****

Programação de soluções computacionais

****

Nome : Júlio César De Lima Moreira
RA : 1232021411

Tendo em vista a proposta do problema de doações, foi elaborado um programa capaz de, receber doações, calcular total de doações e armazenar as informações de doações de forma conjuntiva a uma IA. Fora isso, foi solicitado os requisitos funcionais, crítica a IA e um diagrama de classes durante o processo de elaboração.

Requisitos Funcionais

1. Gerenciamento de Doações:

- O sistema deve permitir o registro de novas doações de diferentes categorias (dinheiro, alimentos, roupas, etc.).
- Para cada doação, o sistema deve registrar o volume doado.
- A data da doação também deve ser capturada.

2. Categorização de Doações de Alimentos e Vestuário:

- Para doações de alimentos, o sistema deve registrar o tipo específico do alimento doado.
- Para doações de roupas, o sistema deve registrar a categoria da roupa (ex: camisas, calças) e a unidade de medida (ex: peças, caixas).

3. Persistência dos Registros de Doação:

- O sistema deve armazenar de forma persistente os registros de doação.
- O código utiliza um arquivo de texto chamado "Contribuicoes.txt" para armazenar os dados no formato CSV.

4. Interface com o Usuário:

- O sistema oferece um menu textual para interação do usuário.

- O usuário pode escolher entre registrar uma nova doação, calcular o volume total de doações ou sair do programa.

5. Tratamento de Erros:

- O sistema tenta lidar com possíveis erros durante operações de arquivo (criação ou leitura do arquivo de doações).
- Parece imprimir mensagens de erro no console em caso de problemas.

6. Relatórios:

- O sistema pode calcular o volume total de todas as doações registradas.

Crítica a IA

Durante o processo de elaboração foi notável que as IAS, forneciam muita informação para diferentes tipos de solicitações, sobre o enunciado proposto pelo trabalho. Sendo assim, a interação entre os processos ficou desmedida, uma vez que ao se gerar os requisitos e o diagrama de classes a codificação se mostrou posteriormente confusa, é como se a ia não acompanhasse os nuances dos requisitos e da funcionalidade das classes. Entregando assim soluções por partes certas, mas que não se interligavam diretamente, contudo no que se refere ao conjunto a codificação ficava confusa e apresentava erros. Tendo isso em mente, uma solução a se pensar foi em ir codando etapa por etapa e ir testando a funcionalidade do sistema, corrigindo juntamente com a iA para assim obter o resultado desejado.

Vale ressaltar que ela não orienta diretamente sobre processos funcionais como o ambiente do lugar onde vai se rodar o código e como executar nele, quais extensões seriam necessárias instalar para poder aplicar o código. Demonstrando assim uma busca a parte e um embate entre solução x orientação de processo. Afinal, conforme estudado na biografia BOND, Martin et al. Aprenda J2EE: Com EJB, JSP, Servlets, JNDI, JDBC e XML O livro discuti como os requisitos de sistema influenciam decisões de arquitetura, design de software e otimização de desempenho, aspectos importantes para sistemas Java, tendo assim a ia um papel no final das contas de suporte para a projeção de um sistema, que envolve mais complexidades do que uma busca por resultados e soluções baseados em perguntas e enunciados, mas sim em uma arquitetura que envolva conhecimentos práticos, noção de desenvolvimento e análises de ambientes.

Diagrama de classes

- **Classe Contribuicao:**
  - **Atributos:**
    - categoria: Representa o tipo de doação (dinheiro, alimentos, roupas etc.).
    - volume: Representa a quantidade da doação (valor em reais, kg de alimentos, número de peças de roupas etc.).
    - dataRegistro: Representa a data em que a doação foi registrada.
    - tipoAlimento (opcional): Somente presente para doações de alimentos, especifica o tipo de alimento doado.
    - categoriaVestuario (opcional): Somente presente para doações de roupas, especifica a categoria da roupa doada.
    - medida (opcional): Somente presente para doações de roupas, especifica a medida da roupa doada.
  - **Métodos:**
    - getCategoria(): Retorna a categoria da doação.
    - getVolume(): Retorna o volume da doação.
    - getDataRegistro(): Retorna a data de registro da doação.
    - getTipoAlimento(): Retorna o tipo de alimento doado (somente para doações de alimentos).
    - getCategoriaVestuario(): Retorna a categoria da roupa doada (somente para doações de roupas).
    - getMedida(): Retorna a medida da roupa doada (somente para doações de roupas).
    - toString(): Retorna uma representação textual da doação, incluindo todas as suas propriedades.
    - paraFormatoCSV(): Converte a doação para o formato CSV (valores separados por vírgula).
    - static Contribuicao deFormatoCSV(String): Cria uma nova instância de Contribuicao a partir de uma string no formato CSV.
- **Classe GerenciadorContribuicoes:**

  - **Atributos:**
    - registroContribuicoes: Uma lista que armazena todas as doações registradas.
    - CAMINHO_ARQUIVO: Uma constante que define o caminho para o arquivo onde as doações são salvas.
  - **Métodos:**
    - GerenciadorContribuicoes(): Construtor da classe, inicializa a lista de doações e tenta carregar as doações do arquivo.
    - adicionarContribuicao(Contribuicao): Adiciona uma nova doação à lista de doações e salva-a no arquivo.
    - salvarContribuicao(Contribuicao): Salva uma doação no arquivo.

        - carregarContribuicoes(): Carrega as doações do arquivo para a lista de doações.
        - calcularTotalDoacoes(): Calcula e retorna o volume total de todas as doações.
        - getRegistroContribuicoes(): Retorna a lista de doações.
        -

- **Classe Trabalho_Final (Que se refere ao sistema criado):**

    - **Método:**
        - main(String[]): Ponto de entrada do programa. Cria um objeto GerenciadorContribuicoes, apresenta um menu ao usuário e lida com as interações do usuário (adicionar doações, calcular total de doações, sair).

Relações entre as Classes

A classe Contribuicao representa um doador individual, uma doação única. Já a classe GerenciadorContribuicoes é responsável por gerenciar a coleção de doações, incluindo adicionar novas doações, salvar e carregar doações de um arquivo, calcular o total de doações e fornecer acesso à lista de doações.
A classe Trabalho_Final é o ponto de entrada do programa e interage com o usuário, utilizando a classe GerenciadorContribuicoes para gerenciar as doações.

Observações

A classe Contribuicao possui propriedades opcionais (tipoAlimento, categoriaVestuario, medida) que só são utilizadas para doações de alimentos e roupas, respectivamente. Isso permite que a classe Contribuicao seja flexível e acomode diferentes tipos de doações.O gerenciador (GerenciadorContribuicoes) persiste as doações em um arquivo (CAMINHO_ARQUIVO) para que os dados sejam mantidos mesmo após o término do programa.