



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Progettare un'interfaccia: costruttore di default e valori di default

Stefano Ghidoni



DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE



Agenda

- Progettare una buona interfaccia
 - Rivedendo l'interfaccia della classe date
- Costruttori e influenza sull'interfaccia



Una buona interfaccia

- Alcuni principi generali
 - L'interfaccia deve essere completa
 - L'interfaccia deve essere minimale
 - Devono essere forniti i costruttori
 - La copia deve essere supportata o proibita
 - Usare tipi adeguati per controllare gli argomenti
 - Identificare le funzioni membro costanti
 - Liberare tutte le risorse nel distruttore



Richiamo – Date

- Date definita in 05_3:

```
class Date {  
    int y, m, d;  
  
public:  
    Date (int y, int m, int d);  
    void add_day(int n);  
    int month() { return m; }  
    int day() { return d; }  
    int year() { return y; }  
};
```



Costruttore di default

- La classe Date ha un costruttore che accetta tre argomenti
 - Ciò rende illegale la definizione senza argomenti

```
Date d0;          // errore: nessuna inizializzazione
Date d1 {};       // errore: inizializzatore vuoto
Date d2 {1998};    // errore: argomenti insufficienti
Date d3 {1, 2, 3, 4}; // errore: troppi argomenti
Date d4 {1, "jan", 2}; // err: tipi sbagliati
Date d5 {1, Month::jan, 2}; // ok!
Date d6 {d5};      // ok: costruttore di copia
```



Se non c'è viene costruito automaticamente dal compilatore



Costruttore di default

- Spesso esistono dei valori di default significativi. Per esempio:
 - string: stringa vuota
 - vector: vettore vuoto
- In questi casi ha senso prevedere un costruttore di default
- Costruttori di default per tipi built-in: inizializzazione a 0
- **Forte influenza sull'interfaccia**



Costruttore di default

- Esempi di costruttori di default

```
// chiamate esplicite
string s1 = string{};
vector<string> v1 = vector<string> {};

// equivalenti, ma più compatte e semplici
string s1;
vector<string> v1;

// tipi built-in
int {};           // int inizializzato a 0
double {};        // double inizializzato a 0.0
```

→ Si e no: è corretto usare le parentesi graffe, che "provocano" una chiamata al costruttore, ma solitamente non viene chiamato un costruttore

I tipi built-in hanno un costruttore?

<https://stackoverflow.com/questions/5113365/do-built-in-types-have-default-constructors>

If it looks like a duck and quacks like a duck, then it is very duck-like



Valore di default nelle classi

- I valori di default dei membri possono essere inseriti:
 - Usando il costruttore di default
 - Nella definizione dei membri stessi
 - Valgono anche per gli altri costruttori



Valore di default nelle classi

```
class Date {  
public:  
    // ...  
    Date();  
    Date(int y, Month m, int d);  
    Date(int y);  
    // ...  
private:  
    int y {2001};  
    Month m {Month::jan};  
    int d {1};  
};
```

In-class
initializers

- `Date(int y)` sfrutta i valori di default di `m` e `d`



Argomenti di default

- Alternativa: prevedere argomenti di default nel costruttore
- Argomenti di default: valori forniti assieme ai parametri
 - Usati se in chiamata non è fornito il relativo argomento



Argomenti di default

```
class Date {  
public:  
    // ...  
    Date(int y = 2001, Month m = Month::jan, int d = 1);  
    // ...  
private:  
    int y;  
    Month m;  
    int d;  
};
```



Argomenti di default

- Un solo costruttore ora gestisce:
 - Costruttore di default
 - Costruttore con un solo argomento (y)
 - Costruttore con due argomenti (y e m)
 - Costruttore con tre argomenti: y, m, d

```
class Date {  
public:  
    // ...  
    Date(int y = 2001, Month m = Month::jan, int d = 1);  
    // ...  
private:  
    int y;  
    Month m;  
    int d;  
};
```



Argomenti di default

- Gli argomenti di default si possono usare per **qualsiasi funzione**
- Una volta fornito un argomento di default, tutti i successivi parametri devono averlo
 - Se fornisco un argomento di default per y, non posso non fornirlo per m e y

```
class Date {  
public:  
    // ...  
    Date(int y = 2001, Month m = Month::jan, int d = 1);  
    // ...  
private:  
    int y;  
    Month m;  
    int d;  
};
```



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Progettare un'interfaccia: costruttore di default e valori di default

Stefano Ghidoni