Command Pattern Code

Jim Fawcett

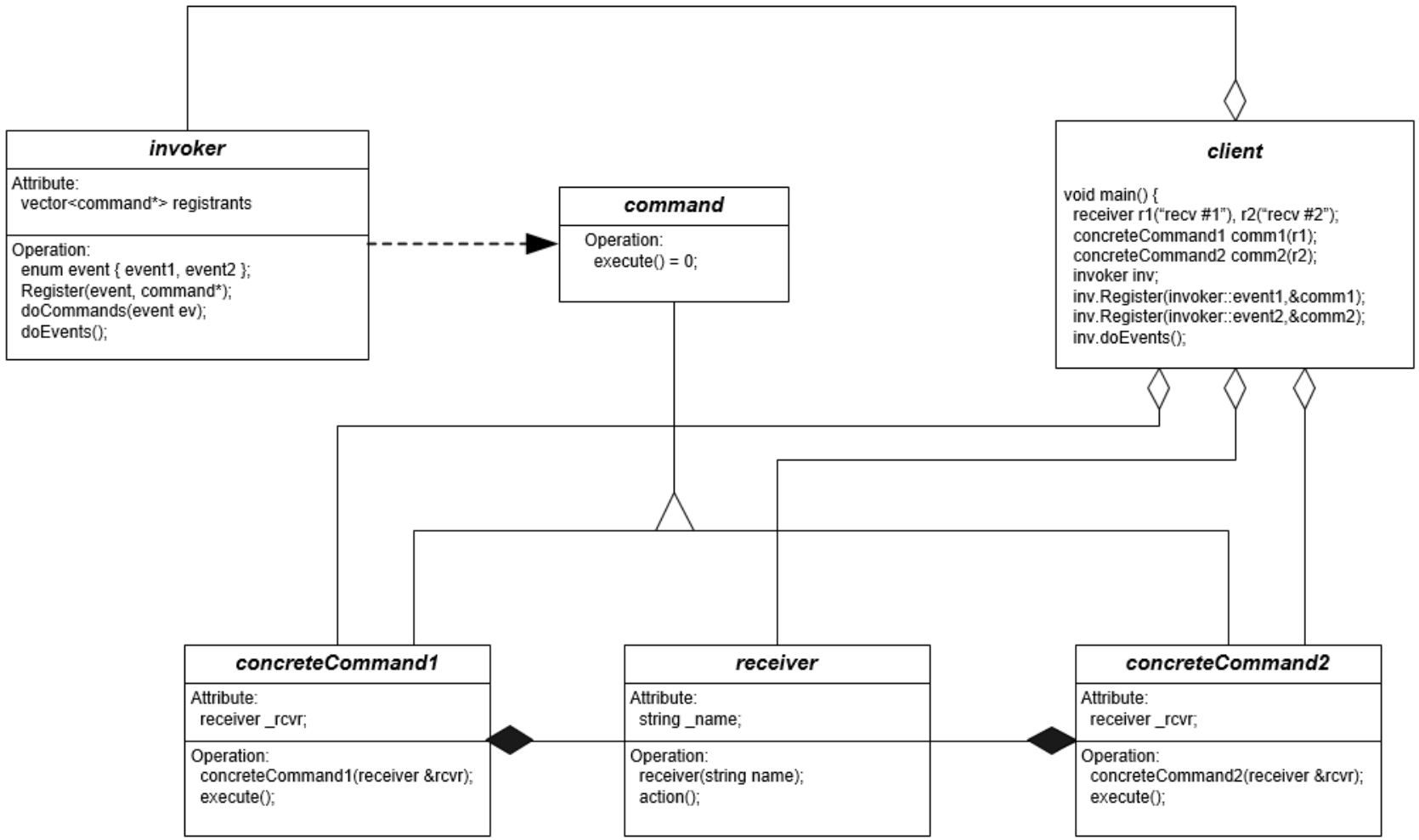
CSE776 Design Patterns

Fall 2018

Command Pattern Skeleton Code

