

Uso de módulos

Cuando necesitamos compartir la definición de tipos o de funciones para no reescribirlas cada vez, o queremos obtener un código ordenado, utilizamos la definición de módulos.

Supongamos que tenemos un archivo denominado Parte1.hs, otro Parte2.hs y otro Principal.hs
Los nombres de los archivos pueden estar en mayúscula o minúscula.

Queremos que las definiciones del archivo Parte1.hs estén disponibles cuando se levanta al ambiente de Hugs el archivo Parte2.hs y las de ambas cuando se levanta el archivo Principal.hs

La primer línea del archivo Parte1.hs será:
module Parte1 where
--siguen las declaraciones de tipos, funciones, etc

Aquí sí es importante que el nombre del módulo, que coincide con el del archivo .hs que lo contiene se escriba en mayúscula y sin la extensión .hs

Las primeras dos líneas de Parte2.hs serán:
module Parte2 where
import Parte1
--siguen las declaraciones de tipos, funciones, etc

Las primeras líneas del archivo Principal.hs serán:
module Principal where
import Parte1
import Parte2
--siguen las declaraciones de tipos, funciones, etc o no, puede tratarse de un módulo que sólo importe declaraciones de otros como módulo principal

A partir de esto, cuando se hace open desde el intérprete de Hugs al archivo Principal.hs o :load a este archivo se cargarán en el ambiente de Hugs todas las definiciones de los tres archivos.

La modularización es positiva ya que permite tener unidades de programas más pequeñas y más fáciles de comprender y manejar. Cada módulo puede resolver un problema en particular, entonces se hace re utilizable en otros módulos dónde no se tendrá la necesidad de re definir. Facilita la detección de errores.