C++ Stock Market

Testarea Sistemelor Software

Alexandru-Christian Codarcea
Florin-Alexandru Gavrilă
Eduard-Valentin Dumitrescul

Prezentare Generală

Acest proiect reprezintă dezvoltarea unei aplicații simple de trading

Funcționalități principale:

- adăugare fonduri
- plasare de ordine de piață
- executarea automată a ordinelor

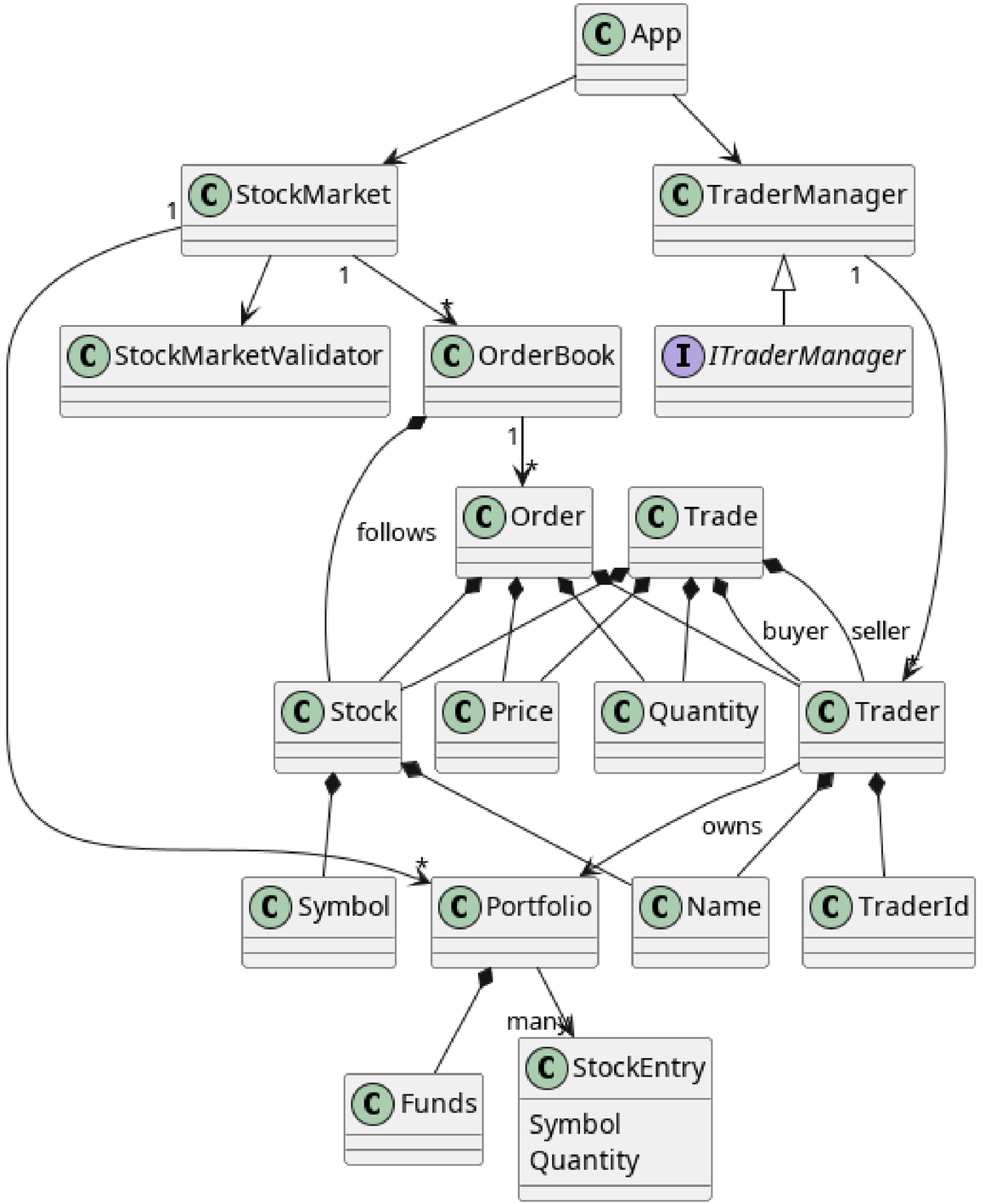
Testare atentă

- testare unitara
- testare de integrare
- testare end-to-end
- mutanți

Arhitectură

Principii POO:

- encapsulare
- abstractizare
- moștenire
- polimorfism



Configurație Software

C++ 20

limbaj de programare

CMake >= 3.30

sistem de build

googletest 1.14.0

framework pentru testare

Windows 11, Ubuntu 24.04, EndeavourOS 6.13

sisteme de operare

Strategii de Testare

01

Testare unitară

Pentru constrângeri stabilite la început și funcționalități simple:

- fondurile sunt mereu între 0 și un miliard
- menținerea ordinelor în ordonate corespunzător tipului
- precum și alte scenarii

02

Testare de integrare

Pentru verificarea cooperării dintre componente:

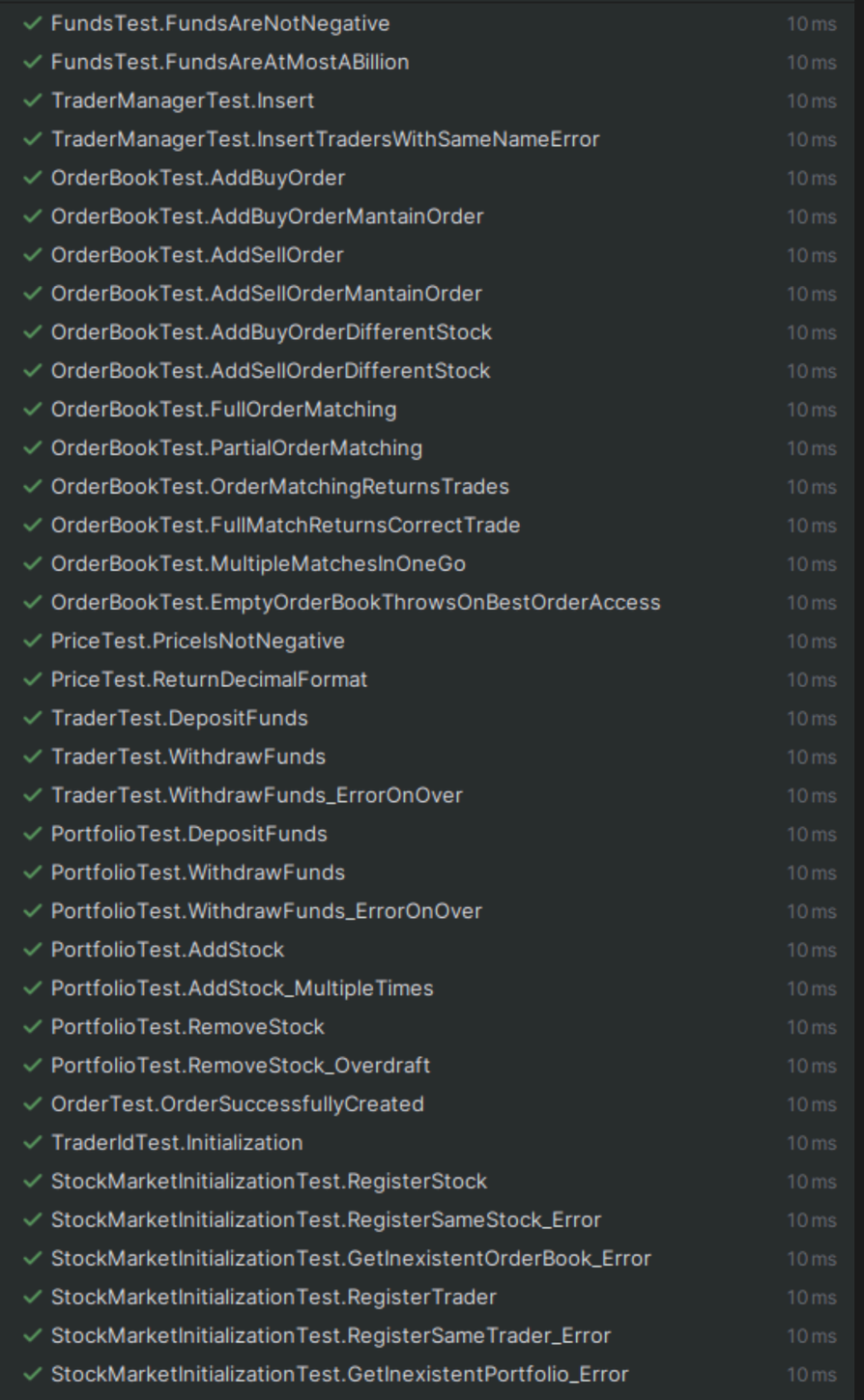
- plasarea unui ordin (StockMarket, Portfolio)
- înregistrarea unui utilizator într-o piață de acțiuni (Trader, StockMarket, Portfolio)

03

Testare end-to-end

Pentru testarea unui întreg flux de acțiuni:

- adăugare fonduri
- plasare ordin de tip sell
- plasare ordin de tip buy
- verificare executare



