

Tipos simples predefinidos en Haskell - Continuación

El tipo Int

Los valores de este tipo son números enteros de precisión limitada. El intervalo de representación de Int depende de la arquitectura de la máquina y de la implementación de Hugs.

Para determinar este intervalo se pueden usar las expresiones minBound y maxBound que también son de tipo entero.

Ejecute en el intérprete de Hugs la expresión minBound y luego maxBound y determine el rango de Int.

El significado del operador :: en Haskell es "tiene tipo" o "es de tipo". Este operador se usa en Haskell para indicar el tipo de cualquier expresión. (Toda expresión en Haskell tiene tipo)

De forma que:

- ◆ `4 :: Int`, significa que 4 es de tipo Int
- ◆ `(8<4) :: Bool`, significa que `(8<4)` es de tipo Bool
- ◆ `&& :: Bool → Bool → Bool`, significa que `&&` es de tip Bool en Bool en Bool

Funciones y operadores del tipo Int

Algunas de las funciones y operadores predefinidos para este tipo son:

- ◆ `(+), (-), (*) :: Int -> Int -> Int`. Suma, resta y producto de enteros.
- ◆ `(^) :: Int -> Int -> Int`. Operador potencia, el exponente debe ser natural
- ◆ `div, mod :: Int -> Int -> Int`. Cociente y resto de la división entera
- ◆ `abs :: Int -> Int`. Valor absoluto
- ◆ `even, odd :: Int -> Bool`. Dado un entero indican si es par o impar

El tipo Integer

Los valores de este tipo son números enteros de precisión ilimitada. Se usan cuando los valores a tratar se salen del rango del tipo Int, pero los cálculos realizados con Integer son menos eficientes que con Int (más lentos).

Los operadores y funciones disponibles para Int también lo están para Integer.

Diferencia entre Int e Integer en Haskell

- Int es el tipo (conjunto de pertenencia) de los enteros que la computadora puede representar, esto depende de la arquitectura interna de cada máquina.
- Integer representa al conjunto de los enteros matemáticos.

Realizar los siguientes cálculos en Hugs:

1. Calcular 2^2
2. Calcular 2^4
3. Calcular 2^{32}
4. Calcular 2^{64}
5. Calcular 2^{200}
6. Calcular $2^4::\text{Int}$
7. Calcular $2^{32}::\text{Int}$
8. Calcular $2^{32}::\text{Integer}$
9. Calcular $2^{31}::\text{Int}$
10. Calcular $2^{31} - 1::\text{Int}$
11. Calcular $\text{minBound}::\text{Int}$ y $\text{maxBound}::\text{Int}$ y comparar con los resultados de 9 y 10

Analizar los resultados

El tipo Char

Un valor de tipo Char representa un caracter, es decir: una letra, un dígito, un signo de puntuación, etc. Un valor constante de tipo Char se escribe siempre entre comillas simples.

- 'a' denota la letra a minúscula
- '1' denota al carácter correspondiente al dígito 1. No confundir con 1 de tipo Int.
- '?' denota al símbolo correspondiente al signo de interrogación

El tipo String

El tipo String es el tipo de las cadenas de caracteres. Un elemento de String puede ser una cadena de cero o más caracteres.

Un texto entre comillas dobles es un elemento del tipo String.

Luego veremos otra notación para los elementos de String, ya que en realidad son Secuencias o Listas de caracteres, que son tipos que veremos más adelante.

- "Hola todos" es un elemento de tipo String
- "20 autos estacionados en 5 filas" es un elemento de tipo String
- "" es un elemento de tipo String, y en particular es el String vacío
- " " es un elemento de String, y es la cadena que tiene un único carácter que es el espacio en blanco

El comando de Hugs `:type <expr>` nos devuelve el tipo de una expresión, es útil para saber si el tipo de una expresión que queremos usar se corresponde con lo que pensamos.

Verificar en Haskell:

1. `:type 'a'`
2. `:type "abcdf"`
3. `:type True`
4. `:type 4<5`
5. `:type "a"`
6. `length "perro"`
7. Deducir el tipo de `length`
8. Verificar `:type length`
9. Escribir y probar en Hugs sus propias expresiones