

# **ГЛАВА 3**

## **Атрибуты `deftemplate`**

# Атрибуты deftemplate

- В языке CLIPS предусмотрен целый ряд атрибутов слота, которые могут быть заданы при определении слотов конструкции deftemplate. Применение этих атрибутов позволяет упростить разработку и сопровождение экспертной системы и обеспечивает строгий контроль типов и проверку ограничений.
- К тому же обеспечивается возможность определить допустимые типы и значения, которые могут храниться в слоте, а для числовых значений может быть указан допустимый диапазон.

# Атрибуты deftemplate

- Конструкции multislots позволяют указывать минимальное и максимальное количество полей, которые они могут содержать.
- Наконец, атрибут default предоставляет возможность определять заданное по умолчанию значение слота, которое будет использоваться, если значение соответствующего слота не задано.

# Атрибуты ограничения

- CLIPS поддерживает четыре атрибута ограничения:

```
<constraint-attribute> ::= <type-attribute> |  
                           <allowed-constant-attribute> |  
                           <range-attribute> |  
                           <cardinality-attribute>
```

- атрибут **type**;
- атрибут допустимого значения;
- атрибут **range**;
- атрибут **cardinality**.

# Атрибут **type**

- Атрибут **type** определяет типы данных, которые могут храниться в слоте.
- Атрибут **type** имеет общий формат

**(type <type-specification>),**

в котором в качестве параметра <type-specification> может быть либо задана переменная ?VARIABLE, либо одна или несколько символических спецификаций: SYMBOL, STRING, LEXEME, INTEGER, FLOAT, NUMBER, INSTANCE-NAME, INSTANCE-ADDRESS, INSTANCE, FACT-ADDRESS или EXTERNAL-ADDRESS.

# Атрибут `type`

- Если используется переменная `?VARIABLE`, то слот может содержать данные любого типа (по умолчанию для всех слотов предусмотрен именно такой способ их применения).
- Если же используется одна или несколько символических спецификаций типа, применение слота ограничивается одним из указанных типов.
- Использование спецификации типа `LEXEME` эквивалентно заданию спецификаций `SYMBOL` и `STRING`.
- Использование спецификации типа `NUMBER` эквивалентно заданию спецификаций `INTEGER` и `FLOAT`, а применение спецификации типа `INSTANCE` эквивалентно заданию спецификаций `INSTANCE-NAME` и `INSTANCE-ADDRESS`.

# Атрибут type

- Пример:

```
(deftemplate person
  (multislot name (type SYMBOL))
  (slot age (type INTEGER)))
```

- После определения этой конструкции deftemplate система CLIPS автоматически предписывает применение заданных ограничений к любым атрибутам слотов.

```
CLIPS> (assert (person (name Fred Smith)
                       (age four)))  
[CSTRNCHK1] A literal slot value found in the assert  
command does not match the allowed types for slot age.  
CLIPS>
```

# Атрибут допустимого значения

- Язык CLIPS позволяет не только регламентировать перечень допустимых типов с помощью атрибута **type**, но и дает возможность задавать список допустимых значений для конкретного типа.
- Например, если в приведенную ранее конструкцию **deftemplate** с именем **person** дополнительно вводится слот **gender** (пол), может быть реализована возможность ограничить перечень допустимых символов для этого слота значениями *male* и *female*:



# Атрибут допустимого значения

```
(deftemplate person
  (multislot name (type SYMBOL))
  (slot age (type INTEGER))
  (slot gender (type SYMBOL)
    (allowed-symbols male female)))
```

- В языке CLIPS предусмотрено восемь различных атрибутов допустимого значения: **allowed-symbols**, **allowed-strings**, **allowed-lexemes**, **allowed-integers**, **allowed-floats**, **allowed-numbers**, **allowed-instance-names** и **allowed-values**.

# Атрибут допустимого значения

```
<allowed-constant-attribute>  
    ::= (allowed-symbols <symbol-list>) |  
        (allowed-strings <string-list>) |  
        (allowed-lexemes <lexeme-list> |  
        (allowed-integers <integer-list>) |  
        (allowed-floats <float-list>) |  
        (allowed-numbers <number-list>) |  
        (allowed-instance-names <instance-list>) |  
        (allowed-classes <class-name-list>)  
        (allowed-values <value-list>)
```

```
<symbol-list> ::= <symbol>+ | ?VARIABLE
```

```
<string-list> ::= <string>+ | ?VARIABLE
```

```
<lexeme-list> ::= <lexeme>+ | ?VARIABLE
```

# Атрибут допустимого значения

- За каждым из этих атрибутов должно следовать либо обозначение переменной `?VARIABLE` (которое указывает на то, что любые значения заданного типа являются допустимыми), либо список значений этого типа, следующего за префиксом `allowed-`.
- Например, за атрибутом **`allowed-lexemes`** должно следовать либо обозначение `?VARIABLE`, либо список символов и (или) строк.
- По умолчанию атрибут допустимого значения для слотов имеет вид (`allowed-values` `?VARIABLE`).