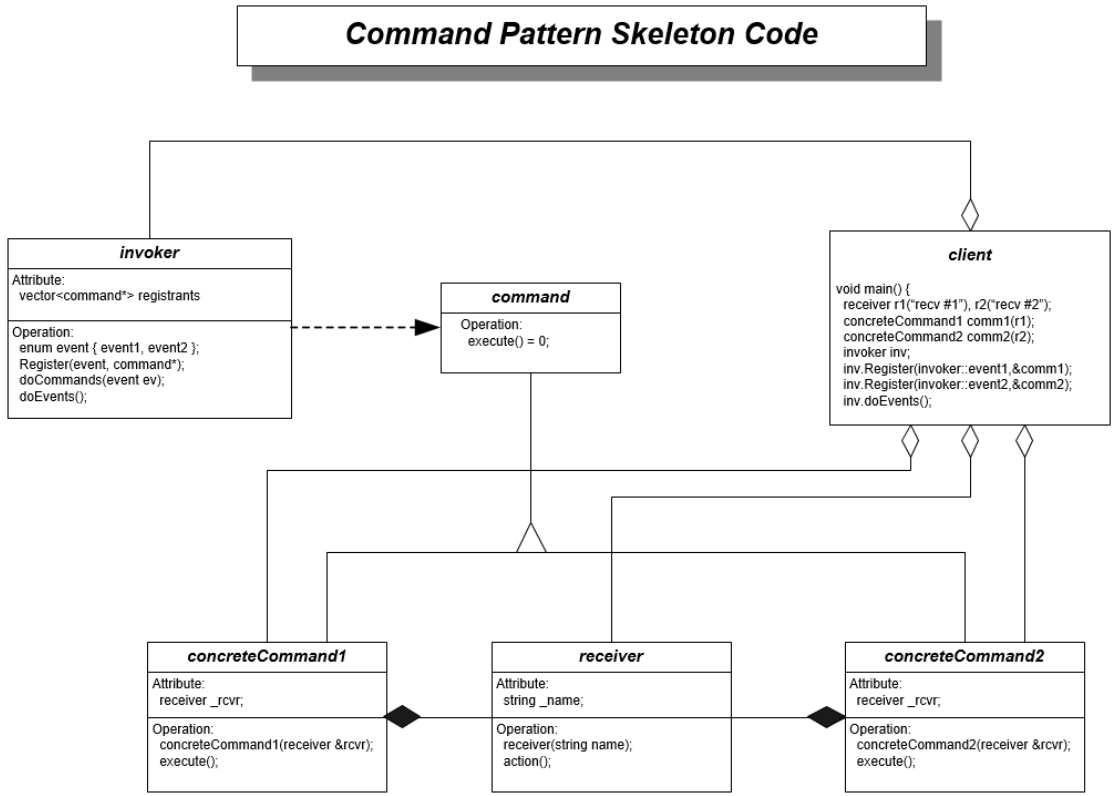
- The simplest code that implements the pattern.
- Uses class names from the “Design Patterns” book.

