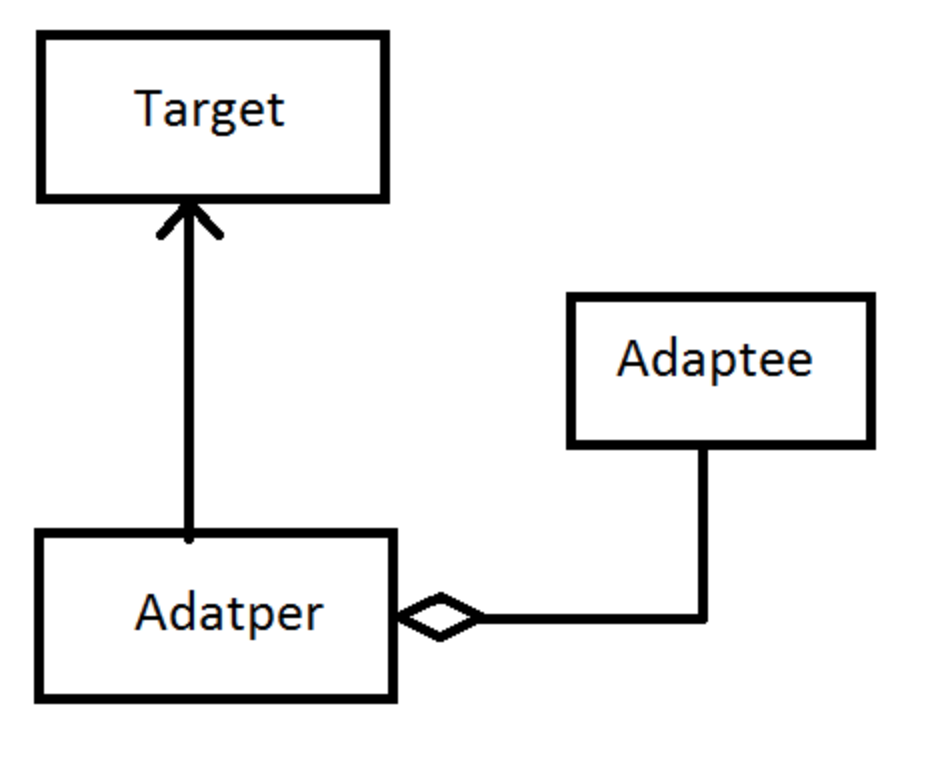
**Adapter**

Convert the interface of a class into another interface clients expect. Adapter lets classes work together that couldn't otherwise because of incompatible interfaces.

Adapter has an adaptee

Adapter is a Target



A switchboard can only use round pin sockets; however, we have a flat switch. What to do now?







*// Abstract Target*

class AbstractPlug {

public:

void virtual RoundPin(){}

void virtual PinCount(){}

};

*// Concrete Target*

class Plug : public AbstractPlug {

public:

void RoundPin() {

cout << " I am Round Pin" << endl;

}

void PinCount() {

cout << " I have two pins" << endl;

}

};

*// Abstract Adaptee*

class AbstractSwitchBoard {

public:

void virtual FlatPin() {}

void virtual PinCount() {}

};

*// Concrete Adaptee*

class SwitchBoard : public AbstractSwitchBoard {

public:

void FlatPin() {

cout << " Flat Pin" << endl;

}

void PinCount() {

cout << " I have three pins" << endl;

}

};

*// Adapter*

class Adapter : public AbstractPlug {

public:

AbstractSwitchBoard \*T;

Adapter(AbstractSwitchBoard \*TT) {

T = TT;

}

void RoundPin() {

T->FlatPin();

}

void PinCount() {

T->PinCount();

}

};

*// Client code*

int main()

{

SwitchBoard \*mySwitchBoard = new SwitchBoard; *// Adaptee*

*// Target = Adapter(Adaptee)*

AbstractPlug \*adapter = new Adapter(mySwitchBoard);

adapter->RoundPin();

adapter->PinCount();

return 0;

}