# Атрибут допустимого значения

- **ЗАМЕЧАНИЕ:** следует отметить, что атрибуты допустимого значения не ограничивают состав допустимых типов слота.
- Например, конструкция (`allowed-symbols male female`) не налагает такого ограничения, чтобы типом слота *gender* был символ.
- Эта конструкция указывает, что **если значением слота является символ**, то им должен быть один из двух символов — либо `male`, либо `female`.
- Если бы атрибут (`type SYMBOL`) был удален, то допустимым значением для слота *gender* были бы любая строка, целое число или число с плавающей точкой.

# Атрибут допустимого значения

- Атрибут `allowed-values` можно использовать, чтобы полностью ограничить множество допустимых значений для слота заданным списком.
- Например, после того как конструкция *deftemplate* с именем *person* будет заменена следующей, в результате произойдет то, что пределы допустимых типов для слота *gender* ограничатся символами:

```
(deftemplate person
  (multislot name (type SYMBOL))
  (slot age (type INTEGER))
  (slot gender (allowed-values male female)))
```

# Атрибут **range**

- Атрибут **range** позволяет задавать минимальные и максимальные допустимые числовые значения.
- Атрибут **range** имеет общий формат **(range <lower-limit> <upper-limit>)**, в котором параметры <lower-limit> и <upper-limit> представляют собой либо обозначение ?VARIABLE, либо числовое значение.
- Терм <lower-limit> указывает минимальное значение для слота, а терм <upper-limit> задает максимальное значение для слота.

# Атрибут `range`

- Обозначение `?VARIABLE` указывает, что не задано либо минимальное, либо максимальное значение (в зависимости от того, находится ли оно на первом или на втором месте).
- Например, чтобы предотвратить возможность помещать в слот отрицательные значения, слот `age` в конструкции `deftemplate` с именем `person` можно изменить следующим образом:

```
(deftemplate person
  (multislot name (type SYMBOL))
  (slot age (type INTEGER) (range 0 ?VARIABLE)))
```

# Атрибут `range`

- А если бы потребовалось явно сформулировать предположение, что никто не сможет прожить больше 125 лет, и ввести тем самым дополнительное ограничение, то можно было бы заменить это определение атрибута **`range`** определением (`range 0 125`).
- Как и в случае атрибутов допустимого значения, атрибут **`range`** не ограничивает тип значения слота таким образом, что если в нем заданы числа, то и тип должен быть числовым.
- Этот атрибут ограничивает только допустимые числовые значения слота заданным диапазоном, **если значение слота является числовым**. По умолчанию в качестве атрибута **`range`** для слотов применяется (`range ?VARIABLE ?VARIABLE`).



# Атрибут **cardinality**

- Атрибут **cardinality** позволяет задавать минимальное и максимальное количество значений, которые могут храниться в конструкции **multislot**.
- Атрибут **cardinality** имеет общий формат (**cardinality** <**lower-limit**> <**upper-limit**>), в котором термы < **lower-limit**> и < **upper-limit**> представляют собой либо обозначение ?**VARIABLE**, либо положительное целое число.
- Терм <**lower-limit**> показывает минимальное количество значений, которое может содержаться в слоте, а терм <**upper-limit**> позволяет указать максимально допустимое количество значений, содержащихся в слоте.

# Атрибут **cardinality**

- Обозначение **?VARIABLE** показывает, что не задано либо минимальное, либо максимальное количество значений, которое может быть указано в слоте (в зависимости от того, находится это обозначение на первом или втором месте).
- По умолчанию атрибут **cardinality** для любого многозначного значения для конструкции **multislot** имеет вид  
(**cardinality ?VARIABLE ?VARIABLE**).

# Атрибут `cardinality`

- Пример:

```
(deftemplate volleyball-team
  (slot name (type STRING))
  (multislot players (type STRING)
                (cardinality 6 6))
  (multislot alternates (type STRING)
                (cardinality 0 2)))
```

- Описан состав волейбольной команды; в этой команде должно быть шесть игроков, а количество запасных игроков может достигать двух.
- К каждому значению, содержащемуся в конструкции **multislot**, применяются ограничения типа допустимого значения и диапазона.

# Атрибут **default**

- Как правило, каждый факт **deftemplate**, вводимый в список фактов, имеет явно заданное значение для каждого слота.
- Часто бывает удобно автоматически сохранять в слоте указанное значение, если в команде добавления фактов явно не задано какое-либо значение.
- Возможность задавать применяемое по умолчанию значение обеспечивается атрибутом **default**.
- Атрибут **default** имеет общий формат  
(**default <default-specification>**),  
в котором терм **<default-specification>** может представлять собой обозначение **?DERIVE** или **?NONE**, единственное выражение (для однозначного слота), либо от нуля и больше выражений (для многозначного слота).