Command Pattern Skeleton Code

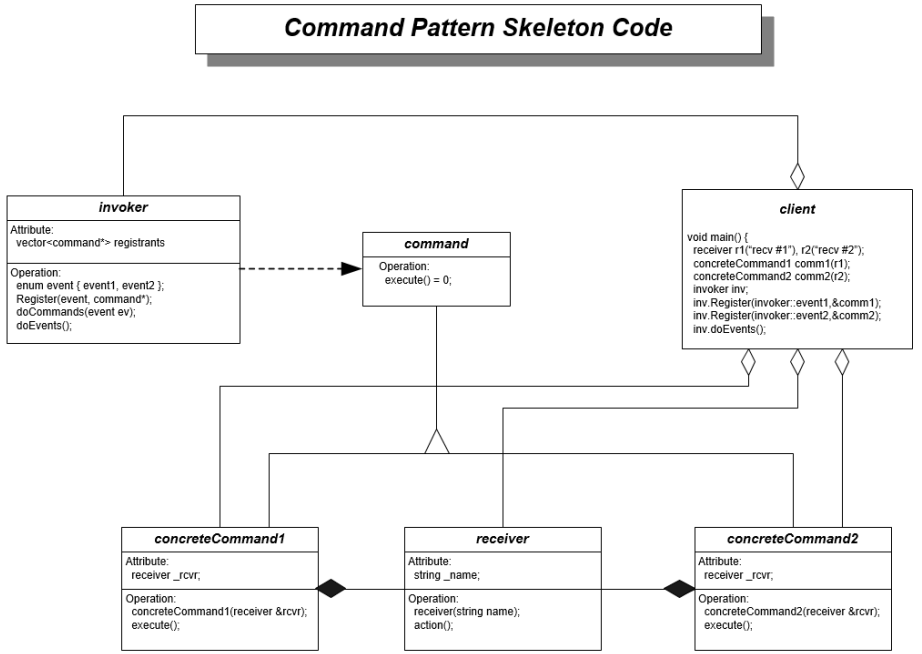


Library Code

```
class invoker {

public:
  enum event { event1, event2, event3 };
  void Register(event,command*);
  void doCommands(event ev);
  void doEvents();

private:
  std::vector<command*> registrants[3];
};


class command {

public:
  virtual void execute() = 0;
};
```



Application Code



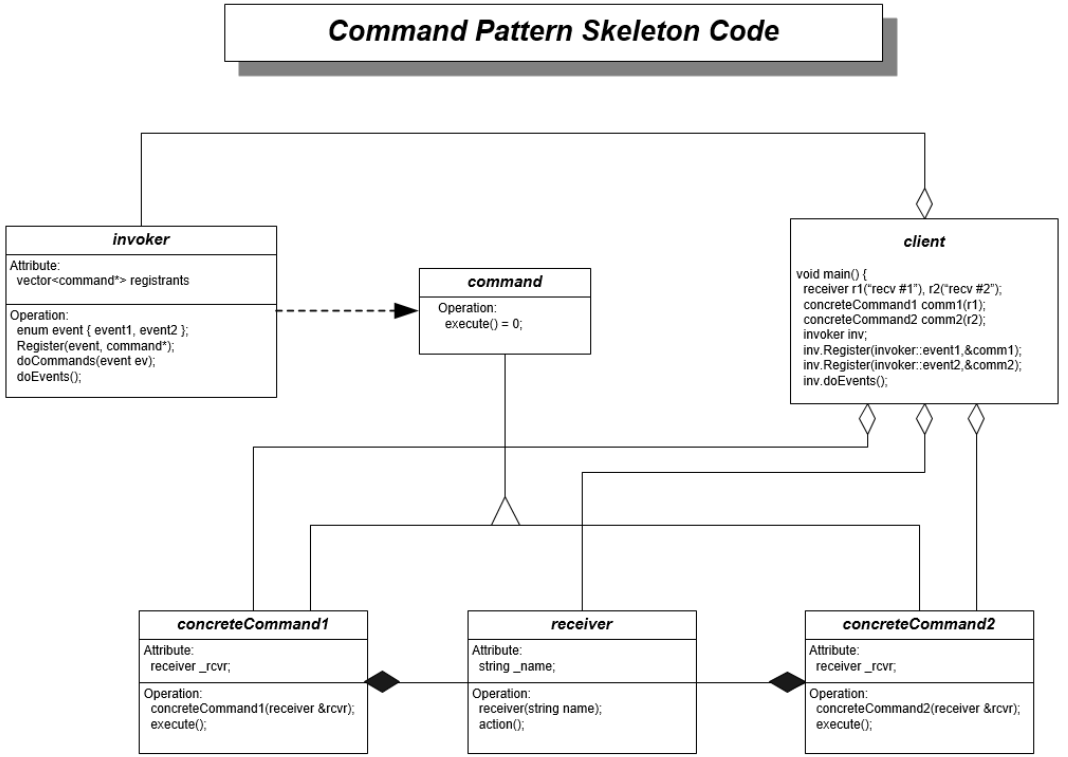

```
class receiver {

public:
  receiver(const std::string &name) : _name(name) { }
  void action();

private:
  std::string _name;
};
```

```
class concreteCommand1 : public command {

public:
  concreteCommand1(receiver &rcvr)
    : _rcvr(rcvr) { }
  void execute();

private:
  receiver _rcvr;
};
```

```
class concreteCommand2 : public command {

public:
  concreteCommand2(receiver &rcvr)
    : _rcvr(rcvr) { }
  void execute();

private:
  receiver _rcvr;
};
```

