Capítulo 2

Começando com Haskell e GHCi

2.1 Primeiro programa Haskell

2.2 Usando Haskell na prática

2.3 Usando GHCi

2.4 O prelúdio¹ padrão e as bibliotecas Haskell

2.5 Módulos

2.6 Um segundo exemplo: imagens

2.7 Erros e mensagens de erro

O capítulo 1 introduziu os fundamentos funcionais da programação em Haskell. Nós, agora, estamos prontos para usar o GHCi para fazer a programação prática.

No começo da programação nós vamos aprender também o básico de Módulos de Sistema Haskell, sobre o qual os programas podem ser escritos múltiplos, independentes arquivos, e podem ser usar o ‘built-in’ funções no prelúdio¹ e bibliotecas.

Nossos exemplos de programação vão focar em usar o Picture example introduzido no Capítulo 1 como também alguns exemplos numerais simples. Com a ajuda dessas informações nos vamos olhar em como conseguir os programas e outros materiais importantes para o livro, como também em obter e instalar GHCi como uma parte da Plataforma Haskell.

Nos concluiremos com breves observações sobre os tipos de mensagens de erro que podem ser gerados a partir de algo escrito incorretamente dento do GHCi.

2.1 Primeiro programa Haskell

Nos começaremos a seção observando o primeiro programa Haskell ou seu script, onde consiste em inúmeros exemplos do Capítulo 1. Este é o arquivo FirstScript.hs, mostrado na Figura 2.1. Códigos Haskell são salvos em arquivos com extensões ‘.hs’.

Como também as definições, um script conterá comentários. Um comentário em um script é a parte da informação que mais vale para o usuário do que para a máquina. Nele deve conter uma explicação informal sobre como funciona, como pode ou não ser usado, de forma geral a interface da livraria e assim por diante. Tudo no arquivo do programa pode ser interpretado como um texto, exceto onde está explicitamente indicado que determinada linha é um comentário.

Comentários são indicados de duas formas. O símbolo ‘--’ começa o comentário onde ocupa parte da linha a direita do símbolo. Comentários também podem ser encapsulados por símbolos ‘{-‘ e ‘-}’. Estes comentários são arbitrários, ou seja, eles podem ocupar mais de uma linha, como também incorporar outros comentários; eles são, portanto, chamados de **nested comments²**.