

Atividade valendo **12 presenças**

Problema

Como coordenar os componentes da interface gráfica (swing) de uma tela de cadastro de forma que dependendo das ações dos usuários, os componentes se adaptem a ação e reduza as falhas e/ou perda de operações em execução.

Estudo de caso

Considere a tela do cadastro de Cursos em um sistema de Controle Acadêmico representado abaixo:

Código	Nome	Sigla

Tela 01 – Tela principal

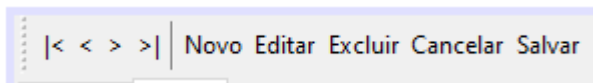
Na barra de botões estão o início das principais ações, mas não exclusivamente. Logo abaixo da barra de ferramentas temos a abas Listagem e Edição. Onde a listagem permite uma consulta rápida e acesso facilitado e arbitrário a qualquer objeto (registro) contido na lista de cursos. Na aba de edição (tela 02) estão os campos de entrada de texto, que possibilita a inclusão e modificação de qualquer objeto da classe Curso. Observe que no presente caso, a tabela permite acesso a todos os atributos, mas isso nem sempre é possível e nestes cenários, a tela de edição permite entrada de dados mais complexas.

Nome: nome do curso

Sigla: TADS Código: 47A

Tela 02 - Edição

Abaixo segue o detalhamento das ações que podem ser iniciadas na barra de ferramentas, essas ações podem ou não estar disponíveis, em função da quantidade de dados que estiverem presentes na coleção ou em função da ação que o usuário estiver executando.



Os quatro primeiros botões atuam diretamente sobre o cursor (ponteiro virtual que indica qual objeto da coleção está em foco/ativo) e permite a movimentação do ponteiro da coleção cursos, os botões são:

Primeiro – desloca/seleciona o primeiro elemento da coleção.

Anterior – desloca o cursor para o objeto anterior ao atual.

Próximo - desloca o cursor para o próximo objeto da lista, subsequente ao atual.

Último - desloca para o último objeto da coleção.

Os cinco botões seguintes permitem as operações de inclusão, edição, exclusão, cancelamento das operações de inserção/edição e de salvar a inclusão/alteração.

Comportamento

Comportamento esperado da interface, de acordo com diversos critérios/condições:

C1 - Estado inicial (coleção vazia): Ao iniciar a aplicação a coleção estará vazia e, portanto, a interface deverá direcionar o usuário direto para o processo de inclusão, apresentando a tela de edição (os campos textfield), habilitando os botões salvar e cancelar e desabilitando os botões de movimentação, novo e editar.

C2 – Estado consulta: (modo padrão para coleção com 1 ou mais objetos). A interface fica no estado inicial, onde a maioria dos botões estão habilitados e os botões de salvar e cancelar ficam inativos.

C3 – O processo de inserção começa com o clique no botão Novo e a interface deverá responder a essa ação, ativando a tela de edição, habilitando os campos de textfield, desabilitando todos os botões, exceto o de salvar/cancelar.

C4 – O processo de edição começa com a seleção de um objeto (tabela ou botões de navegação) e com o clique no botão Editar, que transfere para a tela de edição (se não estiver nela), desabilita todos os botões, exceto o botão salvar e cancelar.

C5 – O processo de salvar: Se estiver em inclusão, a ação de salvar copia os valores da textfield para um novo objeto e insere esse objeto na coleção. Após a inserção o objeto model da tabela deverá disparar a ação de atualização da tabela (veja os métodos fire disponíveis na classe DefaultTableModel ou em AbstractTableModel) e voltar para o modo consulta. Se estiver em modo de edição, a aplicação deverá copiar o conteúdo dos textfield para o objeto em foco (selecionado) e atualizar a tabela.

C6 – O processo de exclusão: Quando a coleção tiver mais de um objeto e não estiver em inserção/edição, a operação de exclusão poderá ser iniciada pelo usuário. Nesse caso a aplicação irá remover da coleção o objeto selecionado e a tabela deverá ser atualizada (ver os métodos fires da AbstractTableModel).

C7 – A ação do cancelar estará disponível quando for iniciada uma inclusão ou edição, essas ações poderão ser canceladas, exceto quando a coleção estiver vazia, que nesse caso a interface deverá ficar no modo de inserção.

C8 – A cada mudança de estado ou do objeto em foco, a linha de status deverá ser atualizada, informado o estado atual (consulta, inserção, edição) e o índice do objeto selecionado e a quantidade de objetos na coleção. Exemplo: Consulta (1/10). Indicando que o primeiro objeto da coleção está selecionado.

C9 - Botões de movimentação: os botões primeiro e anterior devem ficar habilitado quando a coleção tiver mais de um objeto e quando o cursor estiver na posição do segundo elemento ou superior. Em outros casos, deverá ficar inativo.

C10 - Botões de movimentação 2: os botões de próximo e último deverão ficar habilitados quando a coleção tiver mais de um objeto e quando o cursor não estiver na última posição.

O objeto da classe ControleDados associada ao CursoFrame deverá manter e representar o cursor da lista de objetos genéricos e coordenar o comportamento da interface. Mas não diretamente, ou seja, o objeto de ControleDados não irá fazer referência direta ao objeto do CursoFrame, ela fará as notificações via métodos previstos na interface ControleDadosObserver (incompleta no modelo disponível), a classe ControleDados irá manter a lista de objetos que escutam (listeners) e fará as notificações devidas. A classe CursoFrame irá implementar a interface ControleDadosObserver. Dessa forma, se a aplicação tiver mais de um observador do mesmo objeto ControleDados, todos eles serão informados da modificação dos estados.

Como proposta, disponibilizamos o enum type EstadoControleDados que representa os estados possíveis do objeto ControleDados. Segue a declaração do enum EstadoControleDados:

```
public enum EstadoControleDados {  
  
    INATIVO,CARREGANDO,CONSULTA,INCLUSAO,EDICAO,CANCELANDO,SALVANDO;  
  
}
```

A classe ControleDados não deverá ter como dependência os objetos da classe CursoFrame ou Curso.