

Документация ParaCL

Содержание

1	Введение	2
2	Базовые конструкции	2
3	Типы данных	2
4	Массивами	2
4.1	Статические массивы	2
4.2	Динамические массивы	3
4.3	Массивы массивов и инициализация	3
5	Функции и области видимости	3
5.1	Объявление функции	4
5.2	Явный return	4
5.3	Значение блока	4
5.4	Именованные функции и рекурсия	4

1 Введение

Язык **ParaCL** — это простой язык программирования с поддержкой арифметики, ввода, вывода, условий, циклов, массивов и функций.

2 Базовые конструкции

В ParaCL доступны стандартные арифметические операции, условные конструкции `if` и циклы `while`, аналогично языку программирования C.

3 Типы данных

ParaCL поддерживает следующие типы данных:

- Целые числа
- Массивы
- Функции

4 Массивами

Массив — структура данных для хранения однотипных элементов. Существует два вида массивов: статические и динамические.

4.1 Статические массивы

Если размер массива известен на этапе компиляции, он размещается на стеке:

Пример статического массива

```
v0 = 42;  
v1 = repeat(v0, 5); // Массив из 5 элементов со значением 42
```

Индексация осуществляется через квадратные скобки:

Индексирование массива

```
v3 = 3;  
v4 = v1[2] + v1[v3];  
v5 = repeat(v4, v3); // Новый массив из 3 элементов
```

4.2 Динамические массивы

Если размер массива неизвестен заранее, создаётся динамический массив (в куче). Он освобождается автоматически по выходу из области видимости:

Пример динамического массива

```
v7 = ?;  
v8 = repeat(2, v7); // Массив из v7 элементов со значением 2
```

Для создания неинициализированного массива используется ключевое слово `undef`:

Неинициализированный массив

```
v9 = repeat(undef, ?); // Размер читается с stdin
```

4.3 Массивы массивов и инициализация

Массивы могут содержать другие массивы:

Массив массивов

```
v10 = repeat(v9, ?);
```

Для явной инициализации применяется ключевое слово `array`:

Пример инициализации

```
v11 = array(1, 2, 3, 4, 5);
```

Частичная инициализация и вложенные `repeat` также поддерживаются:

Частичная инициализация с `flatten`

```
v12 = array(undef, 1, 2, 3, 4, 5, repeat(undef, 5));  
v13 = array(repeat(1, 5), repeat(2, 5));  
// Результат: [1,1,1,1,1,2,2,2,2,2] (одномерный)
```

5 Функции и области видимости

В ParaCL функции — полноценные объекты.

5.1 Объявление функции

Функции определяются с помощью ключевого слова `func`.

Возврат последнего выражения

```
bar = func() {  
  x = 5;  
  y = 10;  
  x + y;  
}; // точка с запятой здесь возможна но необязательна  
t = bar(); // t == 15
```

5.2 Явный return

Возвращаемое значение с return

```
buz = func(x) {  
  if (x != 5)  
    return x;  
  y = 10;  
  x + y;  
}  
z = buz(6); // z == 6
```

5.3 Значение блока

Любой блок может возвращать значение, также блок захватывает внешние переменные:

Блок как выражение

```
t = { x = 5; y = 10; x + y; } // t == 15
```

5.4 Именованные функции и рекурсия

У функции может быть имя, если оно нужно (например для рекурсии) тогда оно записывается через двоеточие:

Пример рекурсивной функции

```
fact = func(x) : factorial {  
  res = 1;  
  if (x > 0)  
    res = factorial(x - 1);  
  res;  
}
```