

Definindo variáveis e funções

- Editores de texto
- Programas fonte
- Regra do layout
- Comandos úteis do GHCi
- Atividades

Editores de texto

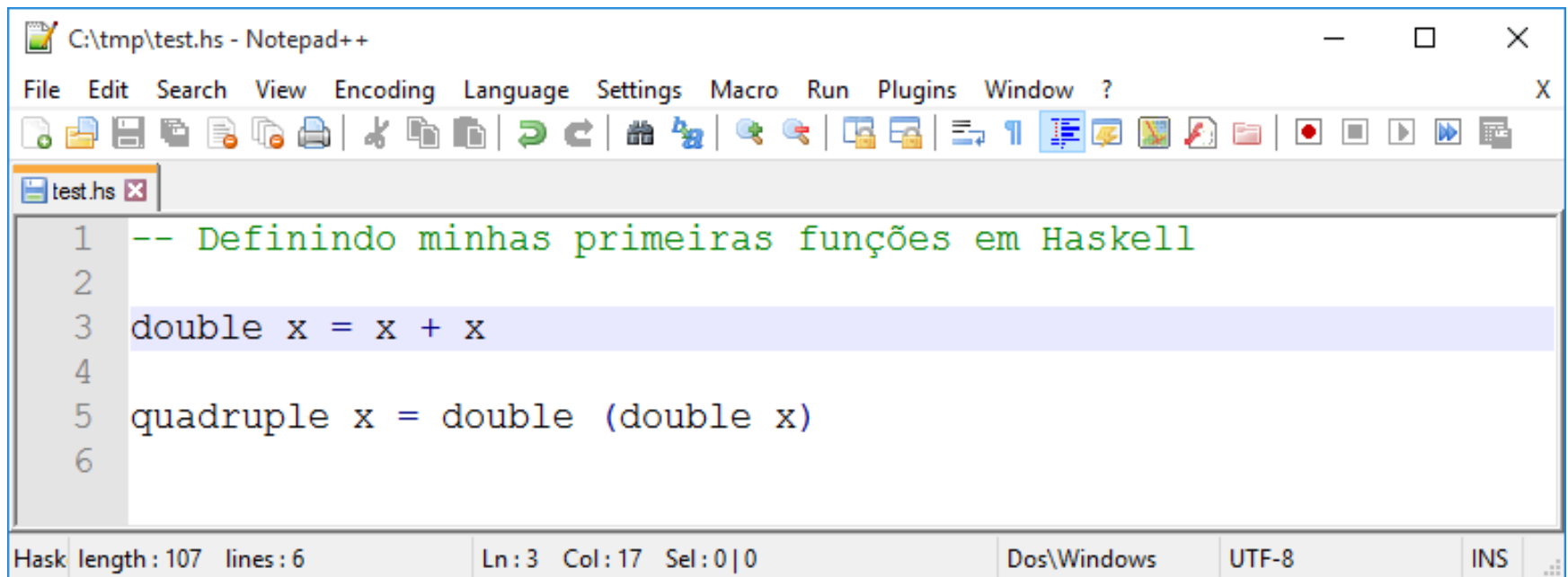
- Ao desenvolver um programa pode ser útil manter duas janelas abertas,
- uma executando um editor de texto para editar o código,
- e outra para o ambiente interativo (GHCi) em execução.

Editores de texto

- Algumas sugestões de editores de texto que poderão ser usados para escrever seus programas são:
- Notepad++ para Windows
- Gedit para Linux

Notepad++ para Windows

- Um editor melhor do que o Bloco de Notas é o Notepad++ (<http://notepad-plus-plus.org/>).

