

Design Patterns

Estimated time for completion: 1 hr.

Overview:

Learn to combine many patterns to solve a problem.

Goals:

* Use Encapsulation
* Use the Observer Pattern

Lab Notes:

Enter any special setup-issues etc for the lab..

Create checking account

We want to add some business rules to the account class without modifying Account. The first rule is if anyone deposits more than $10,000 then print “Report to the government”. The second rule says if the balance ever falls below 0, then print “Balance is below 0”.

Steps:

1. Create a class CheckingAccount that encapsulates the Account class.
2. Apply the observer pattern to the CheckingAccount.
   1. Define a delegate type that takes the amount and balance as parameters, and returns void.
   2. Create an event for Deposit, and another for Withdraw using this delegate type.
3. Raise the event in the appropriate methods.
4. Compile and run. There should be no change yet because we have not added rules yet.
5. Now add the rules in the Main method.

Bonus: Add short-circuiting rules

The code right now only observes the events, but cannot affect the program logic. Add a new rule which stops someone from withdrawing money if it would result in a negative balance. You will need to use Delegate.GetInvocationList to iterate over the delegates and examine their return values.