

PROGRAMMIAMO

C++ - Figure geometriche 1

[C++](#) | [Home](#) | [Contatti](#)

Un altro esempio di programma realizzato con le classi

Per approfondire la nostra conoscenza della programmazione ad oggetti, consideriamo ora un altro semplice programma. Lo scopo è quello di realizzare un programma per il calcolo dell'area di alcune figure geometriche elementari: rettangoli, triangoli, cerchi.

Se dovessimo scrivere questo programma secondo la metodologia tradizionale, probabilmente cominceremmo a progettare il main, quindi i dati (le variabili) che il main usa e infine le eventuali funzioni. Usando invece la programmazione ad oggetti, la progettazione invece comincia naturalmente dalla definizione degli oggetti, cioè delle classi che il programma dovrà usare:

```
class rettangoli {
private:
    double l1,l2;
public:
    void set()
    {
        cout<<"Primo lato: ";
        cin>>l1;
        cout<<"Secondo lato: ";
        cin>>l2;
    }
    double area()
    {
        return l1*l2;
    }
};
```

```
class triangoli {
private:
    double b,h;
public:
    void set()
    {
        cout<<"Base: ";
        cin>>b;
        cout<<"Altezza: ";
        cin>>h;
    }
    double area()
    {
        return 0.5*b*h;
    }
};
```

```

    }
};

class cerchi {
private:
    double r;
public:
    void set()
    {
        cout<<"Raggio: ";
        cin>>r;
    }
    double area()
    {
        return 3.14159*r*r;
    }
};

```

Osserviamo subito che in questo esempio i metodi *set* e *area* sono stati definiti all'interno della definizione di classe. Come sappiamo la possibilità alternativa è definire i metodi esternamente usando l'operatore [scope resolution](#).

E' interessante anche osservare che i metodi usati nelle tre classi hanno sempre gli stessi nomi:

- *set()*: serve per acquisire i dati;
- *area()*: serve per calcolare l'area.

Non solo i nomi sono stati scelti uguali, ma anche l'uso di tali metodi e lo scopo sono praticamente identici per tutte e tre le classi. Non si tratta di una scelta casuale od estetica, anzi: questa è una delle idee fondamentali della programmazione ad oggetti. Infatti è importante progettare oggetti simili in modo tale che sia possibile manipolarli con metodi simili. In questo modo il programmatore non deve imparare nuovi metodi diversi per ogni oggetto, ma può praticamente ripetere le stesse operazioni su tutti gli oggetti. In tale modo le differenze fra un oggetto e l'altro vengono nascoste all'interno degli oggetti stessi (per esempio la formula per il calcolo dell'area è diversa da una figura geometrica a un'altra) e non sono visibili dai programmatori.

Osserviamo a questo proposito la struttura estremamente semplice e lineare che assume il programma main:

```

int main(int argc, char *argv[])
{
    rettangoli ret;
    triangoli tri;
    cerchi cer;
    int scelta;
    double a;

    cout<<"Scegli la figura geometrica:\n";
    cout<<"1) rettangolo\n2) triangolo\n3)cerchio\n";
    cout<<"Scelta? ";
    cin>> scelta;

    switch (scelta)

```

```
{
    case 1:
        ret.set();
        a = ret.area();
        break;
    case 2:
        tri.set();
        a = tri.area();
        break;
    case 3:
        cer.set();
        a = cer.area();
        break;
}

cout<<"Area: "<<a<<"\n";

system("PAUSE");
return EXIT_SUCCESS;
}
```

Si noti come il main ripeta in effetti le stesse operazioni allo stesso modo sui tre differenti oggetti.

 [precedente](#) - [successiva](#) 

Sito realizzato in base al template offerto da

<http://www.graphixmania.it>

Segui @ElePrograMania