# 中间代码设计方案

中间代码设计是方便模块在翻译时的统一，中间代码设计应该要有足够的拓展性，可不断增加句式，在满足拓展性前提下足够简单方便，越简单对于模型捕捉到联系理论会足够容易，也意味着模型准确度会更高。

# 一 定义类别

## 整型

标识符: int

例子：int a；

## 字符型

标识符: string

例子：string a；

## 布尔型

标识符: bool

例子：bool a

## 空

标识符: void

## 数组定义格式

标识符: 类型+元素数量+变量名

例子：int 10 array

## 不指明类型

标识符: auto

例子：auto a

# 二 计算类别

在这个类别中无细分，与C++一致

# 三 判断类别

## 3.1 if类型

标识符: if +条件表达式

## 3.2 else 类型

标识符: else

## 3.3 elif类型

标识符: elif +条件表达式

# 四 循环类别

## 4.1 while类型

标识符: while+条件表达式

## 4.2 for类别

标识符：for + 遍历元素 + 遍历集合（遍历元素可能用temp，就是不指定的情况下）