UNIVERSIDADE FEDERAL DA PARAÍBA – CENTRO DE INFORMÁTICA

POO - Programação Orientada a Objetos - 2024.2

Prof. Carlos Eduardo Batista

Exercício Prático 1 (Prova 1)

- **Entrega por email: bidu @ ci . ufpb . br**
- **Pontuação para a primeira prova**
- **Prazo para valer pontuação máxima (3,0): até 23h59 de 06/02/2025**
- **Prazo para valer pontuação mínima (2,0): até 09h59 de 11/02/2025**
- O título do e-mail deve conter (substituir nome e matrícula): **"[POO-20242-E001] NOME DO ALUNO – MATRICULA"**.
- Arquivo de entrega deve anexar todos os códigos fonte em C/C++ dentro de um diretório nomeado **"MATRICULA_POO-20242-E001"** o qual deve ser comprimido em um arquivo ZIP (**"MATRICULA_POO-20242-E001.zip"**).
- O arquivo ZIP deve conter obrigatoriamente um arquivo de texto chamado **README.txt** (ou **README.md**) contemplando todas as instruções de compilação e execução, além de qualquer observação que se fizer necessária para correção.
- **O código deverá ser todo comentado**, principalmente nas partes relacionadas com o solicitado no exercício.
- **O NÃO ATENDIMENTO ÀS INSTRUÇÕES IMPLICARÁ NA NÃO CORREÇÃO DO EXERCÍCIO**.
- **TRABALHO INDIVIDUAL** - plágio será punido com a não correção do exercício.

Sistema de Exchange de Criptomoedas

Você foi contratado para desenvolver o sistema backend de uma exchange de criptomoedas em C++. O sistema deve gerenciar carteiras e transações simples entre usuários. Utilize os conceitos de encapsulamento, construtores e destrutores, declaração de objetos e alocação dinâmica de memória.

Descrição das classes:

Carteira:

1. Crie uma classe Carteira com os seguintes atributos:
    - endereco (char[50])
    - saldo_btc (float)
    - saldo_eth (float)
    - saldo_brl (float)
2. Implemente métodos para:
    - alterar_saldo()
    - exibir_saldo()

Transacao:

1. Crie uma classe Transacao com os seguintes atributos:
    - tipo (char[20]) // "COMPRA" ou "VENDA"
    - quantidade (float)
    - preco_unitario (float)
    - carteira_origem (Carteira*)
    - carteira_destino (Carteira*)
2. Implemente métodos para:
    - calcular_total()
    - executar_transacao()
    - exibir_detalhes()

Exchange:

1. Crie uma classe Exchange com os seguintes atributos:
    - carteiras (Carteira*) // array dinâmico
    - num_carteiras (int)
    - transacoes (Transacao*) // array dinâmico
    - num_transacoes (int)
2. Implemente métodos para:
    - adicionar_carteira()
    - realizar_transacao()
    - exibir_carteiras()

Métodos Construtores e Destrutores:

- Implemente construtores para inicializar os atributos das classes
- Garanta que a memória alocada dinamicamente seja liberada nos destrutores
- Utilize new/delete para gerenciamento de memória

Encapsulamento:

- Declare os atributos das classes como private
- Crie métodos públicos para acessar e modificar os atributos
- Implemente gets/sets quando necessário

Funcionalidade do Programa:

- Permita criar novas carteiras
- Realize transações entre carteiras
- Exiba saldos e histórico de transações
- Use cout/cin para entrada e saída de dados

Requisitos específicos:

- Use apenas arrays dinâmicos simples (com new/delete)
- Implemente entrada/saída usando cout/cin
- Faça tratamento básico de saldos insuficientes
- Use apenas tipos básicos de dados (int, float, char[])

Implementação da função main():

Você deve implementar uma função main() que demonstre o funcionamento do sistema com pelo menos:

- Criação de 3 carteiras com saldos iniciais diferentes
- Realização de pelo menos 4 transações:
  - Uma compra bem sucedida de BTC
  - Uma venda bem sucedida de ETH
  - Uma tentativa de compra com saldo insuficiente
  - Uma venda com quantidade maior que o disponível
- Exibição dos saldos das carteiras antes e depois das transações

Exemplo de saída esperada:

```
=== Estado Inicial das Carteiras ===
Carteira 1 (0x123f...):
BTC: 1.5000
ETH: 2.0000
BRL: 5000.00

[... mais carteiras ...]

=== Realizando Transações ===
Transação 1: Compra de 0.5 BTC
Transação realizada com sucesso!

[... mais transações ...]

=== Estado Final das Carteiras ===
[... exibição final dos saldos ...]
```

Descrição da pontuação (3,0 pontos totais):

- Declaração das Classes: atributos, métodos (encapsulamento) (1,0)
- Declaração das Classes: métodos construtores e destrutores (1,0)
- Implementação do sistema de transações e alocação dinâmica (1,0)
- (Extra) Uso de make/Makefile para compilação (0,5)