

Erlang Academy

Лекция 2

План

- Типы данных в Erlang
- Булева алгебра и операторы сравнения
- Сопоставление по образцу (pattern matching)
- Рекурсия
- Модули
- Что почитать?

Типы данных

Числа:

- Целое (Integer)
- С плавающей запятой (Float)

Структуры:

- Кортеж (Tuple)
- Список (List)
- Карта (Map)

Идентификаторы:

- Атом (Atom)
- Идентификатор процесса (Pid)
- Порт (Port)
- Ссылка (Reference)

Типы данных

Низкоуровневые:

Бинарные данные (Binary)

Функциональные:

Функция (Fun)

Псевдотипы:

- Строка (String), в действительности является списком
- Запись (Record), в действительности является кортежем
- Булево значение (Boolean), в действительности является атомом

Число

Целое (Integer)

10.

-234.

16#AB10F.

2#110111010.

\$J.

\$€.

Число

С плавающей запятой (Float)

17.368.

-56.654.

12.34E-10.

ATOM

abcef.

start_with_a_lower_case_letter.

'Blanks can be quoted'.

'Anything inside quotes \n\012'.

true.

false.

ok.

error.

Кортеж

{1, 2, 3}.

{123, bcd}.

{123, def, abc}.

{person, 'Joe', 'Armstrong'}.

{abc, {def, 123}, jkl}.

{}.

{ok, []}.

{error, bad_credentials}.

Список

[1,2,3].

[a,b,c,d].

[{key1, value1}, {key2, value2}].

Строка

String = "test".

String = [\$t, \$e, \$s, \$t].

String = [116, 101, 115, 116].

Бинарные данные

Bin1 = <<1,2,3,0,255>>.

Bin2 = <<"Some Text">>.

Bin2 = <<83, 111, 109, 101, 32, 84, 101, 120, 116>>.

Функция

F1 = fun (X) -> X + 1 end.

F2 = fun lists:map/2.

Операторы сравнения

Операторы сравнения:

< меньше

> больше

=< меньше или равно

>= больше или равно

== равно

/= не равно

:= эквивалентно

=/= не эквивалентно

Приоритеты типов:

number < atom < reference < fun < port < pid < tuple < list
< bit string

Сопоставление по образцу

Кортеж

$X = \{1, 2, 3\}.$

$\{1, A, B\} = X.$

A. %% 2

B. %% 3

Список

$L = [1, 2, 3].$

$[H | T] = L.$

H. %% 1

T. %% [2, 3]

Сопоставление по образцу

$A = 10.$	%% Удастся - свяжет A с 10
$\{B, C, D\} = \{10, \text{foo}, \text{bar}\}.$	%% Удастся - свяжет B с 10, %% C с foo, D с bar
$\{A, A, B\} = \{\text{abc}, \text{abc}, \text{foo}\}.$	%% Удастся - свяжет A с abc, %% B с foo
$\{A, A, B\} = \{\text{abc}, \text{def}, 123\}.$	%% Свалится
$[A, B, C] = [1, 2, 3].$	%% Удастся - свяжет A с 1, %% B с 2, C с 3
$[A, B, C, D] = [1, 2, 3].$	%% Свалится

Сопоставление по образцу

Варианты сопоставления для списка

$L = [1, 2, 3]$.

$[1, 2, 3] = L$.

$[1 \mid [2, 3]] = L$.

$[1, 2 \mid [3]] = L$.

$[1, 2, 3 \mid []] = L$.

$[1 \mid [2 \mid [3]]] = L$.

$[1 \mid [2 \mid [3 \mid []]]] = L$.

Сопоставление по образцу

$[A, B | C] = [1, 2, 3, 4, 5, 6, 7]$. %% Удастся - $A = 1$, $B = 2$,
%% $C = [3, 4, 5, 6, 7]$

$[H | T] = [1, 2, 3, 4]$. %% Удастся - $H = 1$, $T = [2, 3, 4]$

$[H | T] = [abc]$. %% Удастся - $H = abc$, $T = []$

$[H | T] = []$. %% Свалится

$\{A, _, [B | _], \{B\}\} = \{abc, 23, [22, x], \{22\}\}$. %% Удастся
%% $A = abc$, $B = 22$

Рекурсия

```
double([H|T]) ->  
    [2*H|double(T)];  
double([ ]) ->  
    [ ].
```

Рекурсия

member(H, [H|_]) ->
true;

member(H, [_|T]) ->
member(H, T);

member(_, []) ->
false.

Модули

Файл: demo.erl

Содержимое:

```
-module(demo).  
-export([double/1]).
```

```
double(X) ->  
    times(X, 2).
```

```
times(X, N) ->  
    X * N.
```

Что почитать?

- [Fred Hebert - Learn You Some Erlang for great good! \(ссылка\)](#)
- [Фред Хеберт - Изучай Erlang во имя добра! \(ссылка\)](#)