

Pretende-se com este trabalho que o aluno consolide os resultados da aprendizagem da disciplina de MPD destacando: sistema de tipos Java (com maior ênfase na utilização na reflexão), padrões de desenho *object-oriented* e acesso a dados reais.

Data de entrega incluindo relatório do projecto: 20 de Julho de 2010

Grupo 1

No contexto do 1º grupo da 3ª série de exercícios, acrescente a capacidade de criação, remoção e alteração de objectos de domínio, usando o padrão *Unit of Work*.

Grupo 2

No contexto do 2º grupo da 3ª série de exercícios, substitua a utilização de `PropertiesViewer<T>` por `PropertiesEditor<T>` de forma a permitir actualizações sobre os objectos de domínio.

Repita o teste efectuado na 3ª série, mas fazendo também modificações sobre os objectos de domínio que devem ser reflectidas sobre a base de dados.

Grupo 3

Baseado no desenho da `ListView`, pretende-se agora implementar uma `GridView`, que encapsula um componente `JTable`. O objectivo final é substituir a utilização de `ListView` por `GridView` no teste realizado no 2º grupo deste trabalho.

A classe `GridView` deve receber como parâmetro de construção uma referência para um `TableModel`, que servirá de modelo à `JTable` encapsulada. Pretende-se também que cada linha corresponda a um objecto de domínio.

1. Implemente uma classe `TableModelAdapter` que é um adaptador de `Collection<V>` para `TableModel`.

As colunas do `TableModel` correspondem às propriedades de `V` anotadas com `@VisibleProperty` do tipo `Simple`. Todas as células são não editáveis.

Teste `TableModelAdapter` numa `GridView` que apresenta uma listagem de produtos.

2. Considere a interface `GridViewModel<V>` definida na Listagem 1.

Implemente a classe `GridViewModelAdapter` que é um adaptador de `Collection<V>` para `GridViewModel<V>`.

```
interface GridViewModel<V> extends TableModel{  
    void add(V line);  
    void remove(V line);  
}
```

Listagem 1

Teste `GridViewModelAdapter` numa `GridView` que apresenta uma listagem de produtos e que tem duas acções para adição e remoção de linhas.

3. Implemente uma classe `GridViewEditor` que estende `GridView` e que recebe como parâmetro de construção um `GridViewModel<V>`.

`GridViewEditor` adiciona automaticamente acções para: adição, remoção e actualização dos elementos apresentados (neste último caso através de um `PropertiesEditor`).

4. No teste efectuado no 2º grupo deste trabalho, substitua a utilização de `ListView` por `GridViewEditor`.