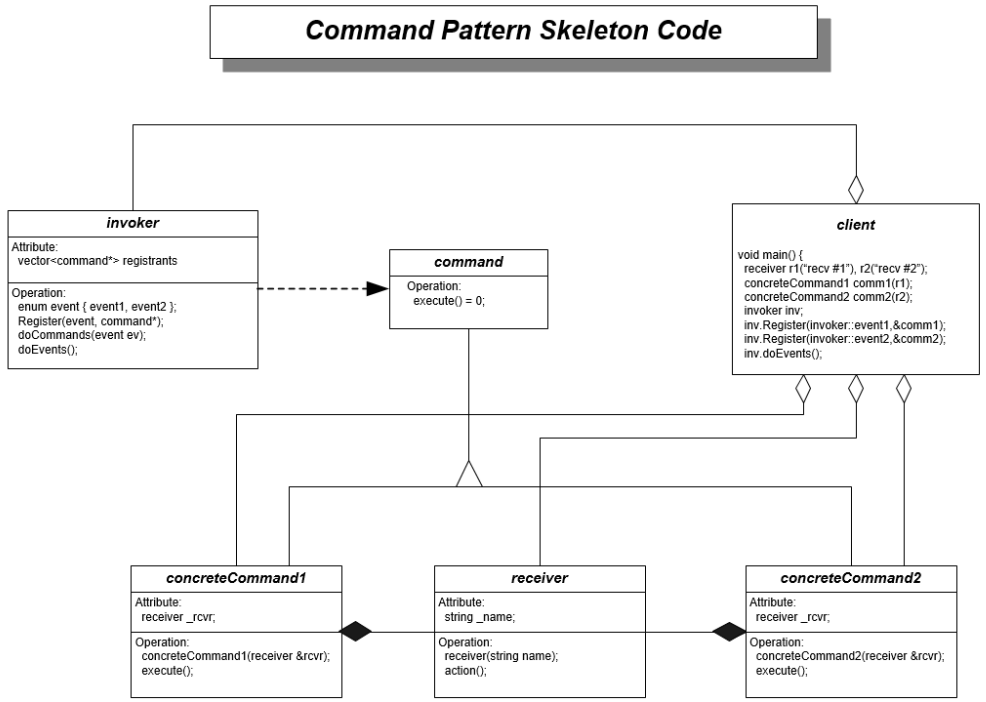
Event Processing

```
void invoker::Register(event ev, command* pComm) {

  registrants[ev].push_back(pComm);
}


void invoker::doCommands(event ev) {

  for(int i=0; i<(int)registrants[ev].size(); i++)
    registrants[ev][i]->execute();
}


void invoker::doEvents() {

  cout << "\n\n  invoker processing event1";
  doCommands(event1);

  cout << "\n\n  invoker processing event2";
  doCommands(event2);

  cout << "\n\n  invoker processing event3";
  doCommands(event3);
}
```



