

Laboratórios de Informática I

2020/2021

Mestrado Integrado em Engenharia Informática



Universidade do Minho

Sessão Laboratorial – Documentação



Introdução

- *Comentários* são uma construção da linguagem que permite aos programadores incluírem texto livre no meio do programa
- Naturalmente que esse texto não influirá no comportamento do programa
- Ou seja, vai acabar por ser ignorado pelo compilador ou interpretador
- Mas pode ser muito útil para facilitar a compreensão do programa por parte de quem lê o código

Em *Haskell*

Em *Haskell* estão previstos dois tipos de comentário:

- bloco de texto envolvido entre as marcas `{-` e `-}`. Os blocos de comentários podem abranger múltiplas linhas;
- desde os caracteres `--` até ao final de linha.

Por exemplo,

```
{- "fact n" calcula o factorial de um número inteiro "n".  
   obs: assume-se que o argumento "n" é não negativo.  
-}  
  
fact :: Integer -> Integer  
fact 0 = 1                -- caso de base da recursividade  
fact n = n * fact (n-1)   -- caso indutivo
```

Boas Práticas

O texto inserido nos comentários é livre, mas existe um conjunto de boas práticas:

- um comentário não deve “repetir” por palavras o que o programa faz
- pode no entanto explicitar a *intenção* do fragmento de código respetivo;
- podem ser usados para explicitar ou explicar as estratégias/algoritmos utilizados na resolução de um problema;
- durante a fase de desenvolvimento, é habitual utilizar os comentários para inserir:
 - descrição sobre funcionalidades que ainda terão de ser implementadas;
 - marcas como `FIXME` ou `TODD` que assinalam pontos que devem merecer ainda atenção por parte do(s) programador(es);
 - para excluir do programa código que se destine a teste/*debug*.

Comentários Estruturados

- LPs modernas usam os comentários como um mecanismo para adicionar informação ao código desenvolvido por forma a permitir a extração automática de documentação;
- O texto dos comentários inclui uma estrutura que possibilita que ferramentas os processem e extraiam de lá a informação necessária para a produção da documentação;
- Ajuda a que a documentação se encontre sempre atualizada em relação à versão do código considerada;
- Permite que o sistema de documentação possa aceder a informação contida no próprio código (e.g. tipos das funções);
- Um exemplo é a própria documentação das bibliotecas do *Haskell* (acessível em <http://www.haskell.org/ghc/docs/latest/html/libraries/index.html>).

Haddock

- haddock é uma ferramenta de extração de documentação para programas *Haskell*;
- Destina-se a processar programas *Haskell* com anotações apropriadas inseridas nos comentários;
- Produz como resultado páginas de documentação em diferentes formatos (e.g. HTML, para se visualizar em *browsers web*);
- Mais detalhes sobre haddock em <http://www.haskell.org/haddock/doc/html/>.

Funcionalidades

- Um comentário iniciado por “{- |” (para blocos), ou “-- |” (para linhas) será interpretado pelo haddock como um comentário referente à declaração que se segue (e.g. funções; tipos; construtores; etc.);
 - Exemplo bloco:

```
{- | A função 'fact' calcula o factorial
    (@fact n@ retorna o factorial
    de de um inteiro @n@).
-}
```
 - Exemplo linhas:

```
-- | A função 'fact' calcula o factorial
--   (@fact n@ retorna o factorial
--   de de um inteiro @n@).
```

Funcionalidades

- Um comentário iniciado por “- ^” diz respeito ao item imediatamente anterior (e.g. tipo envolvido numa assinatura);
 - Exemplo:

```
fact :: Int -- ^ argumento 'n' (assume-se não negativo)
      -> Int -- ^ resultado
```


Formatação de Texto

- Podem-se utilizar marcas para formatação do texto, como:
 - Parágrafos são separados por linhas em branco;
 - `/text/`: imprime text em *itálico*;
 - `__text__`: imprime text em **bold**;
 - `@text@`: imprime text em mono-espacamento;
 - `'name'`: insere hiper-ligação para entidade name;
 - `* item` (no início de um parágrafo): item de uma lista (não enumerada);
 - `1. item` (no início de um parágrafo): item de uma lista (enumeração);
 - `= heading`, `== subheading`, `=== subsubheading`: início de secção, sub-secção; etc.

Invocação do haddock

```
haddock -h -o doc/html Main.hs
```

- -h indica que se pretende que a documentação seja produzida no formato HTML;
- -o doc/html indica que os ficheiros resultantes devem ser gravados na (sub-)diretoria doc/html;
- a documentação ficará assim acessível a partir da página doc/html/index.html.