
Disciplina: Programação Orientada à Objetos (POO)

Professor: Alex Paulo Lopes Batista

Data de Entrega: 12/06/2023

Prova Regimental 2 (P2): Construção de uma aplicação de CRUD de Estudante e Professor em Lista com Programação Orientada à Objetos em NetBeans.

Instruções Iniciais:

1. Leia **atentamente** todas as **instruções iniciais** e as **tarefas do Prova Regimental 2 (P2)** para construir esta aplicação e em caso de dúvidas contate o professor.
2. Crie uma **Aplicação Java** chamada **CRUDListEstudante** conforme abaixo:
 - No menu superior do NetBeans, escolha:
Arquivo > Novo Projeto
 - Na janela **Novo Projeto**, escolha: em **Categorias: Java** e em **Projetos: Aplicação Java**;
 - Na janela **Novo Aplicação Java** configure as seguintes opções para **Nome e Localização**:
 - a. **Nome do Projeto: CRUDListEstudante**
 - b. **Localização do Projeto: C:\fateclins\poo\p2\grupoX**
 - c. **Pasta do Projeto: C:\fateclins\poo\p2\CRUDListEstudante**
 - d. OBS: Marque a opção: **Criar Classe Principal: Main**.
 - Por fim, clique no botão **“Finalizar”** para que uma nova **Aplicação Java** para que o NetBeans possa iniciar o processo de construção do **CRUDListEstudante**.
3. Defina seu **pacote base** com a **URL da Fatec Lins**, em **conjunto** com a **classe Main**, veja o exemplo abaixo:

br.edu.fateclins.crudlistestudante.Main

Prova Regimental 2 (P2) - Tarefas:

Criar o **projeto NetBeans** do **CRUDListEstudante** com os seguintes requisitos:

1. a) Crie um pacote (“**package**”) com o nome **model**, com clique do botão direito do mouse sobre o **pacote base**: **(0,2 ponto)**

Pacote base -> **br.edu.fateclins.crudlistestudante**

Model -> **br.edu.fateclins.crudlistestudante.model**

No **pacote model** crie o **arquivo** que corresponde a **classe do domínio** (**classe do modelo**):

- b) **Estudante.java**: é a **classe Estudante** com os seguintes **atributos privados**: **(0,2 ponto)**

Atributo	Tipo	Descrição
raEstudante	int	Registro do aluno estudante
nomeEstudante	String	Nome do estudante
telefoneEstudante	String	Telefone do estudante
emailEstudante	String	Email do estudante

OBS: Importante todos os atributos da classe Estudante são privados.

Ainda dentro da **classe Estudante** crie:

- c) O **método construtor padrão** e **construtor todos os parâmetros inicializados**; **(0,2 ponto)**
 - d) Os **métodos Getters and Setters** para **construtor padrão**, e; **(0,2 ponto)**
 - e) O **método toString()** com **StringBuilder** para **gerar uma saída de impressão do objeto Estudante**. **(0,2 ponto)**
2. Vá para o **pacote base** e crie o **arquivo**:
 - a) **GerenciamentoListaEstudante.java**: é a **classe responsável pela manipulação do CRUD de estudante** que **permite adicionar (add), editar (update), listar (display) e remover (delete)** da lista de estudante. Também existem **outros métodos** que foram **criados** como: **pesquisar todos os estudantes (find)** e **pesquisar por um determinado estudante específico (findEstudante)**. **(0,5 ponto)**

b) No **pacote base** tem-se a classe **Main.java**: realiza a instanciação da classe **Estudante** e da classe **GerenciamtoListaEstudante** através de um **menu de opções** que permite acessar as **opções de CRUD da lista de estudante**. (0,5 ponto)

No primeiro e segundo exercícios foram feitos nas aulas anteriores e servirão de base para realizar a Prova Regimental 2 (P2).

3. Ainda no **pacote base** inclua código-fonte na classe **Main.java** para:
- a) Criar **um estudante com construtor padrão** e **adicione todos os atributos (getters e setters)** e em seguida **adicionar o estudante na lista de estudante (listaEstudante)**, e; (0,5 ponto)
- b) Criar o **Menu Principal com as opções do CRUD do Estudante**. (0,5 ponto)
4. a) Crie no **pacote model** uma **classe Java** chamada **Professor** com os **seguintes atributos privados**: (0,25 ponto)

Atributo	Tipo	Descrição
rmProfessor	int	Registro do aluno professor
nomeProfessor	String	Nome do professor
telefoneProfessor	String	Telefone do professor
emailProfessor	String	Email do professor

b) A partir dos **atributos privados** (acima) crie **dois métodos construtores**, sendo: (0,25 ponto)

- **Construtor padrão default**, e;
- **Construtor com todos os argumentos (parâmetros) inicializados**.

c) Para **finalizar** essa **Classe Professor** deve ser **gerado** os seus **métodos públicos: métodos Getters e os métodos Setters**. (0,25 ponto)

d) Crie o **método toString()** com **StringBuilder** para **gerar uma saída de impressão do objeto Professor**. (0,25 ponto)

5. Vá para o **pacote base** inclua código-fonte na classe **Main.java** para:
 - a) Criar **um professor com construtor padrão** e **adicione todos os atributos (getters e setters)** e em seguida **adicionar o professor na lista de professor (listaProfessor)**, e;
 - b) Criar o **Menu Professor com as opções do CRUD do Professor**, e;
 - c) Altere o **Menu Principal** para **Menu Estudante**.
6. Volte para o **pacote base** e crie o **arquivo:**
GerenciamentoListaProfessor.java: é a classe responsável pela manipulação do **CRUD de professor** que **permite:**
 - a) **Adicionar (add); (0,25 ponto)**
 - b) **Editar (update); (0,25 ponto)**
 - c) **Listar (display), e; (0,25 ponto)**
 - d) **Remover (delete). (0,25 ponto)**

Também existem **outros métodos** que foram **criados** como:

 - e) **Pesquisar todos os professores (find), e; (0,25 ponto)**
 - f) **Pesquisar por um determinado professor específico (findProfessor).**
(0,25 ponto)
7. Altere o **menu principal** de tal forma que:
 - a) Acesse **primeiro** as seguintes **opções: (0,5 ponto)**
 - **Estudante:** que **redirecione** para o **Menu Estudante**;
 - **Professor:** que **redirecione** para o **Menu Professor**, e;
 - **Sair:** realiza a **saída do programa**.
 - b) **Faça um segundo menu (Menu Estudante)** que **acesse todas as opções e operações do CRUD do Estudante** e com a **possibilidade de retornar/voltar para o menu principal; (0,75 ponto)**
 - c) **Faça um terceiro menu (Menu Professor)** que **acesse todas as opções e operações do CRUD do Professor** e com a **possibilidade de retornar/voltar para o menu principal. (0,75 ponto)**
8. **Por fim, limpe o projeto CRUDListEstudante, compacte o arquivo com o número do seu grupo e poste no MSTeams para ser avaliado pelo professor na data estabelecida para a entrega da atividade. (1 ponto)**