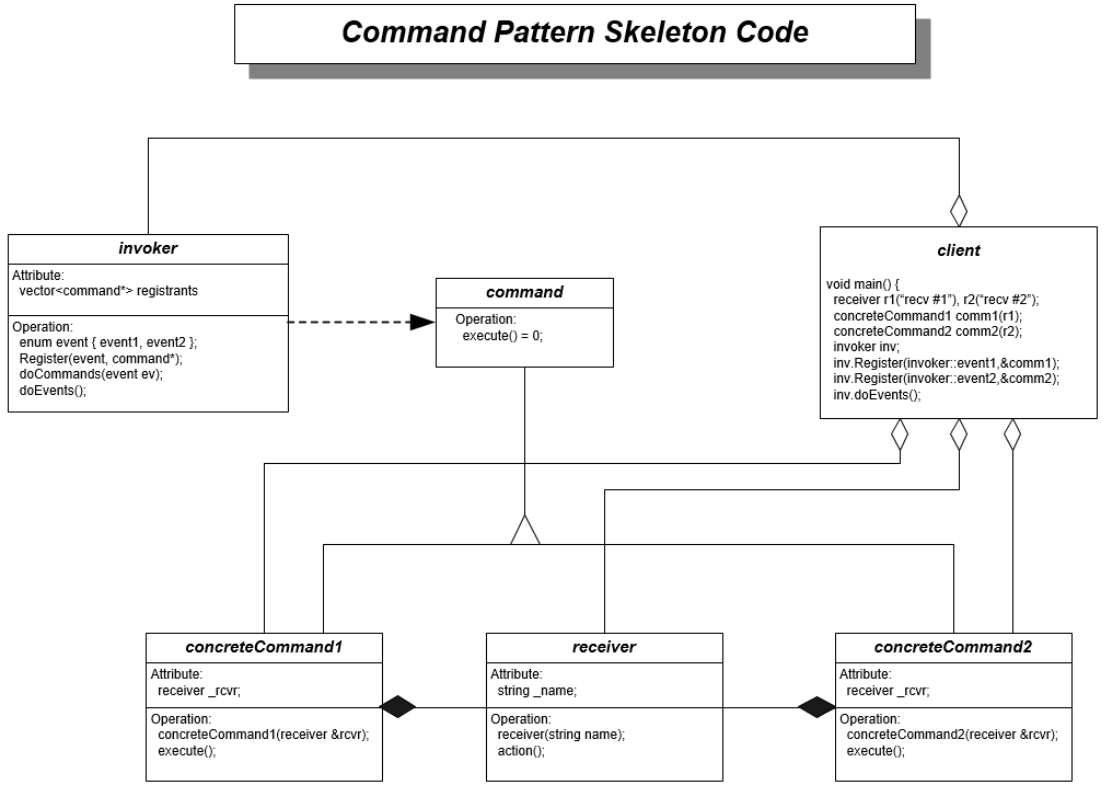
Client Processing

```
cout << "\n  Demonstrate Command Pattern "
     << "\n ============================";

receiver r1("receiver r1");
receiver r2("receiver r2");

concreteCommand1 comm1(r1);
concreteCommand2 comm2(r2);

invoker inv;
inv.Register(invoker::event1,&comm1);   // client #1
inv.Register(invoker::event2,&comm1);   // client #1
inv.Register(invoker::event2,&comm2);   // client #2
inv.Register(invoker::event3,&comm2);   // client #2

inv.doEvents();         // clients get notified here
```



Client Processing

```
 Demonstrate Command Pattern
============================

 invoker processing event1
 concreteCommand1 executing
 receiver r1 receiving action of command object

 invoker processing event2
 concreteCommand1 executing
 receiver r1 receiving action of command object
 concreteCommand2 executing
 receiver r2 receiving action of command object

 invoker processing event3
 concreteCommand2 executing
 receiver r2 receiving action of command object
```

