Facade in C++

[+] Feedback
BOOKMARK
Search ×

Why read if you can watch?
Watch design patterns video tutorial

Read full article

See the code

contents
Design Patterns

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

Facade design pattern demo

01 Discussion. Structuring a system into subsystems helps reduce complexity. A common design goal is to minimize the communication and dependencies between subsystems.

02 One way to achieve this goal is to introduce a “facade” object that provides a single, simplified interface to the many, potentially complex, individual interfaces within the subsystem. In this example, the “subsystem” for responding to a networking service

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

request has been modeled, and a facade (FacilitiesFacade) interposed. The facade “hides” the twisted and bizarre choreography necessary to satisfy even the most basic of requests. All the user of the facade object has to do is make one or two phone calls a week for 5 months, and a completed service request results.

03

```cpp
#include <iostream.h>

class MisDepartment
{
  public:
    void submitNetworkRequest()
    {
        _state = 0;
    }
    bool checkOnStatus()
    {
        _state++;
        if (_state == Complete)
          return 1;
        return 0;
    }
  private:
    enum States
    {
        Received, DenyAllKnowledge, ReferClientToFacilities,
          FacilitiesHasNotSentPaperwork, ElectricianIsNotDone,
          ElectricianDidItWrong, DispatchTechnician, SignedOff, DoesNotWork,
          FixElectriciansWiring, Complete
    };
    int _state;
};

class ElectricianUnion
{
```

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

```
  public:
    void submitNetworkRequest()
    {
        _state = 0;
    }
    bool checkOnStatus()
    {
        _state++;
        if (_state == Complete)
          return 1;
        return 0;
    }
  private:
    enum States
    {
        Received, RejectTheForm, SizeTheJob, SmokeAndJokeBreak,
          WaitForAuthorization, DoTheWrongJob, BlameTheEngineer, WaitToPunchOut,
          DoHalfAJob, ComplainToEngineer, GetClarification, CompleteTheJob,
          TurnInThePaperwork, Complete
    };
    int _state;
};

class FacilitiesDepartment
{
  public:
    void submitNetworkRequest()
    {
        _state = 0;
    }
    bool checkOnStatus()
    {
        _state++;
        if (_state == Complete)
          return 1;
        return 0;
```

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

```
    }
  private:
    enum States
    {
        Received, AssignToEngineer, EngineerResearches, RequestIsNotPossible,
          EngineerLeavesCompany, AssignToNewEngineer, NewEngineerResearches,
          ReassignEngineer, EngineerReturns, EngineerResearchesAgain,
          EngineerFillsOutPaperWork, Complete
    };
    int _state;
};

class FacilitiesFacade
{
  public:
    FacilitiesFacade()
    {
        _count = 0;
    }
    void submitNetworkRequest()
    {
        _state = 0;
    }
    bool checkOnStatus()
    {
        _count++;
        /* Job request has just been received */
        if (_state == Received)
        {
            _state++;
            /* Forward the job request to the engineer */
            _engineer.submitNetworkRequest();
            cout << "submitted to Facilities - " << _count <<
              " phone calls so far" << endl;
        }
        else if (_state == SubmitToEngineer)
```

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

```
        {
            /* If engineer is complete, forward to electrician */
            if (_engineer.checkOnStatus())
            {
                _state++;
                _electrician.submitNetworkRequest();
                cout << "submitted to Electrician - " << _count <<
                  " phone calls so far" << endl;
            }
        }
        else if (_state == SubmitToElectrician)
        {
            /* If electrician is complete, forward to technician */
            if (_electrician.checkOnStatus())
            {
                _state++;
                _technician.submitNetworkRequest();
                cout << "submitted to MIS - " << _count <<
                  " phone calls so far" << endl;
            }
        }
        else if (_state == SubmitToTechnician)
        {
            /* If technician is complete, job is done */
            if (_technician.checkOnStatus())
              return 1;
        }
        /* The job is not entirely complete */
        return 0;
    }
    int getNumberOfCalls()

    {
        return _count;
    }
  private:
```

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

```
    enum States
    {
        Received, SubmitToEngineer, SubmitToElectrician, SubmitToTechnician
    };
    int _state;
    int _count;
    FacilitiesDepartment _engineer;
    ElectricianUnion _electrician;
    MisDepartment _technician;
};

int main()
{
  FacilitiesFacade facilities;

  facilities.submitNetworkRequest();
  /* Keep checking until job is complete */
  while (!facilities.checkOnStatus())
    ;
  cout << "job completed after only " << facilities.getNumberOfCalls() <<
    " phone calls" << endl;
}
```

Output

```
submitted to Facilities - 1 phone calls so far
submitted to Electrician - 12 phone calls so far
submitted to MIS - 25 phone calls so far
job completed after only 35 phone calls
```

List of Facade examples

C# examples

- Facade in C#

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

C++ examples

- Facade in C++ <=[You are here]

Delphi examples

- Facade in Delphi

Java examples

- Facade in Java

PHP examples

- Facade in PHP

‹ Decorator Design Pattern

↑ Facade

Flyweight Design Pattern ›

This work is licensed under a Creative Commons Attribution-Noncommercial-No Derivative Works 3.0 Unported License

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com