



Concurrent Programming

Pass Task 03: Safety Issues and Locks

Overview

Create a simple account class, break it with some concurrent logic, then protect it with locks.

Pass Task 03 — Submission Details and Assessment Criteria

You must submit the following files to Doubtfire:

- Broken account source code (Account.cs)
- Fixed account source code (Account.cs)
- Test class (Test.cs)

Instructions

Create a concurrent program that deposits and withdraws money from an account in \$1 increments.

- Create an Account class - private balance, balance property, deposit and withdraw methods (using parameters for amount to withdraw).
 - Use .NET's decimal type to represent currency values
 - No concurrency control
 - No limit on deposits or withdraws
- Create an Account Test program that
 - Create a thread to deposit \$1.00 a total of 1,000,000 times.
 - Create a thread to withdraw \$1.00 a total of 1,000,000 times.
 - Output the balance at the end of each thread - and at **no** other time!

Explain the results.

Fix the issue in the Account class, and re-run to see that it is working.