>timeout</i> per le operazioni di lettura dal <i>socket</i> (0 = attesa infinita)
public synchronized void close()	chiude il <i>socket</i>
void setReceiveBufferSize(int size)	Specifica dimensione buffer di ricezione
void setSendBufferSize(int size)	Specifica dimensione buffer di invio
Tab. 58– I principali metodi della classe Socket	

a.s. 2012-13	ISTITUTO TECNICO INDUSTRIALE STATALE “E. Fermi” – Frascati (RM) Indirizzo Informatica “Abacus” Laboratorio di Informatica	Codice documento
		LABINF.01

Funzioni classe ServerSocket	
METODO	EFFETTO
public ServerSocket(int <i>port</i>)	crea un <i>socket server</i> sulla porta <i>port</i> , che accetta fino a 50 richieste di servizio se <i>port</i> = 0, viene scelta una porta libera
public ServerSocket(int <i>port</i>, int <i>backlog</i>, InetAddress <i>bindAddr</i>)	crea un <i>socket server</i> sulla porta <i>port</i> , collegato all'indirizzo <i>bindAddr</i> ed accetta fino a <i>backlog</i> richieste di servizio
public Socket accept()	mette il <i>socket</i> in condizioni di ascolto, in attesa di richieste di connessione e restituisce un <i>socket</i> per colloquiare con il richiedente
public synchronized void close()	chiude il <i>socket</i>
Tab. 59– I principali metodi della classe ServerSocket	

a.s. 2012-13	ISTITUTO TECNICO INDUSTRIALE STATALE “E. Fermi” – Frascati (RM) Indirizzo Informatica “Abacus” Laboratorio di Informatica	Codice documento
		LABINF.01

0		32		64	@	96	`	128	Ç	160	á	192	⌂	224	Ó
1	☺	33	!	65	A	97	a	129	ü	161	í	193	⌋	225	ß
2	☹	34	"	66	B	98	b	130	é	162	ó	194	⌌	226	Ô
3	♥	35	#	67	C	99	c	131	â	163	ú	195	⌍	227	Ò
4	♦	36	\$	68	D	100	d	132	ä	164	ñ	196	⌎	228	õ
5	♣	37	%	69	E	101	e	133	à	165	Ñ	197	⌏	229	Õ
6	♠	38	&	70	F	102	f	134	á	166	ª	198	⌐	230	μ
7	•	39	'	71	G	103	g	135	ç	167	º	199	⌑	231	þ
8	◼	40	(72	H	104	h	136	ê	168	¿	200	⌒	232	ƒ
9	◊	41)	73	I	105	i	137	ë	169	®	201	⌓	233	Ú
10	◻	42	*	74	J	106	j	138	è	170	¬	202	⌔	234	Û
11	♂	43	+	75	K	107	k	139	ï	171	½	203	⌕	235	Ü
12	♀	44	,	76	L	108	l	140	î	172	¼	204	⌖	236	ý
13	🎵	45	-	77	M	109	m	141	ì	173	¡	205	⌗	237	Ý
14	🎶	46	.	78	N	110	n	142	Ä	174	«	206	⌘	238	—
15	☀	47	/	79	O	111	o	143	Å	175	»	207	◻	239	˙
16	▶	48	0	80	P	112	p	144	É	176	▤	208	◊	240	-
17	◀	49	1	81	Q	113	q	145	æ	177	▥	209	Ⓓ	241	±
18	↕	50	2	82	R	114	r	146	Æ	178	▦	210	Ê	242	—
19	!!	51	3	83	S	115	s	147	ø	179	▧	211	Ë	243	¾
20	¶	52	4	84	T	116	t	148	ö	180	▨	212	È	244	¶
21	§	53	5	85	U	117	u	149	ò	181	Á	213	Ì	245	§
22	—	54	6	86	V	118	v	150	û	182	Â	214	Í	246	÷
23	↕	55	7	87	W	119	w	151	ù	183	Ã	215	Î	247	,
24	↑	56	8	88	X	120	x	152	ÿ	184	©	216	Ï	248	°
25	↓	57	9	89	Y	121	y	153	Ö	185	▩	217	⌋	249	..
26	→	58	:	90	Z	122	z	154	Ü	186	▪	218	⌌	250	.
27	←	59	;	91	[123	{	155	ø	187	▫	219	▬	251	¹
28	⌂	60	<	92	\	124		156	£	188	▭	220	▮	252	³
29	↔	61	=	93]	125	}	157	Ø	189	¢	221	▯	253	²
30	▲	62	>	94	^	126	~	158	×	190	¥	222	▰	254	■
31	▼	63	?	95	_	127	△	159	f	191	⌏	223	▱	255	