




Язык программирования ML: Данные

Буслама Анис
Фатахов Тимур

5030102/20201



Введение

- ML (MetaLanguage) — семейство функциональных языков программирования
- Основные особенности:
 - Статическая сильная типизация.
 - Автоматическое выведение типов.
 - Поддержка функционального стиля и рекурсии.
- Система типов играет ключевую роль в языке.



Примитивные типы

- Основные встроенные типы:

- Int – целые числа
- Real – вещественные числа
- Bool – логические значения (true/false)
- Char – символ
- String - строка

```
val x : int = 42;  
val pi : real = 3.14;  
val flag : bool = true;  
val c : char = #"a";  
val s : string = "Hello, ML!";
```



Составные типы

- Кортеж (tuple) — фиксированное количество элементов разных типов.
- Список (list) — упорядоченный набор элементов одного типа.

```
val pair = (1, "abc");           (* кортеж *)  
val numbers = [1,2,3,4,5];      (* список *)  
val mixed = [(1,true), (2,false)];
```



Пользовательские типы

- В ML можно определять новые типы:
 - алгебраические типы (перечисления и конструкции с параметрами);
 - записи (record);
 - псевдонимы для типов (type).

```
datatype color = Red | Green | Blue;
```

```
datatype shape =  
    Circle of real  
    | Rectangle of real * real;
```

```
type person = {name:string, age:int};
```



Операции над типами

- Арифметические операции: +, -, *, /
- Логические операции: andalso, orelse, not
- Строковые операции: ^ (конкатенация)
- Операции со списками:
 - :: - добавить элемент в начало;
 - @ - объединить списки;
 - hd, tl - первый элемент и "хвост".

```
val sum = 3 + 5;  
val text = "Hello " ^ "World";  
val lst = 1 :: [2,3,4];  
(* добавление элемента *)  
val combined = [1,2] @ [3,4];  
(* объединение списков *)
```



Преобразование типов

- В ML нет автоматического приведения типов.
- Используются стандартные функции для преобразований:

<code>val r = Real.fromInt 5;</code>	<code>(* int → real *)</code>
<code>val i = Real.floor 3.99;</code>	<code>(* округление вниз *)</code>
<code>val s = Int.toString 42;</code>	<code>(* int → string *)</code>
<code>val n = String.size "abc";</code>	<code>(* string → int *)</code>



Сравнение типов

- Для сравнения значений используется = и <>.
- Сравнение поддерживается для:
 - примитивных типов;
 - списков, кортежей;
 - пользовательских алгебраических типов.

```
val eq1 = (3 = 3);           (* true *)  
val eq2 = ("abc" = "def");   (* false *)  
val eq3 = ([1,2] = [1,2]);   (* true *)
```

```
val isRed = (Red = Red);     (* true *)  
val isSameShape = (Circle 2.0 = Circle 2.0); (* true *)
```



Заключение

- Типы данных — фундамент языка ML.
- Система типов обеспечивает:
 - надежность программ;
 - простоту написания кода (автовывод типов);
 - возможность создания сложных структур.
- ML стал основой для многих современных языков (OCaml, F#, Haskell).



ИСТОЧНИКИ

- Robin Milner, Mads Tofte, Robert Harper, David MacQueen — The Definition of Standard ML (Revised), MIT Press, 1997.
- Paulson, L. C. — ML for the Working Programmer, Cambridge University Press, 1996.
- Michael R. Hansen, Hans Rischel — Functional Programming Using F#, Springer, 2013 (для сравнения с ML).
- Официальная документация по Standard ML:
 - SML Family GitHub
 - SML Basis Library
- Введение в Standard ML:
 - Learn SML in Y Minutes

