OOP

Clausole di visibilità

I metodi get e set

CC BY

PERCHÉ ABBIAMO INTRODOTTO IL COSTRUTTORE ?

1
Per non incorrere nell'errore:
«la var potrebbe non essere inizializzata»

```
Contatore c = new Contatore();
c.incrementa();
```

(il costruttore implicito)

2
Perché gli attributi dell'oggetto
abbiano **in ogni istante** valori significativi.

```
// dichiaro la data
Data d = new Data(19,4,1976);
```

(il costruttore parametrizzato)

CC BY

PERCHÉ ABBIAMO INTRODOTTO IL COSTRUTTORE ?

1
Per non incorrere nell'errore:
«la var potrebbe non essere inizializzata»

2
Perché gli attributi dell'oggetto
abbiano **in ogni istante** valori significativi.

In realtà,
con il costruttore ci assicuriamo che l'oggetto abbia un senso **solo appena istanziato** ma
nessuno ci garantisce che subito dopo non si diano **valori assurdi** agli attributi.

```
// dichiaro la data
Data d = new Data(12,12,1969);
d.giorno = 72;
d.mese = 24;
d.anno = -1;
```

Per far sì che
questo non avvenga
dovremmo impedire al nostro utente
(il programmatore)
di accedere direttamente agli attributi.

Ci serve un meccanismo per
obbligare il programmatore ad accedere agli attributi tramite l'invocazione di un metodo.
Ovviamente il codice del metodo farà da filtro per i valori inopportuni.

CC BY

CLAUSOLE DI VISIBILITÀ

Sono lo strumento con il quale specifichiamo cosa può usare il programmatore.

Se vogliamo concedere
al programmatore di accedere
direttamente ad un certo attributo
lo dichiareremo **public**.

Se **non** vogliamo concedere
al programmatore di accedere
direttamente ad un certo attributo
lo dichiareremo **private**.

```
public class Data {
    // attributi
    public int mese;

    // il resto della classe
}
```

```
Data d = new Data();
d.mese = 24;          // è lecito
```

```
public class Data {
    // attributi
    private int mese;

    // il resto della classe
}
```

```
Data d = new Data();
d.mese = 72;          // dà errore
```

Tipicamente gli attributi sono **sempre** privati.

CC BY

CLAUSOLE DI VISIBILITÀ

Sono lo strumento con il quale specifichiamo cosa può usare il programmatore.

Se vogliamo concedere al programmatore di accedere direttamente ad un certo attributo lo dichiareremo **public**.

```
public class Data {
    // attributi
    public int mese;

    // il resto della classe
}
```

```
Data d = new Data();
d.mese = 72;        // dà errore
```

Se **non** vogliamo concedere al programmatore di accedere direttamente ad un certo attributo lo dichiareremo **private**.

```
public class Data {
    // attributi
    private int mese;

    // il resto della classe
}
```

```
Data d = new Data();
d.mese = 72;        // dà errore
```

Tipicamente gli attributi sono **sempre** privati.

CC BY

COME ACCEDIAMO DUNQUE AGLI ATTRIBUTI ?

È prassi creare per ciascun attributo un metodo **set** (per accedere in scrittura) ed un metodo **get** (per accedere in lettura)

```
Data d = new Data();

d.setMese(72); // non dà errore
               //  ma non modifica
               //  niente

d.setMese(2);  // non dà errore
               //  e modifica
               //  correttamente
               //  il valore dell'
               //  attributo
```

```
public class Data {
    // attributi
    private int mese;

    // altri attributi e/o metodi

    // il metodo per scrivere
    void setMese(int x) {
        if (x>0 && x<13) {
            mese = x;
        }
    }

    // il metodo per leggere
    int getMese() {
        return mese;
    }

    // altri attributi e/o metodi
}
```

CC BY

IL TIPO DEI METODI SET

È prassi creare
per ciascun attributo
un metodo **set**
(per accedere in scrittura)
ed un metodo **get**
(per accedere in lettura)

Di norma,
il metodo **set** è sempre di tipo **void**

Noi troviamo più utile
restituire un **booleano**:
true → modifica effettuata
false → valore sbagliato

```
public class Data {
    // attributi
    private int mese;

    // altri attributi e/o metodi

    // il metodo per scrivere
    void setMese(int x) {
        if (x>0 && x<13) {
            mese = x;
        }
    }

    // il metodo per leggere
    int getMese() {
        return mese;
    }

    // altri attributi e/o metodi
}
```


CC BY

IL TIPO DEI METODI SET

È prassi creare
per ciascun attributo
un metodo **set**
(per accedere in scrittura)
ed un metodo **get**
(per accedere in lettura)

Di norma,
il metodo **set** è di tipo **void**

Certe volte è più utile
restituire un **booleano**:
true → modifica effettuata
false → valore NON ammesso

```
public class Data {
    // attributi
    private int mese;

    // altri attributi e/o metodi

    // il metodo per scrivere
    boolean setMese(int x) {
        if (x>0 && x<13) {
            mese = x;
            return true;
        } else {
            return false;
        }
    }

    // tutto il resto
```

CC BY

IL TIPO DEI METODI SET

```
Data d = new Data();

System.out.print("Inserisci il mese: ");
while(!d.setMese(Lettore.readInt())){
    System.out.print("Inserisci il mese: ");
}
```

```
public class Data {
    // attributi
    private int mese;

    // altri attributi e/o metodi

    // il metodo per scrivere
    boolean setMese(int x) {
        if (x>0 && x<13) {
            mese = x;
            return true;
        } else {
            return false;
        }
    }

    // tutto il resto
```

Certe volte è più utile
restituire un **booleano**:
true → modifica effettuata
false → valore NON ammesso

CC BY

CLAUSOLE DI VISIBILITÀ PER I METODI

Anche i metodi
devono essere dichiarati
pubblici o **privati**.

Tipicamente
i **metodi** sono **pubblici**
a meno che
non siano stati scritti
per esigenze di programmazione.

```
public class Data {
    // attributi
    private int giorno, mese, anno;
    // i costruttori
    public Data(int g, int m, int a){…}
    public Data() {…}
    // gli altri metodi
    public void setGiorno(int x) {…}
    public void setMese(int x) {…}
    public void setAnno(int x) {…}
    public int getGiorno() {…}
    public int getMese() {…}
    public int getAnno() {…}
    public String getDataComeStringa() {…}
    public void aggiungiUnGiorno() {…}
    private boolean annoBisestile() {…}
    private int ultimoGiorno() {…}
    public boolean isUguale(Data d) {…}
    public int differenzaInGiorni(Data d) {…}
    public boolean isMaggiore(Data d) {…}
}
```