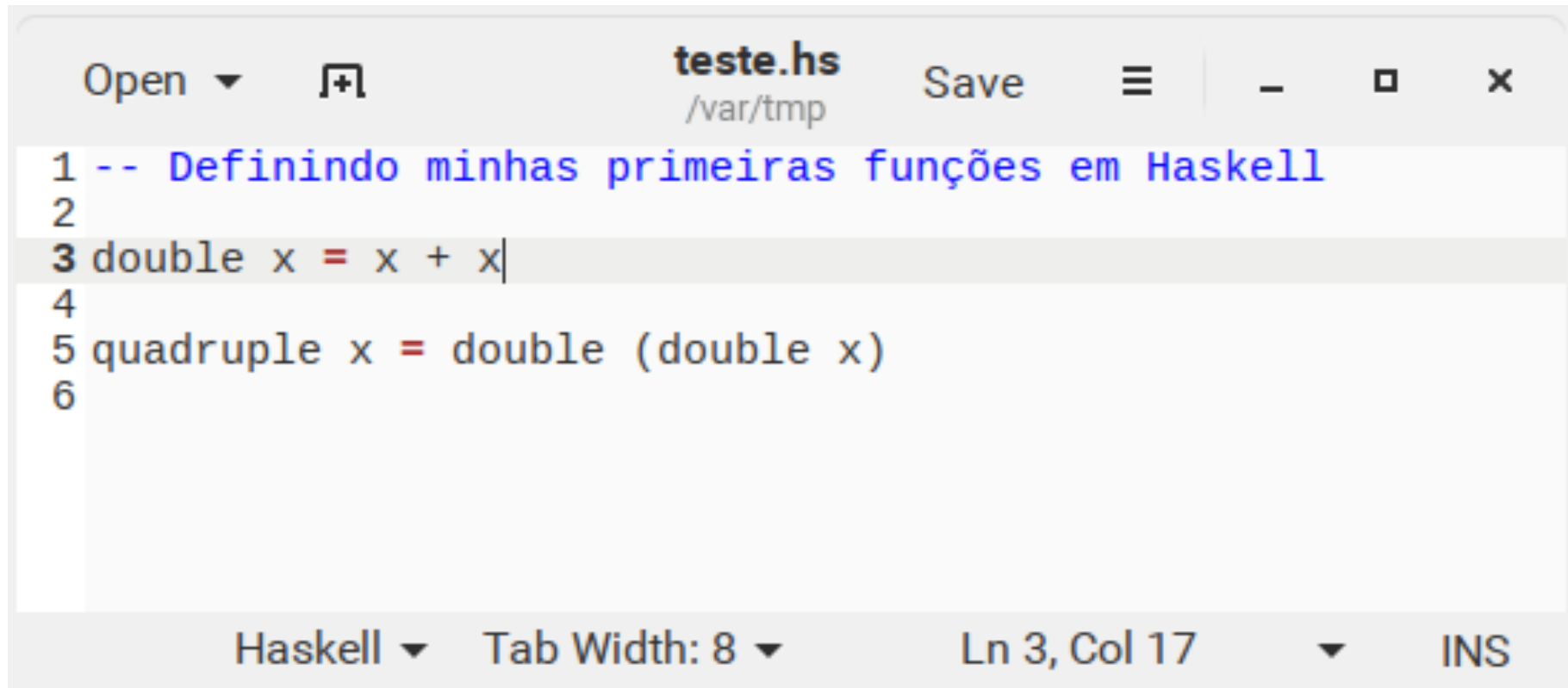


The screenshot shows the Notepad++ application window. The title bar reads "C:\tmp\test.hs - Notepad++". The menu bar includes File, Edit, Search, View, Encoding, Language, Settings, Macro, Run, Plugins, Window, and ?. The toolbar contains various icons for file operations and editing. The editor area shows a file named "test.hs" with the following content:

```
1  -- Definindo minhas primeiras funções em Haskell
2
3  double x = x + x
4
5  quadruple x = double (double x)
6
```

The status bar at the bottom displays "Hask", "length: 107", "lines: 6", "Ln: 3", "Col: 17", "Sel: 0 | 0", "Dos\Windows", "UTF-8", and "INS".

Gedit para Linux



The image shows a screenshot of the Gedit text editor interface. The title bar at the top indicates the file is named `teste.hs` and is located at `/var/tmp`. The menu bar includes 'Open', 'Save', and a hamburger menu icon. The main editing area contains the following Haskell code:

```
1 -- Definindo minhas primeiras funções em Haskell
2
3 double x = x + x|
4
5 quadruple x = double (double x)
6
```

The status bar at the bottom shows the current language is 'Haskell', the tab width is 8, the cursor is at line 3, column 17, and the input mode is 'INS'.

