

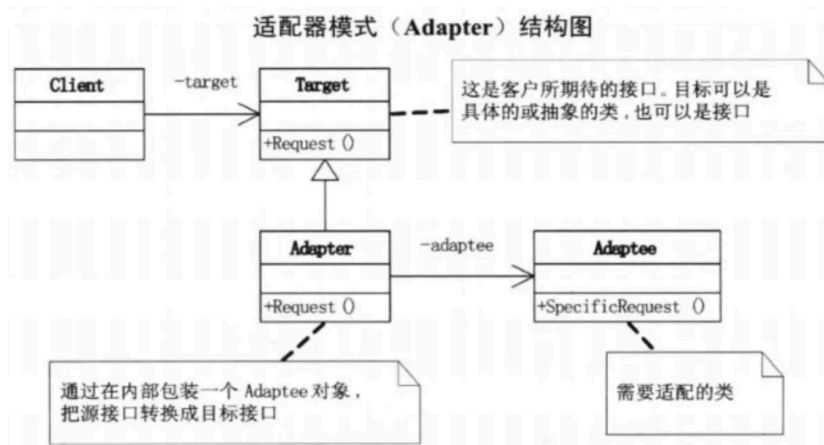
**动机:**在软件系统中,由于应用环境的变化,常常需要将"一些现存的对象"放在新的环境中应用,但是新环境要求的接口是这些现存对象所不满足的. 如何应对这种"迁移的变化"? 如何既能利用现有对象的良好实现,同时又能满足新的应用环境所要求的接口?

**定义:**将一个类的接口转换成客户希望的另一个接口,适配器模式使得原来由于接口不兼容而不能一起工作的那些类可以一起工作。  
<设计模式>GoF

**Target (目标抽象类) :** 目标抽象类定义客户所需接口, 可以是一个抽象类或接口, 也可以是具体类。

**Adapter (适配器类) :** 适配器可以调用另一个接口, 作为一个转换器, 对Adaptee和Target进行适配, 适配器类是适配器模式的核心, 在对象适配器中, 它通过继承Target并关联一个Adaptee对象使二者产生联系。

**Adaptee (适配者类) :** 适配者即被适配的角色, 它定义了一个已经存在的接口, 这个接口需要适配, 适配者类一般是一个具体类, 包含了客户希望使用的业务方法, 在某些情况下可能没有适配者类的源代码。



**要点总结:**该模式主要应用于"希望复用一些现存的类,但是接口又与复用环境要求不一致的情况",在遗留代码复用,类库迁移等方面非常有用。

```
1 #include <iostream>
2 using namespace std;
3 class Target{
4 public:
5     virtual ~Target(){}
6     virtual void Request() = 0;
7 };
8 class Adaptee{
9 public:
10     void SpecialRequest(){
11         cout << "Adaptee's SpecialRequest" << endl;
12     }
13 };
14 class Adapter : public Target{
15 private:
16     Adaptee * pAdaptee;
17 public:
18     Adapter() {
19         pAdaptee = new Adaptee();
20     }
21     ~Adapter(){
22         delete pAdaptee;
23     }
24     void Request(){
25         pAdaptee->SpecialRequest();
26     }
27 }
```

```

27 };
28 int main(){
29     Target * pA = new Adapter();
30     pA->Request();
31     delete pA;
32     return 0;
33 }

```

```

1 #include <iostream>
2 using namespace std;
3 class Target{
4 public:
5     virtual ~Target(){}
6     virtual void Request() = 0;
7 };
8 class Adaptee{
9 public:
10     void SpecialRequest(){
11         cout << "Adaptee's SpecialRequest" << endl;
12     }
13 };
14 class Adapter : public Target, Adaptee{
15 public:
16     void Request(){
17         Adaptee::SpecialRequest();
18     }
19 };
20 int main(){
21     Target* pA = new Adapter();
22     pA->Request();
23     delete pA;
24     return 0;
25 }

```

```

192:DesignPattnsStudy weishichun$ ls Adapter
Adapter_1.cpp      Adapter_2.cpp      Adapter适配器模式.pdf
192:DesignPattnsStudy weishichun$ g++ -o Adapter1.out Adapter_1.cpp
192:DesignPattnsStudy weishichun$ ./Adapter1.out
Adaptee's SpecialRequest
192:DesignPattnsStudy weishichun$ g++ -o Adapter2.out Adapter_2.cpp
192:DesignPattnsStudy weishichun$ ./Adapter2.out
Adaptee's SpecialRequest
192:DesignPattnsStudy weishichun$ █

```