



Relatório discente de acompanhante

Objetivos da prática

1. Utilizar herança e polimorfismo na definição de entidades.
2. Utilizar persistência de objetos em arquivos binários.
3. Implementar uma interface cadastral em modo texto.
4. Utilizar o controle de exceções da plataforma Java.
5. arquivos binários.

1º Procedimento | Criação das Entidades e Sistema de Persistência.

A fase inicial do processo envolveu a criação das entidades e a implementação do sistema de persistência dos dados, seguido por testes de funcionalidade. As classes Pessoa, PessoaFisica (derivada da classe Pessoa) e PessoaJuridica (também derivada de Pessoa) foram utilizadas para criar instâncias que foram armazenadas dentro das classes PessoaJuridicaRepo e PessoaFisicaRepo. Estas classes, por meio do método de inserção, registravam os objetos em uma estrutura de dados do tipo ArrayList.

Para testar a persistência dos dados, foram instanciadas duas classes de repositório para cada categoria (Pessoa Física e Pessoa Jurídica). Na primeira, dois objetos foram adicionados manualmente através do código e, em seguida, o método de persistência foi invocado, especificando o nome do arquivo no qual os dados seriam salvos. Posteriormente, na segunda instância, que até então não continha nenhum objeto armazenado, foi executado o método "recuperar", passando o mesmo nome de arquivo como parâmetro. Com isso, o repositório carregou os objetos que haviam sido armazenados pela outra instância anteriormente.

Análise e Considerações Finais:

- a. Uma das maiores vantagens da utilização de herança é a capacidade de reutilização de código, o que simplifica significativamente a manutenção. Isso se deve à clareza na visibilidade das relações entre classes e à organização estrutural que proporciona.
- b. A presença da interface `Serializable` se torna fundamental, pois sinaliza à JDK (Java Development Kit) que a classe pode ser serializada, ou seja, pode ser convertida em uma sequência de bytes para ser armazenada ou transmitida.
- c. A introdução da API `Stream` no Java, parte essencial da biblioteca `Java.util.stream`, trouxe consigo um paradigma funcional inovador. Esta abordagem permite lidar com coleções de dados de maneira mais concisa e expressiva, ampliando as possibilidades de manipulação dos dados de forma eficiente.
- d. Em ambientes Java, ao lidar com a persistência de dados em arquivos, é comum recorrer ao padrão de projeto "DAO" (Data Access Object). Este padrão oferece uma abstração para o acesso aos dados, separando a lógica de negócios da manipulação dos dados, promovendo assim uma melhor organização e modularidade no sistema.

2º Procedimento | Criação do Cadastro em Modo Texto

Este processo envolveu a criação de uma interface destinada ao usuário para que ele pudesse interagir e executar as ações disponíveis no programa.

O menu foi construído utilizando um laço "Do while", sendo que a condição de saída é acionada ao receber a resposta 0. As funcionalidades de cada opção são executadas através de um bloco "switch case".

Nas opções 1, 2, 3, 4 e 5, o sistema solicita a escolha entre os tipos de Pessoa, se Física ou Jurídica, para prosseguir com as operações correspondentes.

Para as opções de persistência e recuperação de dados, é solicitado um nome de arquivo que será utilizado como prefixo.

"Todas as funcionalidades foram testadas e estão operando conforme o esperado em termos de utilidade.

Relatório discente de acompanhante:

a. Elementos estáticos podem ser acessados sem a necessidade de instanciar a classe correspondente. Por isso, o método main utiliza esse modificador, pois sem ele, nada seria executado.

b. O Scanner é utilizado para observar e, por meio de seus métodos, capturar dados na aplicação.

c. A utilização de classes proporcionou uma organização significativa no código. A abordagem de se aproximar da realidade, onde cada objeto na aplicação reflete elementos e realiza ações correspondentes à vida real, facilita consideravelmente a lógica da aplicação."