

# aForth. Словари. Слова.

Шамин Алексей Алексеевич

21.07.08

## 1 Описание слов словаря *stddict*

### 1.1 Слово: /

#### Назначение

Операция деления 1го элемента стека на 0й. Со стека снимаются 2 элемента, на стек кладётся их частное. Требует наличия на стеке двух аргументов.

#### Пример использования

```
<целое1> <целое0> /  
<вещественное1> <вещественное0> /  
<вещественное1> <целое0> /  
<целое1> <вещественное0> /
```

При делении целого и вещественного тип приводится к вещественному.

### 1.2 Слово: -

#### Назначение

Операция вычитания 0го элемента стека из 1го. Требует наличия на стеке двух аргументов. Со стека снимаются 2 элемента, на стек кладётся их разность.

#### Пример использования

```
<целое1> <целое0> -  
<вещественное1> <вещественное0> -  
<вещественное1> <целое0> -  
<целое1> <вещественное0> -
```

При вычитании целого и вещественного тип приводится к вещественному.

### 1.3 Слово: \*

#### Назначение

Операция умножения 0го элемента стека на 1й. Требует наличия на стеке двух аргументов. Со стека снимаются 2 элемента, на стек кладётся их произведение.

#### Пример использования

```
<целое1> <целое0> *  
<вещественное1> <вещественное0> *  
<вещественное1> <целое0> *  
<целое1> <вещественное0> *
```

При умножении целого и вещественного тип приводится к вещественному.

### 1.4 Слово: +

#### Назначение

Операция сложения. Требует наличия на стеке двух аргументов. Со стека снимаются 2 элемента, на стек кладётся их сумма.

#### Пример использования

```
<целое1> <целое0> +  
<строка1> <строка0> +  
<вещественное1> <вещественное0> +  
<вещественное1> <целое0> +  
<целое1> <вещественное0> +
```

При сложении целого и вещественного тип приводится к вещественному.

### 1.5 Слово: =

#### Назначение

Операция сравнения двух верхних эл-тов стека. Со стека снимаются 2 элемента, на стек кладётся результат их сравнения. Если они равны, то на вершину ложится 1, если не равны, то 0. Требует наличия на стеке двух аргументов.

#### Пример использования

```
<целое1> <целое0> =  
<вещественное1> <вещественное0> =  
<вещественное> <целое> =  
<целое> <вещественное> =
```

При сравнении целого и вещественного тип приводится к вещественному.

### 1.6 Слово: >

#### Назначение

Операция сравнения двух верхних эл-тов стека. Со стека снимаются 2 элемента, на стек кладётся результат их сравнения. Если 1й элемент стека больше 0го, то на вершину ложится 1, иначе 0. Требует наличия на стеке двух аргументов.

#### Пример использования

```
<целое1> <целое0> >  
<вещественное1> <вещественное0> >  
<вещественное> <целое> >  
<целое> <вещественное> >
```

При сравнении целого и вещественного тип приводится к вещественному.

### 1.7 Слово: <

#### Назначение

Операция сравнения двух верхних эл-тов стека. Со стека снимаются 2 элемента, на стек кладётся результат их сравнения. Если 1й элемент стека меньше 0го, то на вершину ложится 1, иначе 0. Требует наличия на стеке двух аргументов.

#### Пример использования

```
<целое1> <целое0> <  
<вещественное1> <вещественное0> <  
<вещественное> <целое> <  
<целое> <вещественное> <
```

При сравнении целого и вещественного тип приводится к вещественному.

### 1.8 Слово: : (двоеточие)

#### Назначение

Начинает определение нового слова (его компиляцию).

#### Пример использования

```
: <новое слово> <слова> ;
```

### 1.9 Слово: ; (точка с запятой)

#### Назначение

Заканчивает определение нового слова (его компиляцию).

#### Пример использования

: <новое слово> <слова> ;

### 1.10 Слово: doword

#### Назначение

Исполняет строку как строку программы.

#### Пример использования

<строка> doword

### 1.11 Слово: include

#### Назначение

Загружает и исполняет файл.

#### Пример использования

<имя файла> include

### 1.12 Слово: if

#### Назначение

Условие if else endif

#### Пример использования

Форма 1 - краткая

<условие>

if

<операторы, выполняются, если условие не 0 (истинно)>

endif

Форма 2 - полная

<условие>

if

<операторы, выполняются, если условие не 0 (истинно)>

else

<операторы, выполняются, если условие 0 (ложно)>

endif

### 1.13 Слово: until

#### Назначение

Цикл begin-until. Цикл выполняется хотябы один раз и повторяется при ЛОЖНОМ (0) условии

### Пример использования

```
begin
  <Операторы, выполняются хотябы 1 раз>
  <Условие<>0 - выход, 0- повтор цикла>
until
```

## 1.14 Слово: while

### Назначение

Цикл begin-while-repeat. Цикл может не выполниться ни разу и повторяться при ИСТИННОМ (не 0) условии.

### Пример использования

```
begin
  <условие, если <>0, то цикл выполняется>
while
  <тело цикла>
repeat
```

## 1.15 Слово: aresize

### Назначение

Изменение размера массива в байтах. Изменяет размер массива на заданный. Если новый размер больше прежнего, то добавляемое место забивается нулями. Если новый размер меньше прежнего, то массив усекается.

### Пример использования

```
<новый размер массива> <массив> aresize
```

## 1.16 Слово: free

### Назначение

Освободить память, начальный адрес которой, находится на стеке. Память должна быть ранее выделена словом malloc.

### Пример использования

```
<адрес памяти> free
```

## 1.17 Слово: &

### Назначение

Требует один аргумент. Получить адрес переменной. Возвращает на стек адрес переменной или массива, лежащей на его вершине.

#### **Пример использования**

<переменная или массив> &

### **1.18 Слово: getmchar**

#### **Назначение**

Получить байт по указанному адресу, преобразовать в целое и поместить на стек.

#### **Пример использования**

<адрес> getmchar

### **1.19 Слово: getmint**

#### **Назначение**

Получить целое по указанному адресу и поместить на стек.

#### **Пример использования**

<адрес> getmint

### **1.20 Слово: getmreal**

#### **Назначение**

Получить вещественное по указанному адресу и поместить на стек.

#### **Пример использования**

<адрес> getmreal

### **1.21 Слово: getmshort**

#### **Назначение**

Получить короткое целое по указанному адресу и поместить на стек.

#### **Пример использования**

<адрес> getmshort

## **1.22 Слово: getmstr**

### **Назначение**

Получить строку по указанному адресу и поместить на стек.

### **Пример использования**

<адрес> getmstr

## **1.23 Слово: malloc**

### **Назначение**

Требует один аргумент. Выделить память размером в N байт и вернуть начальный адрес на стеке. Если память выделить не удалось - возвращается 0.

### **Пример использования**

<N> malloc

## **1.24 Слово: realloc**

### **Назначение**

Изменить размер памяти, выделенной с помощью слова malloc. Возвращает на стек адрес начала блока памяти или 0, если изменение размера не удалось.

### **Пример использования**

<новый размер в байтах> <адрес> realloc

## **1.25 Слово: setmchar**

### **Назначение**

Запись байта по адресу. Байт и адрес на стеке представлены целыми числами.

### **Пример использования**

<байт> <адрес> setmchar

## **1.26 Слово: setmint**

### **Назначение**

Запись целого по адресу.

**Пример использования**

<целое32> <адрес> setmint

**1.27 Слово: setmreal**

**Назначение**

Запись вещественного по адресу.

**Пример использования**

<вещественное> <адрес> setmreal

**1.28 Слово: setmshort**

**Назначение**

Запись короткого целого по адресу.

**Пример использования**

<короткое целое> <адрес> setmshort

**1.29 Слово: setmstr**

**Назначение**

Запись строки по адресу.

**Пример использования**

<строка> <адрес> setmstr

**1.30 Слово: mktime**

**Назначение**

Поместить на стек время сборки - строку в формате дд.мм.гггг-чч.мм.сс.

**Пример использования**

mktime



### 1.31 Слово: `subversion`

#### Назначение

Поместить на стек подверсию версию aForth - строку.

#### Пример использования

```
subversion
```

### 1.32 Слово: `version`

#### Назначение

Поместить на стек версию aForth - строку, содержащую несколько чисел, разделённых точками.

#### Пример использования

```
version
```

### 1.33 Слова: `.` `.x` `.X` `.C`

#### Назначение

Печать значения с вершины стека.

`.` - Целые печатаются в 10м виде.

`.x` - Целые печатаются в 16м виде. Буквы, обозначающие 16е цифры - строчные.

`.X` - Целые печатаются в 16м виде. Буквы, обозначающие 16е цифры - прописные.

`.C` - Печать символа, код которого находится на стеке в виде целого.

#### Пример использования

```
10 . " десятичная печать "
```

```
10 .x " шестнадцатеричная печать, строчные буквы-цифры "
```

```
10 .X " шестнадцатеричная печать, прописные буквы-цифры "
```

```
48 .C " печать символа с кодом 48 "
```

### 1.34 Слово: `intsize`

#### Назначение

Поместить на стек размер целого (в байтах).

#### Пример использования

```
intsize
```

### 1.35 Слово: `realloc`

#### Назначение

Поместить на стек размер вещественного (в байтах).

#### Пример использования

```
realloc
```

### 1.36 Слово: `shortsize`

#### Назначение

Поместить на стек размер короткого целого (в байтах).

#### Пример использования

```
shortsize
```

### 1.37 Слово: `size`

#### Назначение

Поместить на стек размер (в байтах) переменной или константы.

#### Пример использования

```
<Переменная или константа> size
```

### 1.38 Слово: `drop`

#### Назначение

Уничтожает вершину стека. Кол-во элементов стека уменьшается на 1.

#### Пример использования

```
<константа или переменная> drop
```

### 1.39 Слово: `dup`

#### Назначение

Дублирует значение вершины стека.

#### Пример использования

<константа или переменная> dup

#### 1.40 Слово: pover

##### Назначение

Копирование произвольного значения стека. На вершине - номер от начала стека

0 - верхнее значение и т.п.

1 0 pover - на вершине 1

3 1 1 pover - на вершине 3

0 pover то же, что и dup

1 pover то же, что и over

##### Пример использования

<константы или переменные> <константа или переменная> pover

#### 1.41 Слово: over

##### Назначение

Дублирует значение, следующее за вершиной стека (1й элемент стека).

##### Пример использования

<константа или переменная> <константа или переменная> over

#### 1.42 Слово: swap

##### Назначение

Меняет местами два верхних значения на стеке (0й и 1й элементы стека).

##### Пример использования

<константа или переменная> <константа или переменная> swap

#### 1.43 Слово: array

##### Назначение

Создаёт массив байт заданной длины и инициализирует его нулями.

##### Пример использования

<длина в байтах> array <имя массива>

## 1.44 Слово: variable

### Назначение

Создаёт переменную и инициализирует её начальным значением.

### Пример использования

<значение> variable <переменная>

Тип переменной определяется из контекста.

1 variable A - Создаёт целочисленную переменную A

1.0 variable A - Создаёт вещественную переменную A

"Строка" variable A - Создаёт строковую переменную A

## 1.45 Слово: !!

### Назначение

Снимает значение с вершины стека и помещает его в переменную.

### Пример использования

<значение> <переменная> !!

Типы переменной и значения должны быть совместимы. Если типы переменной и значения, то значение всегда приводится к типу переменной.