18231047 王肇凯

错误处理

最初设计

建立错误类，存储错误类型、行号、列号、额外提示信息；再将所有错误对象整合为一个数组存储，并输出到文件中。

为了识别语义错误，需要建立符号表管理各个层次的变量，并存储各种信息（如变量维度、函数返回值类型等等）。在读到标识符时进行增/查操作，在类型不一致时报错。此外还需要求表达式的类型，判断其为char型或者int型。

实现与完善

将错误类别以宏的形式进行定义，作为错误编码

```
#define ERR_LEXER 'a'
#define ERR_REDEFINED 'b'
#define ERR_UNDEFINED 'c'
#define ERR_PARA_COUNT 'd'
#define ERR_PARA_TYPE 'e'
#define ERR_CONDITION_TYPE 'f'
#define ERR_NONRET_FUNC 'g'
#define ERR_RET_FUNC 'h'
#define ERR_INDEX_CHAR 'i'
#define ERR_CONST_ASSIGN 'j'
#define ERR_SEMICOL 'k'
```

```
#define ERR_RPARENT 'l'
#define ERR_RBRACK 'm'
#define ERR_ARRAY_INIT 'n'
#define ERR_CONST_TYPE 'o'
#define ERR_SWITCH_DEFAULT 'p'
```

设计`Error`类存储单个错误的各种信息。

```
class Error {
public:
    string msg;
    int line{};
    int column{};
    int eid;
    char err_code{};
    string rich_msg;
```

`Errors`类用静态成员变量存储全部错误对象，并提供静态方法将错误输出至文件。

```
class Errors {
public:
    static vector<Error> errors;

    static void add(const string &s, int line, int col,
int id);

    static void add(const string &s, int id);

    static void save_to_file(const string &out_path);
```

错误信息输出示例如下

```
Error in line 52, column 21: Para count mismatch (EID: d)
Error in line 53, column 22: Para type mismatch (EID: e)
Error in line 55, column 21: Para type mismatch (EID: e)
```

`SymTableItem`和`SymTable`类分别代表符号表项和整个符号表。设定了增加符号表项、查询符号表、栈式符号表增减层的方法。

```
enum STIType {
    constant,
    var,
    para,
    func
};

enum DataType {
    integer,
    character,
    void_ret,
    invalid
};

class SymTableItem {
public:
    string name;
    STIType stiType{};
    DataType dataType{};
    int num_para = 0;
    int dim = 0;
    bool valid = true;
    vector<DataType> types;

```

```
25      SymTableItem(string name, STIType stiType1, DataType
    dataType1) :
26              name(std::move(name)), stiType(stiType1),
    dataType(dataType1) {};
27
28      explicit SymTableItem(bool valid) : valid(valid) {};
29
30      SymTableItem() = default;
31
32      string to_str() const;
33  };
34
35  class SymTable {
36  public:
37      static vector<SymTableItem> items;
38      static vector<int> layers;
39      static unsigned int max_name_length;
40
41      static void add(const Token& tk, STIType stiType,
    DataType dataType);
42
43      static void add(const Token& tk, STIType stiType,
    DataType dataType, int dim);
44
45          static void add_func(const Token& tk, DataType
    dataType, int num_para, vector<DataType> types);
46
47      static void add_layer();
48
49      static void pop_layer();
50
51      static SymTableItem search(const Token &tk);
52
53      static void show();
54
```

符号表显示效果如下

```
==============================
NAME            KIND  TYPE DIM
------------------------------
func_switch_ch  func  int  0
func_switch_int func  int  0
------------------------------
c               para  char 0
tmp             var   int  0
==============================
```

在语法分析程序中增加对于`error()`函数的调用，以在适当地方进行报错，完成跳读，并将错误信息存储到`Errors`类的静态成员变量中。在整个程序分析结束后，将所有错误输出至文件。