

# Missão Prática | Nível 1 | Mundo 3

Material de orientações para desenvolvimento da missão prática do 1º nível de conhecimento.

RPG0014 - Iniciando o caminho pelo Java

Implementação de um cadastro de clientes em modo texto, com persistência em arquivos, baseado na tecnologia Java.

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. No final do projeto, o aluno terá implementado um sistema cadastral em Java, utilizando os recursos da programação orientada a objetos e a persistência em arquivos binários.

As práticas devem ser feitas individualmente.

Materiais necessários para a prática

- JDK e IDE NetBeans.
- Equipamentos: Computador com JDK e NetBeans instalados.

Desenvolvimento da prática

Vamos colocar a mão na massa?! Siga as instruções abaixo para desenvolvimento desta missão.

1º Procedimento | Criação das Entidades e Sistema de Persistência

- 1. Criar um projeto do tipo Ant..Java Application no NetBeans, utilizando o nome CadastroPOO para o projeto.

- 2. Criar um pacote com o nome "model", para as entidades e gerenciadores.
- 3. No pacote model criar as entidades, com as seguintes características:
  - a. Classe Pessoa, com os campos id (inteiro) e nome (texto), método exibir, para impressão dos dados, construtor padrão e completo, além de getters e setters para todos os campos.
  - b. Classe PessoaFisica, herdando de Pessoa, com o acréscimo dos campos cpf (texto) e idade (inteiro), método exibir polimórfico, construtores, getters e setters.
  - c. Classe PessoaJuridica, herdando de Pessoa, com o acréscimo do campo cnpj (texto), método exibir polimórfico, construtores, getters e setters.
  - d. Adicionar interface Serializable em todas as classes
- 4. No pacote model criar os gerenciadores, com as seguintes características:
  - a. Classe PessoaFisicaRepo, contendo um ArrayList de PessoaFisica, nível de acesso privado, e métodos públicos inserir, alterar, excluir, obter e obterTodos, para gerenciamento das entidades contidas no ArrayList.
  - b. Classe PessoaJuridicaRepo, com um ArrayList de PessoaJuridica, nível de acesso privado, e métodos públicos inserir, alterar, excluir, obter e obterTodos, para gerenciamento das entidades contidas no ArrayList .
  - c. Em ambos os gerenciadores adicionar o método público persistir, com a recepção do nome do arquivo, para armazenagem dos dados no disco.
  - d. Em ambos os gerenciadores adicionar o método público recuperar, com a recepção do nome do arquivo, para recuperação dos dados do disco
  - e. Os métodos persistir e recuperar devem ecoar (throws) exceções
  - f. O método obter deve retornar uma entidade a partir do id
  - g. Os métodos inserir e alterar devem ter entidades como parâmetros
  - h. O método excluir deve receber o id da entidade para exclusão
  - i. O método obterTodos deve retornar o conjunto completo de entidades.
- 5. Alterar o método main da classe principal para testar os repositórios:
  - a. Instanciar um repositório de pessoas físicas (repo1).
  - b. Adicionar duas pessoas físicas, utilizando o construtor completo.
  - c. Invocar o método de persistência em repo1, fornecendo um nome de arquivo fixo, através do código.
  - d. Instanciar outro repositório de pessoas físicas (repo2).
  - e. Invocar o método de recuperação em repo2, fornecendo o mesmo nome de arquivo utilizado anteriormente.
  - f. Exibir os dados de todas as pessoas físicas recuperadas.
  - g. Instanciar um repositório de pessoas jurídicas (repo3).
  - h. Adicionar duas pessoas jurídicas, utilizando o construtor completo.
  - i. Invocar o método de persistência em repo3, fornecendo um nome de arquivo fixo, através do código.
  - j. Instanciar outro repositório de pessoas jurídicas (repo4).
  - k. Invocar o método de recuperação em repo4, fornecendo o mesmo nome de arquivo utilizado anteriormente.
  - l. Exibir os dados de todas as pessoas jurídicas recuperadas.
- 6. Ajustar as características para obter uma execução como a seguinte:

7. Verificar as funcionalidades solicitadas e os arquivos gerados

✔ Resultados esperados

1. É importante que o código seja organizado.
2. Outro ponto importante é explorar as funcionalidades oferecidas pelo NetBeans para melhoria da produtividade.
3. Nesse exercício, é esperado que o estudante demonstre as habilidades básicas para a programação Java com persistência em arquivo.

📄 Relatório discente de acompanhamento

Os Relatórios de Práticas deverão ser confeccionados em arquivo no formato PDF, com a Logo da Universidade, nome do Campus, nome do Curso, nome da Disciplina, número da Turma, semestre letivo, nome dos integrantes da Prática. Além disso, o projeto deve ser armazenado em um repositório no GIT e o respectivo endereço deve constar na documentação. A documentação do projeto deve conter:

1. Título da Prática;
2. Objetivo da Prática;
3. Todos os códigos solicitados neste roteiro de aula;
4. Os resultados da execução dos códigos também devem ser apresentados;
5. Análise e Conclusão:
  - a. Quais as vantagens e desvantagens do uso de herança?
  - b. Por que a interface Serializable é necessária ao efetuar persistência em arquivos binários?
  - c. Como o paradigma funcional é utilizado pela API stream no Java?
  - d. Quando trabalhamos com Java, qual padrão de desenvolvimento é adotado na persistência de dados em arquivos?

