Haskell. Typy danych.

Typem nazywamy zbiór wartości (obiektów) tego samego rodzaju. Na przykład liczb, słów, rekordów i bardziej złożonych struktur.

Typy podstawowe

Typami podstawowymi nazywamy typy predefiniowane, które nie są zbudowane z innych typów.

Typy całkowite
• **Int** (fixed-precision) - liczby całkowite z zakresu [-2^31 .. 2^31-1]
• **Integer** (arbitrary- precision) - dowolna liczba całkowita (zarówno ujemna, jak i dodatnia).

Typy rzeczywiste
• **Float** - liczba zmiennoprzecinkowa pojedynczej precyzji,
• **Double** - liczba zmiennoprzecinkowa podwójnej precyzji.

Typ znakowy
• **Char** - typ pojedynczego znaku. Jest to typ wyliczeniowy, którego wartości reprezentują znaki Unicode. W Haskellu pojedyncze znaki (Char) umieszczamy w apostrofach, np. 'a' 'B' '$', itp.

Typ boolowski
• **Bool** - służy do reprezentowania zmiennych logicznych. Jest to typ wyliczeniowy posiadający dwie wartości : False (fałsz - 0) i True (prawda - 1)

Typ relacji między elementami
• **Ordering** - jest to typ wyliczeniowy posiadający trzy wartości:
- LT (less then - mniejszy niż)
- EQ (equal - równy)
- GT (greater then - większy niż)

Wartości tego typu są zwracane między innymi przez funkcję **`compare`** porównującą dwa elementy w standardowym porządku termów.

Typy strukturalne

• **Listy**.
Lista jest strukturą homogeniczną. Rozmiar listy nie jest określony - można dołączać do niej kolejne elementy.

• **Krotki**.
Krotka jest strukturą heterogeniczną. Rozmiar krotki jest ściśle określony w chwili jej tworzenia. Nie jest możliwe dołączenie elementów do istniejącej krotki.

Typy funkcji.

Na typ funkcji składają się typy przyjmowanych przez nią parametrów oraz typ wartości zwracanej przez funkcję. Typy te przedstawiamy następująco:

nazwa_funkcji :: TypParam_1 -> TypParam_2 -> ... -> TypParam_n -> TypWartości

Synonimy typów

Synonimy typów umożliwiają nadanie własnej nazwy dla dowolnego typu. Wykorzystanie synonimów typów jest możliwe tylko w zewnętrznym pliku (skrypcie).

Przykładem zastosowania synonimów typów jest **String**, czyli synonim listy znaków **[Char].**

Typy polimorficzne .

Typ polimorficzny oznacza rodzinę typów.
Na przykład, [a] jest rodziną typów zawierającą listy dowolnych typów.

Identyfikatory, takie jak zastosowana powyżej litera a nazywamy zmiennymi wartości i w przeciwieństwie do nazw typów są pisane małą literą.

Klasy typów

Dla typów polimorficznych często zachodzi potrzeba ograniczenia klasy typów w zależności od kontekstu zastosowań.

Predefiniowane klasy typów:

Num - klasa typów liczbowych
Eq - klasa typów, dla których zdefiniowane jest porównywanie (operatory == i /=)
Show - klasa typów, których wartości można wypisać na ekranie
Enum - klasa typów wyliczeniowych, dla których muszą istnieć m. in. operacje brania poprzednika i następnika

W interpreterze Hugs możemy wyznaczyć typ dowolnej zmiennej, funkcji lub wyrażenia korzystając z polecenia

: type < wyrażenie>

lub w skrócie

:t <wyrażenie>