/ Design Patterns / Structural patterns / Adapter

Adapter in C++

Back to **Adapter** description

Adapter design pattern demo

Discussion. LegacyRectangle's interface is not compatible with the system that would like to reuse it. An abstract base class is created that specifies the desired interface. An "adapter" class is defined that publicly inherits the interface of the abstract class, and privately inherits the implementation of the legacy component. This adapter class "maps" or "impedance matches" the new interface to the old implementation.

```
#include <iostream.h>

typedef int Coordinate;
typedef int Dimension;

// Desired interface
class Rectangle
{
  public:
    virtual void draw() = 0;
};

// Legacy component
class LegacyRectangle
{
  public:
    LegacyRectangle(Coordinate x1, Coordinate y1, Coordinate x2, Coordinate y2)
    {
        x1_ = x1;
        y1_ = y1;
        x2_ = x2;
        y2_ = y2;
        cout << "LegacyRectangle:  create.  (" << x1_ << "," << y1_ << ") => ("
          << x2_ << "," << y2_ << ")" << endl;
    }
    void oldDraw()
    {
        cout << "LegacyRectangle:  oldDraw.  (" << x1_ << "," << y1_ <<
          ") => (" << x2_ << "," << y2_ << ")" << endl;
    }
  private:
    Coordinate x1_;
    Coordinate y1_;
    Coordinate x2_;
    Coordinate y2_;
};

// Adapter wrapper
class RectangleAdapter: public Rectangle, private LegacyRectangle
{
  public:
    RectangleAdapter(Coordinate x, Coordinate y, Dimension w, Dimension h):
      LegacyRectangle(x, y, x + w, y + h)
    {
        cout << "RectangleAdapter: create.  (" << x << "," << y <<
          "), width = " << w << ", height = " << h << endl;
    }
    virtual void draw()
    {
        cout << "RectangleAdapter: draw." << endl;
        oldDraw();
    }
};

int main()
{
  Rectangle *r = new RectangleAdapter(120, 200, 60, 40);
  r->draw();
}
```

Output

```
LegacyRectangle:  create.  (120,200) => (180,240)
RectangleAdapter: create.  (120,200), width = 60, height = 40
RectangleAdapter: draw.
LegacyRectangle:  oldDraw.  (120,200) => (180,240)
```

Read next

This article is taken from our book **Design Patterns Explained Simply**.

All of the design patterns are compiled there. The book is written in clear, simple language that makes it easy to read and understand (just like this article).

We distribute it in PDF & EPUB formats so you can get it onto your iPad, Kindle, or other portable device immediately after your purchase.

Learn more

Code examples

Java	**Adapter in Java: Before and after**	**Adapter in Java**
C++	**Adapter in C++**	**Adapter in C++: External Polymorphism**
PHP	**Adapter in PHP**	
Delphi	**Adapter in Delphi**	

Design Patterns
AntiPatterns
Refactoring
UML

My account
Forum
Contact us
About us

Terms / Privacy policy