🔑 2º Procedimento | Criação do Cadastro em Modo Texto

1. Alterar o método **main** da classe principal do projeto, para implementação do cadastro em modo texto:
  - a. Apresentar as opções do programa para o usuário, sendo 1 para incluir, 2 para alterar, 3 para excluir, 4 para exibir pelo id, 5 para exibir todos, 6 para salvar dados, 7 para recuperar dados e 0 para finalizar a execução.
  - b. Selecionada a opção **incluir**, escolher o tipo (Física ou Jurídica), receber os dados a partir do teclado e adicionar no repositório correto.
  - c. Selecionada a opção **alterar**, escolher o tipo (Física ou Jurídica), receber o **id** a partir do teclado, apresentar os dados atuais, solicitar os novos dados e alterar no repositório correto.
  - d. Selecionada a opção **excluir**, escolher o tipo (Física ou Jurídica), receber o **id** a partir do teclado e remover do repositório correto.
  - e. Selecionada a opção **obter**, escolher o tipo (Física ou Jurídica), receber o **id** a partir do teclado e apresentar os dados atuais para a entidade.
  - f. Selecionada a opção **obterTodos**, escolher o tipo (Física ou Jurídica) e apresentar os dados de todas as entidades do repositório correto.
  - g. Selecionada a opção **salvar**, solicitar o **prefixo** dos arquivos e persistir os dados nos arquivos **[prefixo].fisica.bin** e **[prefixo].juridica.bin**.
  - h. Selecionada a opção **recuperar**, solicitar o **prefixo** dos arquivos e obter os dados a partir dos arquivos **[prefixo].fisica.bin** e **[prefixo].juridica.bin**.
  - i. Nas opções **salvar** e **recuperar** devem ser tratadas as exceções.
  - j. Selecionada a opção **sair**, finalizar a execução do sistema.
2. Ajustar as características para obter uma execução como a seguinte.

3. Verificar todas as funcionalidades solicitadas e os arquivos gerados.

✔ Resultados esperados

1. É importante que o código seja organizado.
2. Outro ponto importante é explorar as funcionalidades oferecidas pelo NetBeans para melhoria da produtividade.
3. Nesse exercício, é esperado que o estudante demonstre as habilidades básicas para a programação Java com persistência em arquivo.

📄 Relatório discente de acompanhamento

Os Relatórios de Práticas deverão ser confeccionados em arquivo no formato PDF, com a Logo da Universidade, nome do Campus, nome do Curso, nome da Disciplina, número da Turma, semestre letivo, nome dos integrantes da Prática. Além disso, o projeto deve ser armazenado em um repositório no GIT e o respectivo endereço deve constar na documentação (PDF). A documentação do projeto deve conter:

1. Título da Prática;
2. Objetivo da Prática;
3. Todos os códigos solicitados neste roteiro de aula;
4. Os resultados da execução dos códigos também devem ser apresentados;
5. Análise e Conclusão:
  - a. O que são elementos estáticos e qual o motivo para o método main adotar esse modificador?
  - b. Para que serve a classe Scanner?
  - c. Como o uso de classes de repositório impactou na organização do código?

Observações

Pré-requisitos:

1. Os estudantes precisam instalar o JDK e o NetBeans.

Referência Bibliográfica:

1. <https://stensenime.blob.core.windows.net/hmlgrehpoh/00212ti/01703/index.html>
2. <https://stensenime.blob.core.windows.net/hmlgrehpoh/00212ti/01338/index.html>
3. <https://stensenime.blob.core.windows.net/hmlgrehpoh/00212ti/01662/index.html>
4. Padrões de Projeto. Disponível em <https://refactoring.guru/pt-br/design-patterns>. Acessado em 01/03/2023.
5. Os 4 pilares da Programação Orientada a Objetos. Disponível no endereço <https://www.devmedia.com.br/os-4-pilares-da-programacao-orientada-a-objetos/9264>. Acessado em 01/03/2023.
6. Programação Orientada a Objetos com Java – Dev Media. Disponível em <https://www.devmedia.com.br/programacao-orientada-a-objetos-com-java/18449>. Acessado em 01/03/2023.
7. Tutorial Data Persistence in Java – Code HS. Disponível no endereço <https://codehs.com/tutorial/david/data-persistence-in-java>. Acessado em 01/03/2023.

Entrega da prática

Chegou a hora, gamer!

- 🔥 Armazene o projeto em um repositório no GIT.
- 🔥 Anexar a documentação do projeto (PDF) no GIT.
- 🔥 Compartilhe o link do repositório do GIT com o seu tutor para correção da prática, por meio da **Sala de Aula Virtual**, na aba "Trabalhos" do respectivo nível de conhecimento.
- 🔥 **Ei, não se esqueça de entregar este trabalho na data estipulada no calendário acadêmico!**

Feito com o Microsoft Sway

Crie e compartilhe apresentações, histórias pessoais, relatórios interativos e muito mais.

Introdução

