

Revisão

QUESTÃO 1 - Sistema de Gestão de Notas de Alunos

Um professor deseja desenvolver um sistema para gerenciar as notas dos alunos de sua turma. Esse sistema deve permitir cadastrar alunos, registrar suas notas e calcular a média da turma. Além disso, o sistema precisa salvar essas informações em um arquivo e ser capaz de lidar com possíveis erros de entrada e saída. Siga os requisitos abaixo para implementar a solução.

1. Estrutura de dados :

- Use uma lista para armazenar um dicionário para cada aluno. Cada dicionário deve conter os seguintes dados: nome (string), notas (lista de floats), e media (float).

2. Funcionalidades do programa :

- Crie uma função `adicionar_aluno` que solicite o nome de um aluno e as suas notas (no mínimo 2 e no máximo 5 notas). Essa função deve calcular a média das notas e armazenar esses dados no dicionário do aluno. O aluno deve então ser adicionado à lista de alunos.
- Crie uma função `ordenar_alunos` que ordene a lista de alunos por média de forma decrescente (do aluno com maior média para o menor).
- Crie uma função `salvar_em_arquivo` que salve os dados de todos os alunos em um arquivo de texto (`alunos.txt`). Cada linha do arquivo deve conter o nome do aluno e a sua média, separados por uma vírgula.

3. Tratamento de Exceções :

- Utilize exceções para garantir que as notas inseridas sejam válidas (ou seja, números entre 0 e 10). Caso o usuário insira uma nota inválida, solicite a nota novamente.
- Trate possíveis erros ao salvar os dados no arquivo (como problemas de permissão ou de disco cheio). Caso ocorra um erro, exiba uma mensagem apropriada.

4. Execução do programa :

- No final, exiba os alunos ordenados pela média e salve o resultado no arquivo. Caso o arquivo já exista, permita que o usuário escolha entre sobrescrever ou cancelar a operação.

Dicas para implementação :

- Teste o programa para verificar o tratamento correto das exceções.
- Organize seu código em funções conforme os requisitos acima.

Exemplo de saída esperada:

Nome do aluno: João

Notas: 8.0, 7.5

Nome do aluno: Maria

Notas: 9.0, 8.5, 10.0

Alunos ordenados por média:

Maria - Média: 9.17

João - Média: 7.75

Os dados foram salvos no arquivo 'alunos.txt'.

QUESTÃO 2 - Biblioteca de Livros Digitais

Você foi contratado para desenvolver um sistema básico de biblioteca digital, onde os usuários podem registrar informações sobre livros, pesquisar e ordenar os registros. O sistema deve lidar com erros de entrada de dados e ser capaz de salvar e carregar informações de um arquivo. Siga as instruções abaixo para implementar a solução.

1. Estrutura de Dados :

- Use uma lista para armazenar um dicionário para cada livro. Cada dicionário deve conter os seguintes dados: titulo (string), autor (string), ano (int) e paginas (int).

2. Funcionalidades do Programa :

- Crie uma função `adicionar_livro` que solicite ao usuário o título, o autor, o ano de publicação e o número de páginas do livro. Essa função deve armazenar esses dados no dicionário do livro e adicioná-lo à lista de livros.
- Crie uma função `listar_livros` que mostre todos os livros cadastrados.
- Crie uma função `ordenar_livros` que permita ao usuário ordenar a lista de livros por ano de publicação ou pelo número de páginas. O usuário deve escolher a ordem (crescente ou decrescente).
- Crie uma função `salvar_livros` que salve as informações dos livros em um arquivo de texto (`biblioteca.txt`), com cada linha contendo os dados do livro separados por vírgulas.
- Crie uma função `carregar_livros` que leia o arquivo `biblioteca.txt` (caso exista) e carregue os dados para a lista de livros.

3. Tratamento de Exceções :

- Utilize exceções para garantir que o ano e o número de páginas sejam válidos (ou seja, inteiros positivos). Caso o usuário insira valores inválidos, solicite os dados novamente.
- Ao salvar ou carregar dados do arquivo, trate exceções relacionadas a erros de entrada e saída (como problemas de leitura, permissões, ou falta de espaço). Exiba mensagens apropriadas para cada erro.