# Атрибут **default**

- Если в атрибуте **default** задано обозначение **?DERIVE**, то для данного слота должно быть выведено логическим путем определенное значение, которое соответствует всем атрибутам слота.
- Если для слота не задан атрибут **default**, то предполагается, что этот атрибут имеет вид (**default ?DERIVE**).
- Применительно к однозначному слоту это означает, что выбирается значение, которое удовлетворяет всем требованиям к атрибутам типа, диапазона и допустимого значения для этого слота.

# Атрибут default

- Если в заданном по умолчанию значении для многозначного слота содержится одно или несколько значений, то каждое из этих значений должно соответствовать атрибутам типа, диапазона и допустимого значения для этого слота.

```
CLIPS> (clear).  
CLIPS>  
(deftemplate example  
  (slot a)  
  
  (slot b (type INTEGER))  
  (slot c (allowed-values red green blue))  
  (multislot d)  
  (multislot e (cardinality 2 2)  
               (type FLOAT)  
               (range 3.5 10.0))).  
CLIPS> (assert (example)).
```

# Атрибут default

- Пример значений, выведенных логическим путем

```
CLIPS> (facts)␣  
f-0 (example (a nil)  
      (b 0)  
      (c red)  
      (d)  
      (e 3.5 3.5))
```

# Атрибут default

- По умолчанию значения для слотов в приведенном выше примере берутся из ограничения по типу.
- Это **nil** для типа `symbol`, `""` для типа `string`, **0** для `INTEGER`, **0.0** для `FLOAT`, `[nil]` для `INSTANCE-NAME`, указатель на несуществующий объект для `INSTANS-ADDRESS`, указатель на несуществующий факт для `FACT-ADDRESS` и `NULL` для `EXTERNAL-ADDRESS`.



# Атрибут **default**

- Если в атрибуте **default** задано обозначение **?NONE**, то необходимо предусматривать применение некоторого значения для данного слота во время ввода факта в список фактов.
- Иными словами, в таком случае применяемое по умолчанию значение не предусмотрено. В качестве примера можно привести следующий диалог:

```
CLIPS> (clear).  
CLIPS>  
(deftemplate example  
  (slot a)  
  (slot b (default ?NONE))).  
CLIPS> (assert (example)).  
[TMPLTRHS1] Slot b requires a value because of its  
(default ?NONE) attribute.
```

# Атрибут default

```
CLIPS> (assert (example (b 1)))↵
<Fact-0>
CLIPS> (facts)↵
f-0      (example (a nil) (b 1))
For a total of 1 fact.
CLIPS>
```

# Атрибут **default**

- Если используется одно или несколько выражений с атрибутом **default**, то во время синтаксического анализа слота эти выражения вычисляются и полученное значение сохраняется в слоте каждый раз, когда значение остается не заданным в команде `assert`.

```
CLIPS> (clear).  
CLIPS>  
(deftemplate example  
  (slot a (default 3))  
  (slot b (default (+ 3 4)))  
  (multislot c (default a b c))  
  (multislot d (default (+ 1 2) (+ 3 4))))
```

# Атрибут **default**

- Атрибут **default** для однозначного слота должен содержать одно и только одно выражение.
- Если же в атрибуте **default** для многозначного слота выражения не заданы, то для применяемого по умолчанию значения используется многозначная величина с количеством значений, равным нулю.
- В противном случае возвращаемые значения всех выражений группируются вместе для формирования одного многозначного значения.

# Атрибут default

```
CLIPS> (clear).┘
```

```
CLIPS>
```

```
(deftemplate example
```

```
  (slot a (default 3))
```

```
  (slot b (default (+ 3 4)))
```

```
  (multislot c (default a b c))
```

```
  (multislot d (default (+ 1 2) (+ 3 4))))).┘
```

```
CLIPS> (assert (example)).┘
```

```
<Fact-0>
```

```
CLIPS> (facts).┘
```

```
f-0      (example (a 3) (b 7) (c a b c) (d 3 7))
```

```
For a total of 1 fact.
```

```
CLIPS>
```

# Атрибут **default-dynamic**

- Если используется атрибут **default**, то применяемое по умолчанию значение для слота определяется во время синтаксического анализа объявления слота.
- Предусмотрена также возможность обеспечить выработку применяемого по умолчанию значения во время ввода в список фактов того факта, в котором будет использоваться это значение, предусмотренное по умолчанию.
- Для выполнения такой задачи используется атрибут **default-dynamic**.

# Атрибут **default-dynamic**

- Если значение слота, в котором применяется атрибут **default-dynamic**, остается не заданным в команде **assert**, то вычисляется выражение, заданное с помощью атрибута **default-dynamic**, которое затем используется в качестве значения слота.

```
CLIPS> (deftemplate templ
(slot a (default-dynamic (+ 2 ?*x*))))
CLIPS> (defglobal ?*x* = 2)
CLIPS> (assert (templ))
==> f-5      (templ (a 4))
<Fact-5>
```