

# Программирование на языке CLIPS



## 1. Правила и функции в CLIPS

# Основные модули CLIPS

CLIPS включает

- язык *представления порождающих правил;*
- язык *описания процедур.*

# Основные компоненты языка описания правил

Основными компонентами языка описания правил являются:

- база фактов (*fact base*);
- база правил (*rule base*).

На них возлагаются следующие функции:

- база фактов представляет исходное состояние проблемы;
- база правил содержит операторы, которые преобразуют состояние проблемы, приводя его к решению.

# *Машина логического вывода*

## CLIPS

*Машина логического вывода CLIPS* циклически сопоставляет факты и правила и выясняет, какие из правил можно активизировать.

Каждый цикл состоит из трех шагов:

1. *сопоставление* фактов и правил;
2. *выбор* правила, подлежащего активизации;
3. *выполнение* действий, предписанных правилом.

Такой трехшаговый циклический процесс иногда называют "циклом распознавание— действие".

# 1. Правила и функции в CLIPS



## 1.1. Факты

# Включение в базу фактов

- Сразу после запуска CLIPS-приложения на выполнение на экране появится приглашение, извещающее пользователя, что он работает с интерпретатором.
- CLIPS>
- В режиме интерпретатора пользователь может использовать множество команд

# Включение в базу фактов

Факты можно включить в базу фактов прямо из командной строки с помощью команды *assert*, например:

```
CLIPS> (assert (today is Sunday))
```

```
<Fact-0>
```

```
CLIPS> (assert (weather is warm))
```

```
<Fact-1>
```

# Вывод списка фактов

Для вывода списка фактов, имеющихся в базе, используется команда *facts*:

```
CLIPS> (facts)
```

```
f-0 (today is Sunday)
```

```
f-1 (weather is warm)
```

В последних версиях CLIPS, в частности, в той, которая работает в операционной среде Windows, такие команды, как *facts*, можно вызывать с помощью меню.



# Удаление фактов из базы

Для удаления фактов из базы используется команда *retract*.

```
CLIPS> (retract 1)
```

```
CLIPS> (facts)
```

```
f-0 (today is Sunday)
```

# Включение в базу массива фактов

В тексте программы факты можно включать в базу не по одиночке, а целым массивом.

Для этого в CLIPS имеется команда *deffacts*.

```
(deffacts today  
  (today is Sunday)  
  (weather is warm) )
```

Выражение начинается с команды *deffacts*, затем приводится имя списка фактов, который необходимо определить (в нашем примере — *today*), а за ним следуют элементы списка, причем их количество не ограничивается.

# Удаление массива фактов

Массив фактов можно удалить из базы командой *undeffacts*:

```
CLIPS> (undeffacts today)
```

# Загрузка фактов из файла

Выражение *deffacts* можно записать в текстовый файл с помощью редактора CLIPS или любого другого текстового редактора.

Загрузить этот файл в дальнейшем можно с помощью команды в меню *File* либо из командной строки.

```
CLIPS> (load "my file")
```

# Команда *reset*

Однако после загрузки файла факты не передаются сразу же в базу фактов CLIPS.

Команда *deffacts* просто указывает интерпретатору, что существует массив *today*, который содержит множество фактов.

Собственно загрузка выполняется командой *reset*.

```
CLIPS> (reset)
```

# Команда *reset*

Команда *reset* сначала очищает базу фактов, а затем включает в нее факты из *всех* ранее загруженных массивов. Она также добавляет в базу единственный системно определенный факт:

f-0 (initial-fact)

# Системно определенный факт

Добавление системно определенного факта делается по умолчанию, поскольку иногда имеет смысл включить в программу правило *start rule*, которое может быть сопоставлено с этим фактом и позволит выполнить какие-либо нестандартные инициализирующие операции.

# Режим слежения среды разработки

Можно проследить, как выполняется команда *reset*, если перед выполнением приведенных выше команд установить режим слежения среды разработки.

Для этого нужно вызвать команду *Watch* из меню *Execution* и установить в ней флажок *Facts*.



# 1. Правила и функции в CLIPS



## 1.2. Правила