Testare Riguroasă

Proces în 3 pași:

- adăugare de teste semnificative
- implementarea funcționalității
- rularea testelor și repararea eventualelor erori

O suită de 44 de teste menite să detecteze erorile din implementare

Generare de Mutanți

Script pentru generarea mutanților

main_mutants.cpp

Rularea testelor pe codul modificat

și observarea rezultatului

Imbunătățirea suitei de teste

în momentul în care mutanții trec toate testele

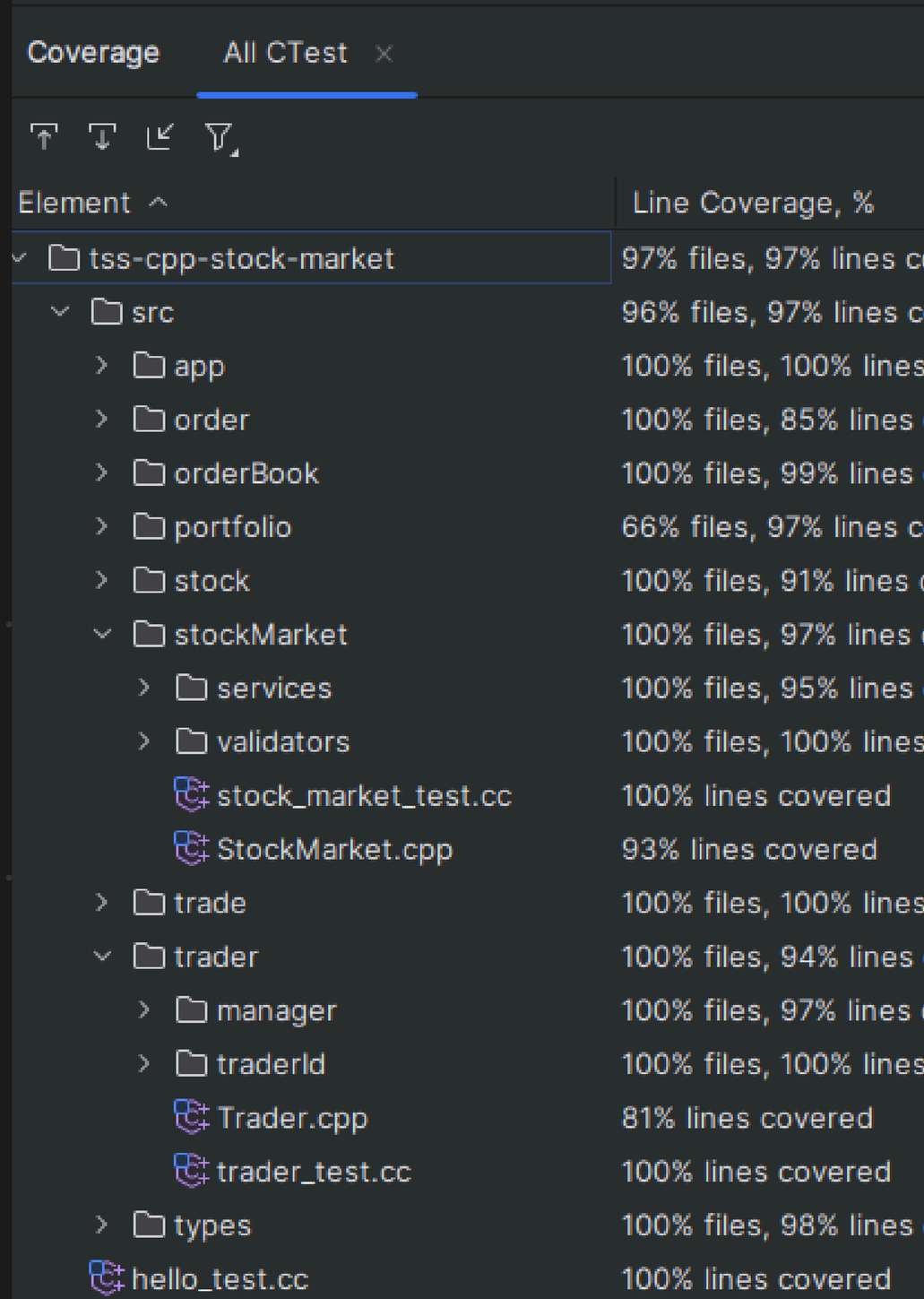
Analiza Testelor

Efecuată cu ajutorul tool-ului integrat implicit in IDE-ul CLion.

Acoperirea codului 97%

Acoperirea liniilor 97%

Acoperirea ramurilor 49%



Demo