4. Execução do Programa :

- O programa deve oferecer ao usuário um menu com as opções: adicionar livro, listar livros, ordenar livros, salvar dados e carregar dados.
- No final da execução, pergunte ao usuário se ele deseja salvar os dados no arquivo antes de sair.

Dicas para implementação :

- Verifique se o arquivo biblioteca.txt existe antes de tentar carregá-lo.
- Teste o programa para garantir que ele lide corretamente com entradas inválidas e erros de leitura e gravação de arquivos.

Exemplo de Saída Esperada :

Bem-vindo à Biblioteca Digital!

Escolha uma opção:

1. Adicionar livro
2. Listar livros
3. Ordenar livros
4. Salvar dados em arquivo
5. Carregar dados do arquivo
6. Sair

> 1

Título do livro: O Senhor dos Anéis

Autor: J.R.R. Tolkien

Ano de publicação: 1954

Número de páginas: 1178

Livro adicionado com sucesso!

Deseja salvar os dados antes de sair? (S/N): S

Os dados foram salvos no arquivo 'biblioteca.txt'.

QUESTÃO 3: Sistema de Gestão de Estoque de Loja

Uma loja precisa de um sistema para gerenciar o estoque de produtos. Esse sistema deve permitir cadastrar produtos, atualizar quantidades, listar produtos em ordem de preço ou quantidade, e salvar as informações em um arquivo para garantir a persistência dos dados. Siga os requisitos abaixo para implementar a solução.

1. Estrutura de Dados :

- Use uma lista para armazenar um dicionário para cada produto. Cada dicionário deve conter os seguintes dados: nome (string), categoria (string), preço (float), quantidade (int).

2. Funcionalidades do Programa :

- Crie uma função `adicionar_produto` que solicite o nome, a categoria, o preço e a quantidade do produto. Valide os dados e, caso estejam corretos, armazene-os em um dicionário e adicione à lista de produtos.
- Crie uma função `atualizar_quantidade` que permita ao usuário selecionar um produto e atualizar sua quantidade no estoque. Esta função deve permitir o aumento ou a redução da quantidade.
- Crie uma função `listar_produtos` que exiba todos os produtos cadastrados.
- Crie uma função `ordenar_produtos` que permita ordenar a lista de produtos por preço ou quantidade, em ordem crescente ou decrescente, conforme a escolha do usuário.
- Crie uma função `salvar_estoque` que salve as informações dos produtos em um arquivo (`estoque.txt`), com cada linha contendo o nome, a categoria, o preço e a quantidade do produto, separados por vírgulas.
- Crie uma função `carregar_estoque` que leia o arquivo `estoque.txt` (caso ele exista) e carregue as informações para a lista de produtos.

3. Tratamento de Exceções :

- Utilize exceções para validar que o preço e a quantidade sejam valores numéricos (o preço deve ser um número positivo e a quantidade um número inteiro não negativo).
- Ao salvar ou carregar dados do arquivo, trate exceções relacionadas a erros de leitura ou gravação, como permissões, espaço insuficiente ou arquivo inexistente. Exiba uma mensagem apropriada para cada erro.

4. Execução do Programa :

- O programa deve apresentar um menu ao usuário com as opções: adicionar produto, atualizar quantidade, listar produtos, ordenar produtos, salvar estoque e carregar estoque.
- Antes de sair, pergunte ao usuário se ele deseja salvar os dados no arquivo.

Dicas para Implementação :

- Verifique se o arquivo `estoque.txt` existe antes de carregar os dados.
- Teste o programa para verificar o tratamento adequado de exceções e para assegurar que os dados são armazenados corretamente.

Exemplo de Saída Esperada :

Bem-vindo ao Sistema de Gestão de Estoque!

Escolha uma opção:

1. Adicionar produto
2. Atualizar quantidade
3. Listar produtos
4. Ordenar produtos
5. Salvar dados em arquivo
6. Carregar dados do arquivo
7. Sair

> 1

Nome do produto: Teclado Mecânico

Categoria: Eletrônicos

Preço: 299.99

Quantidade: 15

Produto adicionado com sucesso!

Deseja salvar o estoque antes de sair? (S/N): S

Os dados foram salvos no arquivo 'estoque.txt'.