

# STAGE-5：数组

计04 何秉翔 2020010944

## 1. 实验内容

### 1.1 step-11：数组

我们按照从前端、中端到后端的顺序分别介绍，在中间主要介绍实现的思路以及关键部分的代码。

#### 1.1.1 语法分析

数组首先需要有关于数组的声明，在此我们沿用原有的 `Declaration` 结点，在其中增加一些属性，比如数组初始化维度列表，初始值列表，来标记关于数组初始化的性质。

```
1 class Declaration(Node):
2     """
3     AST node of declaration.
4     """
5     def __init__(
6         self,
7         var_t: TypeLiteral,
8         ident: Identifier,
9         array_dim_list: Optional[list[IntLiteral]] = None,
10        init_expr: Optional[Expression] = None,
11        init_array_elements: Optional[list[IntLiteral]] = None,
12    ) -> None:
13        super().__init__("declaration")
14        self.var_t = var_t
15        self.ident = ident
16        self.array_dim_list = array_dim_list or NULL
17        self.init_expr = init_expr or NULL
18        self.init_array_elements = init_array_elements or NULL
19    ...
```

然后，对于数组元素的访问，比如读或者写，我们需要一个语法树节点，在此我们建立一个 `ArrayElement` 结点，在其中我们存放数组的标识符以及访问元素的 `index` 列表，比如对于 `a[0][1][2]`，访问列表就是 `{0, 1, 2}`

```

1 class ArrayElement(Expression):
2     """
3     AST node of array element.
4     """
5     def __init__(
6         self,
7         ident: Identifier,
8         array_dim_list: list[Expression],
9     ) -> None:
10         super().__init__("arrayElement")
11         self.ident = ident
12         self.array_dim_list = array_dim_list

```

接着我们更改对于 `Declaration` 的语法解析：

```

1 """
2 declaration : type Identifier
3             | type Identifier index_list
4 """

```

其中 `index_list` 为数组索引：

```

1 index_list : index_list_r LBrack expression RBrack
2 index_list_r : index_list_r LBrack expression RBrack
3              | empty

```

然后我们支持对于数组元素的赋值语句 `Assign`：

```

1 assignment : Identifier index_list Assign expression

```

对于数组值参数的传递，我们修改 `postfix` 如下，以支持数组元素传参：

```

1 postfix : Identifier index_list

```

### 1.1.2 语法树建立

### 1.1.3 语义分析 & 符号表构建

#### 1.1.4 TAC 生成

#### 1.1.5 后端

### 1.2 step-12: 为数组添加更多支持

我们从前端到后端依次进行分析。

#### 1.2.1 前端

#### 1.2.2 中端

#### 1.2.3 后端

## 2. 思考题

### 2.1 step-11 思考题

问题:

解答:

### 2.2 step-12 思考题

问题:

解答: