



Instituto Superior Técnico

Programação com Objectos

Enunciado do Projecto – 2015/2016

Introdução

O objectivo do projecto é criar uma aplicação que permite ver e alterar um documento textual. A aplicação mantém o nome do documento onde deve ser guardado o documento em disco. A aplicação apenas edita um documento de cada vez, embora possa trocar de documento.

Neste texto, o tipo fixo indica um literal; o símbolo `amsmath` indica um espaço; e o tipo *itálico* indica uma parte variável.

Estrutura de um Documento

Um documento tem uma estrutura interna, correspondente à organização do seu conteúdo, sendo possível alterar este conteúdo.

Os documentos têm um título, um ou mais autores, e um conjunto de parágrafos e secções. A dimensão de um documento é o numero de caracteres presentes no documento, sendo considerados na contagem todos os parágrafos e títulos de todas as secções do documento.

Cada autor tem um nome e um contacto (email).

As secções têm um título, uma sequência de parágrafos e uma sequência de subsecções (ambas potencialmente vazias). Os parágrafos e as subsecções são identificados implicitamente em cada secção pelo seu número de ordem (ambos com início em 0). Cada parágrafo (sequências de frases relacionadas contextualmente) ou secção pode ter um identificador único no contexto do documento (cadeia de caracteres).

É possível adicionar/remover secções e parágrafos a/de documentos e secções. É possível alterar o conteúdo de um parágrafo.

Interacção com o Utilizador: Editor de Documentos

Descreve-se abaixo a **funcionalidade máxima** da interface com o utilizador. Em geral, os comandos pedem toda a informação antes de proceder à sua validação (excepto onde indicado). Todos os menus têm automaticamente a opção **Sair** (fecha o menu).

As operações de leitura e escrita **devem** realizar-se através do objecto **ist.po.ui.Dialog.IO**. As mensagens são produzidas pelos métodos das bibliotecas de suporte (**po-uilib** e **edt-support**). As mensagens não podem ser usadas no núcleo da aplicação. Além disso, não podem ser definidas novas. Potenciais omissões devem ser esclarecidas antes de qualquer implementação.

As excepções usadas na interacção, excepto se indicado, são subclasses de **ist.po.ui.DialogException**, são lançadas pelos comandos e tratadas por **ist.po.ui.Menu**. Outras excepções não devem substituir as fornecidas nos casos descritos.

Note-se que os comandos e menus a seguir descritos, assim como o programa principal, já estão parcialmente implementados nas classes das packages **edt.textui**, **edt.textui.main** e **edt.textui.section**. Estas classes são de uso obrigatório e estão disponíveis no CVS (módulo **edt-textui**).

Menu Principal

As acções do menu, listadas em **edt.textui.main.MenuEntry**, permitem gerir a salvaguarda do estado da aplicação, assim como operar sobre o documento actual: [Criar](#), [Abrir](#), [Guardar](#), [Listar meta-informação](#), [Adicionar autor](#), [Listar secções](#), [Mostrar elemento de texto](#) e [Editar](#). A classe **edt.textui.main.Message** define os métodos para geração das mensagens de diálogo. Inicialmente, a aplicação tem um documento vazio e anónimo.

Salvaguarda do Documento Actual

O conteúdo do documento pode ser guardado para posterior recuperação (via serialização Java: **java.io.Serializable**). Na leitura e escrita do estado da aplicação, devem ser tratadas as excepções associadas. A funcionalidade é a seguinte:

- **Criar** -- Cria um novo documento anónimo não associado a nenhum ficheiro.
- **Abrir** -- Carrega um documento anteriormente salvaguardado, ficando o documento carregado associado ao ficheiro nomeado: pede-se o nome do ficheiro a abrir (**openFile()**). Caso o ficheiro não exista, é apresentada a mensagem **fileNotFound()**.
- **Guardar** -- Guarda o documento actual no ficheiro associado. Se não existir associação, pede-se o nome do ficheiro a utilizar, ficando a ele associada. Esta interacção realiza-se através do método **newSaveAs()**. Não é executada nenhuma acção se não existirem alterações desde a última salvaguarda.

Apenas existe um documento no editor (explicitamente criado ou aberto). A opção **Sair** nunca guarda o estado da aplicação, mesmo que existam alterações.

Estes três comandos já estão parcialmente implementados nas classes da *package* **edt.textui.main** (disponível no CVS): **New**, **Open**, **Save**.

Listar meta-informação

É apresentada a seguinte informação: título do documento, autores do documento (ordem alfabética), número de secções de topo do documento, dimensão do documento (bytes) e número de identificadores únicos (das secções e parágrafos do documento). Cada linha é gerada por uma mensagem de **edt.textui.main.Message**.

```
Título: título do documento
Autor: nome do autor 1/email do autor 1
...
Autor: nome do autor N/email do autor N
Secções de topo: número de secções de topo
Dimensão do documento (bytes): dimensão do documento (soma da dimensão dos títulos e dos parágrafos)
Identificadores únicos: número de identificadores
```

Este comando já está parcialmente implementado na classe da *package* **edt.textui.main** (disponível no CVS): **ShowMetadata**.

Adicionar autor

É pedido o nome (**requestAuthorName()**) e o email (**requestEmail()**) do novo autor. Se o nome já estiver associado ao documento, apresenta a mensagem **duplicateAuthor()** e não executa nenhuma acção.

Este comando já está parcialmente implementado na classe da *package* **edt.textui.main** (disponível no CVS): **AddAuthor**.

Listar secções de topo

É apresentada a lista de títulos de todas as secções de topo do documento (um título por linha): cada título entre chavetas é precedido pelo seu identificador único entre parênteses rectos (ou só os parênteses, caso o identificador não esteja definido). O título do documento deve ser apresentado antes da lista de secções, tal como os títulos das secções (entre chavetas). Se algum título for vazio, apresentam-se apenas as chavetas. Note-se que apesar de o documento tem uma estrutura semelhante à das secções, não possui identificador, pelo que o seu título não é precedido de nenhuma marca identificação (nem sequer parênteses rectos vazios).

Exemplos:

```
{Título do documento}  
[x123] {Isto é um título}  
[] {Isto é outro título}  
[w83] {}  
[] {}
```

Este comando já está parcialmente implementado na classe da *package* **edt.textui.main** (disponível no CVS): **ShowIndex**.

Mostrar elemento de texto

Permite apresentar o conteúdo do elemento de texto indicado pelo utilizador (através da especificação do identificador único **requestElementId()**). Se o elemento de texto não existir, é apresentada a mensagem **noSuchTextElement()** e a acção termina. Caso contrário,

é apresentado o conteúdo. Se o elemento de texto for um parágrafo, apresenta-se simplesmente o seu texto. Se for uma secção, apresenta-se tal como especificado em [mostrar conteúdo](#).

Este comando já está parcialmente implementado na classe da *package* **edt.textui.main** (disponível no CVS): **ShowTextElement**.

Editar

Abre o menu de edição (alteração) do documento e do seu conteúdo, estabelecendo como secção actual o nível de topo do documento.

Este comando já está parcialmente implementado na classe da *package* **edt.textui.main** (disponível no CVS): **Edit**.

Menu de Edição

Este menu permite efectuar operações sobre um documento. A lista completa é a seguinte: [Alterar título](#), [Listar secções](#), [Mostrar conteúdo](#), [Ir para secção](#), [Inserir secção](#), [Nomear secção](#), [Remover secção](#), [Inserir parágrafo](#), [Nomear parágrafo](#), [Alterar parágrafo](#), [Remover parágrafo](#).

As etiquetas das opções deste menu estão definidas na classe **edt.textui.section.MenuEntry**. Todos os métodos correspondentes às mensagens de diálogo para este menu estão definidos na classe **edt.textui.section.Message**.

Em todos os casos, só podem ser indicados secções e parágrafos filhos da secção actual. O documento é estruturalmente semelhante a uma secção, sendo considerado a secção raiz (na primeira abertura deste menu).

Alterar título

Permite alterar o título da secção actual (ou do próprio documento). Para tal, pede-se o novo título (**requestSectionTitle()**).

Este comando já está parcialmente implementado na classe da *package* **edt.textui.section** (disponível no CVS): **ChangeTitle**.

Listar Secções

É apresentada a lista de títulos de todas as subsecções da secção actual (recursivamente): cada título entre chavetas é precedido pelo seu identificador único entre parênteses rectos (ou só os parênteses, caso o identificador não esteja definido). Se algum título for vazio, apresentam-se apenas as chavetas.

Exemplos:

```
[x123] {Isto é um título}  
[] {Isto é outro título}  
[w83] {}  
[] {}
```

Este comando já está parcialmente implementado na classe da *package* **edt.textui.section** (disponível no CVS): **ListSections**.

Mostrar conteúdo

Apresenta o título e todo o conteúdo da secção actual: o texto de cada parágrafo é apresentado por ordem; cada secção é apresentada tal como no caso da apresentação do índice, seguida dos seus parágrafos e subsecções (apresentação recursiva).

Este comando já está parcialmente implementado na classe da *package* **edt.textui.section** (disponível no CVS): **ShowContent**.

Ir para secção

Permite seleccionar a secção indicada pelo utilizador (através da especificação do identificador local **requestSectionId()**). Se a secção indicada não existir, continua na secção actual, sendo apresentada a mensagem **noSuchSection()**. Caso contrário, é apresentada a mensagem **newActiveSection()** e o menu de edição de secções é aberto para a secção seleccionada.

Este comando já está parcialmente implementado na classe da *package* **edt.textui.section** (disponível no CVS): **SelectSection**.

Inserir secção

Permite inserir uma subsecção antes de outra de referência (indicada pelo identificador local **requestSectionId()**). É pedido o título (**requestSectionTitle()**) da nova secção. Se a subsecção de referência não existir, insere a nova subsecção no final da secção actual.

Este comando já está parcialmente implementado na classe da *package* **edt.textui.section** (disponível no CVS): **InsertSection**.

Nomear secção

Permite atribuir um identificador único (no contexto do documento) a uma secção. É pedido o identificador local (inteiro) da subsecção a identificar (**requestSectionId()**) e o identificador único a atribuir (**requestUniqueId()**) (cadeia de caracteres). Se a secção não existir é apresentada a mensagem **noSuchSection()** e a acção termina. Se o identificador já existir, passa a designar a nova secção (a ligação anterior é destruída). Se a secção já tiver um identificador, a mensagem de aviso **sectionNameChanged()** é apresentada e é atribuído o novo identificador único à secção.

Este comando já está parcialmente implementado na classe da *package* **edt.textui.section** (disponível no CVS): **NameSection**.

Remover Secção

Permite remover a secção indicada pelo utilizador através do identificador local (**requestSectionId()**). Se a secção indicada não existir, apresenta a mensagem **noSuchSection()** e não realiza nenhuma acção.

Este comando já está parcialmente implementado na classe da *package* **edt.textui.section** (disponível no CVS): **RemoveSection**.

Inserir parágrafo

Permite inserir um novo parágrafo antes de outro parágrafo da secção actual. É pedido o número do parágrafo de referência (**requestParagraphId()**) e o texto do novo parágrafo (**requestParagraphContent()**). Se o parágrafo de referência não existir, insere o novo parágrafo no final da sequência de parágrafos da secção.

Este comando já está parcialmente implementado na classe da *package* **edt.textui.section** (disponível no CVS): **InsertParagraph**.

Nomear parágrafo

Permite atribuir um identificador único (no contexto do documento) a um parágrafo. É pedido o número do parágrafo a identificar (**requestParagraphId()**) e o identificador único a atribuir (**requestUniqueId()**) (cadeia de caracteres). Se o parágrafo não existir é apresentada a mensagem **noSuchParagraph()** e a acção termina. Se o identificador já existir, passa a designar o novo parágrafo (a ligação anterior é destruída). Se o parágrafo já tiver um identificador, a mensagem de aviso **paragraphNameChanged()** é apresentada e é atribuído o novo identificador único ao parágrafo.

Este comando já está parcialmente implementado na classe da *package* **edt.textui.section** (disponível no CVS): **NameParagraph**.

Alterar parágrafo

Permite alterar o texto do parágrafo indicado. É pedido o número do parágrafo a alterar (**requestParagraphId()**) e o texto do novo parágrafo (**requestParagraphContent()**). Se o parágrafo indicado não existir, apresenta a mensagem **noSuchParagraph()** e não realiza nenhuma acção.

Este comando já está parcialmente implementado na classe da *package* **edt.textui.section** (disponível no CVS): **EditParagraph**.

Remover parágrafo

Permite remover o parágrafo indicado pelo utilizador (**requestParagraphId()**). Se o parágrafo indicado não existir, apresenta a mensagem **noSuchParagraph()** e não realiza nenhuma acção.

Este comando já está parcialmente implementado na classe da *package* **edt.textui.section** (disponível no CVS): **RemoveParagraph**.

Leitura de um Documento a Partir de Ficheiro

Além das opções de manipulação de ficheiros descritas no [menu principal](#), é possível iniciar a aplicação com um ficheiro de texto especificado pela propriedade Java **import** (apresentada abaixo; este exemplo está no ficheiro **test.import**). Estes dados são apenas uma forma cómoda de inicialização e **nunca** são produzidos pela aplicação (nem mesmo para salvarguardar o estado para execuções futuras). Quando se especifica a propriedade, o sistema de ficheiros é povoado com as entidades do ficheiro indicado (uma por linha).

Nestes ficheiros, as secções são sempre filhas do documento e os parágrafos pertencem à secção criada mais recentemente (ou ao documento, se não tiverem sido ainda criadas secções).

Assume-se que não há entradas mal-formadas (nestes ficheiros, as secções têm sempre identificador; os parágrafos não são identificados). Sugere-se a utilização do método **String.split**, para dividir uma cadeia de caracteres em campos.

```
Objects in Space  
Obi-Wan Kenobi|obl@tatooine.geocities.com|Master Yoda/yoda@dagobah.net|Luke Skywalker/luke@jedi.org  
SECTION|lfd|A Walk in the Desert  
PARAGRAPH|These aren't the droids you're looking for.  
SECTION|wm|Patience You Must Have  
PARAGRAPH|Do or do not.  
PARAGRAPH|There is no try.  
SECTION|dvls|Drama in the Clouds  
PARAGRAPH|Obi-Wan never told you what happened to your father.  
PARAGRAPH|He told me enough! He told me YOU killed him.  
PARAGRAPH|N00000000000000000000...
```

O editor nunca produz ficheiros neste formato, nem os pode ler a partir de nenhum menu.

Considerações sobre Flexibilidade e Eficiência

Devem ser possíveis extensões ou alterações de funcionalidade com impacto mínimo no código já produzido para o editor de texto. O objectivo é aumentar a flexibilidade da aplicação relativamente ao suporte de novas funções.

Em particular, a solução encontrada para visualizar o conteúdo do documento deve ser suficientemente flexível por forma a que se possa visualizar o conteúdo de um documento

noutro formato (por exemplo, XML) sem que isso implique alterações no código core da aplicação.

Execução dos Programas e Testes Automáticos

Usando os ficheiros `test.import`, `test.in` e `test.out`, é possível verificar automaticamente o resultado correcto do programa. Note-se que é necessária a definição apropriada da variável **CLASSPATH** (ou da opção equivalente **-cp** do comando **java**), para localizar as classes do programa, incluindo a que contém o método correspondente ao ponto de entrada da aplicação (`edt.textui.TextEditor.main`). As propriedades são tratadas automaticamente pelo código de apoio.

```
java -Dimport=test.import -Din=test.in -Dout=test.outhyp edt.textui.TextEditor
```

Assumindo que aqueles ficheiros estão no directório onde é dado o comando de execução, o programa produz o ficheiro de saída `test.outhyp`. Em caso de sucesso, os ficheiros das saídas esperada (`test.out`) e obtida (`test.outhyp`) devem ser iguais. A comparação pode ser feita com o comando:

```
diff -b test.out test.outhyp
```

Este comando não deve produzir qualquer resultado quando os ficheiros são iguais. Note-se, contudo, que este teste não garante o correcto funcionamento do código desenvolvido, apenas verificando alguns aspectos da sua funcionalidade.