Primeiro programa fonte

- Primeiro você tem que escolher um editor de texto (editor de programa)
- Depois tem que instalar em seu computador
- Se você ainda não tem um, recomendo o Notepad++ para Windows
- Pode ser encontrado neste site:
- <http://notepad-plus-plus.org/>

Primeiro programa fonte

- Inicie um editor de texto, digite as seguintes definições de função:
- $\text{dobro } x = x + x$
- $\text{quadruplo } x = \text{dobro } (\text{dobro } x)$
- e salve o arquivo com o nome teste.hs

Primeiro programa fonte

- Deixando o editor aberto, em outra janela execute o GHCi carregando o novo arquivo fonte:
- ```
> ghci teste.hs
```

GHCi, version 7.10.3: <http://www.haskell.org/ghc/> :? for help [1 of 1] Compiling Main  
( teste.hs, interpreted ) Ok, modules loaded:  
Main.
- ```
*Main>
```


Primeiro programa fonte

- Agora, tanto Prelude.hs como teste.hs são carregados, e as funções de ambos os arquivos fonte podem ser usadas:
- *Main> quadruplo 10
- 40
- *Main> 5*(dobro 2) - 3
- 17
- Observe que o GHCi usa o nome de módulo Main se o arquivo fonte não define um nome para o módulo.

Modificando o primeiro programa fonte

- Deixando o GHCi aberto, volte para o editor, adicione as seguintes definições ao arquivo fonte teste.hs, e salve-o.
- `areaCirculo r = pi * r^2`
- `*Main> areaCirculo 5`
- O que aconteceu?

Modificando o primeiro programa fonte

- O GHCi não detecta automaticamente que o arquivo fonte foi alterado.
- Assim o comando `:reload` deve ser executado para que as novas definições possam ser usadas:
- `*Main> :reload`
`[1 of 1] Compiling Main Ok, modules loaded: Main.`
- `*Main> areaCirculo 5`
- `78.53981633974483`

Regra de layout

- Em uma seqüência de definições, cada definição deve começar precisamente na mesma coluna:

a = 10

b = 20

c = 30

Regra de layout

- Se uma definição for escrita em mais de uma linha, as linhas subsequentes à primeira devem começar em uma coluna mais à direita da coluna que começa a sequência de definições.

```
a = 10 + 20 + 30 +  
    40 + 50 + 60 +  
    70 + 80  
b = sum [10, 20, 30]
```

Regra de layout

- A regra de layout evita a necessidade de uma sintaxe explícita para indicar o agrupamento de definições usando `{, }` e `;`.

-- agrupamento implícito

```
a = b + c
  where
    b = 1
    c = 2

d = a * 2
```

-- agrupamento explícito

```
a = b + c
  where { b = 1 ; c = 2 }

d = a * 2
```

Regra de layout

- Para evitar problemas com a regra de layout, é recomendado não utilizar caracteres de tabulação para indentação do código fonte,
- uma vez que um único caracter de tabulação pode ser apresentado na tela como vários espaços.
- O texto do programa vai aparentar estar alinhado na tela do computador, mas na verdade pode não estar devido ao uso do tabulador.

Comandos úteis do GHCi

comando	abrev	significado
:load <i>name</i>	:l	carrega o programa fonte <i>name</i>
:reload	:r	recarrega o programa fonte atual
:edit <i>name</i>	:e	edita o programa fonte <i>name</i>
:edit	:e	edita o programa fonte atual
:type <i>expr</i>	:t	mostra o tipo de <i>expr</i>
:info <i>name</i>	:i	dá informações sobre <i>name</i>
:browse <i>Name</i>		dá informações sobre o módulo <i>Name</i> , se ele estiver carregado
let <i>id</i> = <i>exp</i>		associa a variável <i>id</i> ao valor da expressão <i>exp</i>
:! <i>comando</i>		executa <i>comando</i> do sistema
:help	:h, :?	lista completa dos comandos do GHCi
:quit	:q	termina o GHCi