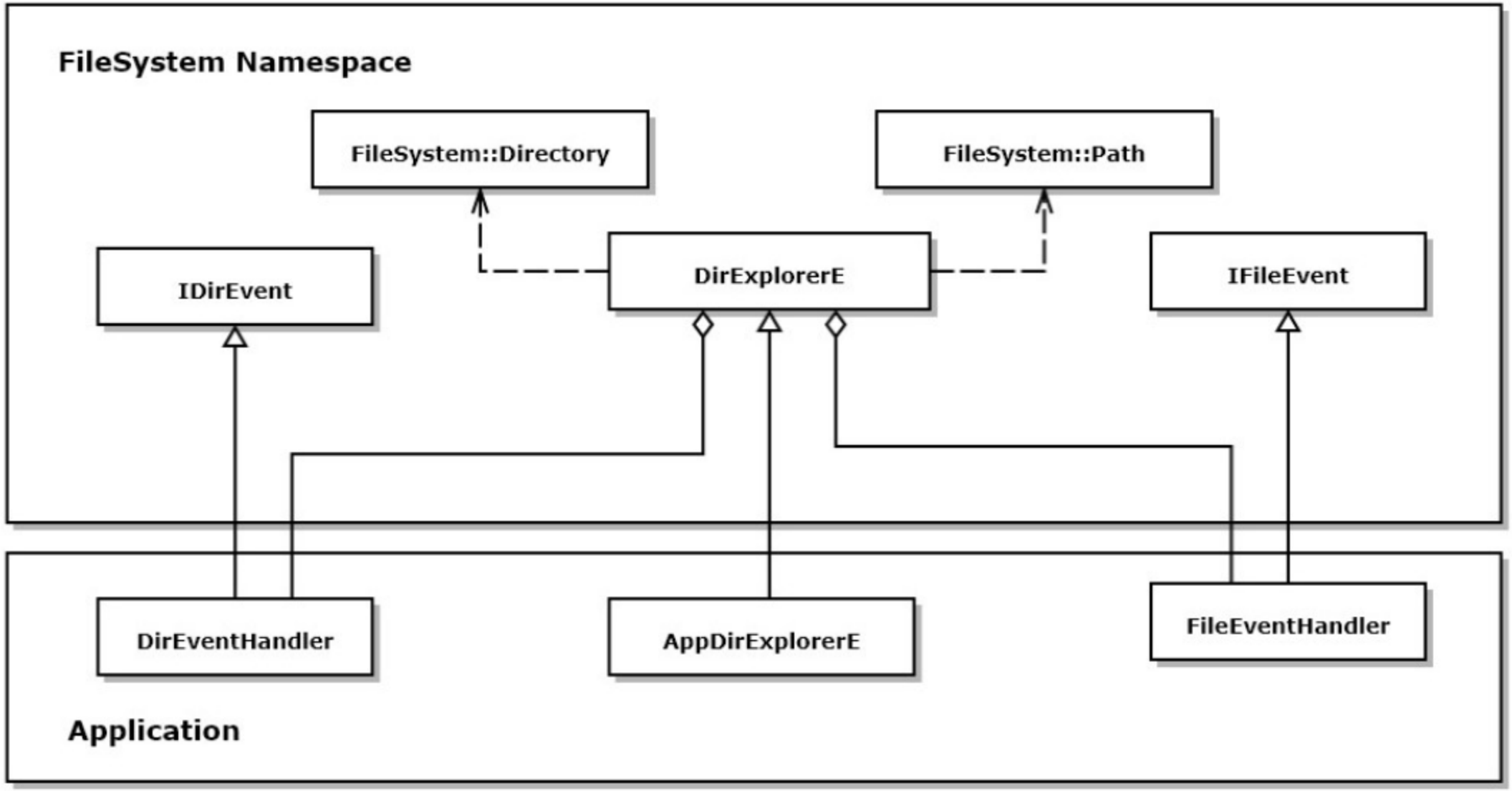


Directory Navigator using Command Pattern

Demo DirectoryExplorer-Events

Handouts\Repository\Cpp\DirectoryNavigator

DirectoryExplorer-Events in Repository\Cpp



DirectoryExplorer-Events Output

```
 Demonstrate DirExplorer-Events, ver1.2
=======================================
 Command Line: .. /s /a /h *.* 7

 Application modified done() invoked
 dir--->  C:\su\temp\DirectoryNavigator
 file-->    DirectoryNavigator.sln
 file-->    DirectoryNavigator.zip
 file-->    logFile.txt
 dir--->  C:\su\temp\DirectoryNavigator\.vs\DirectoryNavigator\v15
 file-->    .suo
 file-->    Browse.VC.db
 file-->    Browse.VC.opendb
 file-->    Solution.VC.db
 file-->    Solution.VC.db-shm
 file-->    Solution.VC.db-wal

 processed 9 files in 2 directories
 Application modified done() invoked
 stopped because max number of files exceeded

Press any key to continue . . .
```