

Tipi primitivi: int, char, float e double, boolean

boolean ≠ int

boolean b = true;

Conversioni: Java converte implicitamente valori di un tipo più piccolo in valori di un tipo più grande, NON viceversa

1) media di n numeri

```
import java.util.Scanner;
```

```
public class MediaNum {
```

```
    public static void main (String[] args) {
```

```
        System.out.println ("Quanti numeri?");
```

```
        Scanner sc = new Scanner (System.in);
```

```
        int nValori = sc.nextInt();
```

```
        do {
```

```
            System.out.println ("Errore!");
```

```
        } while (nValori < 1);
```

```
        int somma = 0;
```

```
        for (int i = 0; i < nValori; i++) {
```

```
            int value;
```

```
            System.out.println ("Inserire il valore");
```

```
            value = sc.nextInt();
```

```
            somma = somma + value;
```

```
        }
```

```
        float media = ((float) somma) / nValori,
```

```
System.out.println("la media dei numeri inseriti è: " + media);
```

```
}
```

```
}
```

funzioni: interne a classi, e sono metodi, NON è previsto overloading di operatori

→ passaggio di parametri

- primitivi: per valore

- reference (astrazione di un puntatore): per riferimento

Classi:

- la visibilità dei campi deve essere specificata per ogni singolo campo/metodo

```
public class C1 {
```

```
    private int x;
```

```
    public C1() {...};           //costruttore con zero parametri
```

```
    private void func1() {...}
```

```
    public int f2(int a) {...};
```

```
}
```

Oggetti: l'istanziamento di oggetti avviene sempre tramite l'operatore **new**, come per l'allocazione dinamica in Cpp.

C1 x; //dichiarazione di una variabile di tipo C1: crea una **reference** ad un oggetto di tipo C1 inizializzato a null

x = new C1(); //creazione dinamica dell'oggetto: crea un oggetto di tipo C1 e ritorna una **reference** all'oggetto appena creato

Java

```
C1 x;
```

```
x = new C1();
```

x è una **reference** ad
oggetti di tipo C1

CPP

```
C1* x;
```

```
x = new C1();
```

x è un **puntatore** ad
oggetti di tipo C1

puntatore (CPP): indirizzi di memoria

in Java non ci sono puntatori ma solo con il nome mi riferisco all'oggetto

accesso a campi e metodi

```
public class C1 {  
  
    private int x;  
  
    public C1 () {...};  
  
    private void func1 () {...}  
  
    public int f2 (int a) {...}  
  
}
```

C1 x = new C1();

x.f2(3);

x è una reference ad oggetti di tipo C1

a differenza di Cpp, x viene defenenziato automaticamente

2) Classe Data

```
import java.util.Scanner;
```

```
public class Data {
```

```
    private int giorno;
```

```
    private int mese;
```

```
    private int anno;
```

```
    public Data () {
```

//costruttore default con zero parametri

```
        giorno = 1;
```

```
        mese = 1;
```

```
        anno = 1970; }
```

```
    private boolean isValid(int giorno, int mese, int anno) {
```

```
        if (giorno < 1 || giorno > 31) {
```

```
            return false; }
```

```
        if (mese < 1 || mese > 12) {
```

```
            return false; }
```

```
        if (anno < 1) {
```

```
            return false; }
```

```
return true; }
```

```
public Data (int giorno, int mese , int anno) {
```

```
if ( isValid (giorno, mese, anno) {
```

```
this.giorno = giorno;
```

```
this.mese = mese ;
```

```
this.anno = anno;
```

```
return; }
```

```
{
```

```
System.err.println("Errore- data non valida: converisco la data al default");
```

```
this.giorno = 1;
```

```
this.mese = 1;
```

```
this.anno = 1970; }
```

```
public void print() {
```

```
System.out.print ( giorno + "/" + mese + "/" + anno);
```

```
}
```

```
public int getGiorno () {
```

```
return giorno; }
```

```
public void setGiorno (int giorno) {
```

```
if ( giorno < 1 || giorno > 31) {
```

```
System.err.println("Errore - utilizzo il default);
```

```
this.giorno = 1;
```

```
return; }
```

```
this.giorno = giorno; }
```

```
public int get Mese () {
```

```
return mese ; }
```

```
public void setMese (int mese ){
```

```
if (mese < 1 || mese > 12 ) {
```

```
System.err.println("Errore - utilizzo il default);
```

```
this.mese = 1;
```

```
return; }
```

```
this.mese = mese; }
```

```
public int getAnno (){
```

```
return anno ; }
```

```
public void setAnno (int anno){
```

```
if (anno < 1970){
```

```
System.err.println("Errore - utilizzo il default);
```

```
this.anno = 1970;
```

```
return; }
```

```
this.anno = anno ; }
```

```
public static void main (String[] args){
```

```
Data first = new Data(30, 3, 2022);
```

```
first.print;
```

```
Data second = new Data (29, 3, 2022);
```

```
second.print();
```

```
first = second; // dealloca first automaticamente
```

```
}
```