# 测试用例

|  |
| --- |
| module A  {  module innerModule  {  struct internal  {  short c1 = "a"; // 错误：将String类型变量赋值给Short类型变量。  short i1=100000; // 错误：数据超出Short类型变量的上界  int16 i2=10;  int32 i2=100; // 错误：同个声明域不能声明同名变量。  int32 i3=100;  };  };  struct innerStruct  {  long i3=100;  int32 i4=100;  innerModule::internal test1;  internal test2; // 错误：命名空间引用不对。  };  };  module B  {  module middle  {  module inner  {  struct bottom  {  unsigned short i7=10;  uint16 i8=10;  uint32 i10=100;  long long i5=1000;  int64 i6=1000;  unsigned long i9=100;  };  };  };  };  struct C  {  unsigned long long i11=1000;  uint64 i12=100;  char c0='a';  string c1="abc";  boolean c2=true;  float c3=10.901f;  double c4=23.234d;  long double c5=12.23456432235d;  short arr[10]={0,1,2,3,4,5,6,7,8,9};  short arr2[true]; // 错误：数组的长度需为整数值。  short arr3[5]={0,'a',1.5f,"abc",9}; // 错误：数组内容与数组类型不符。  C testBool1;  A::innerModule::internal testBool2;  int16 testInt = 2 + 5 \* 2 / 3;  float testFloat = 1.2 \* 3.0 - 2.0 % 1.0;  float testFloat2 = ~2.0;  }; |

语义分析的过程中，需要检查出以上测试用例中，标红的错误。如果程序能够检查出所有的错误，语义分析便达到要求。

# 控制台打印结果

|  |
| --- |
| [Line 7:14]:Constant type "String" cannot be assigned to type "short".  [Line 8:12]:Value "100000" cannot be assigned to type "short".  [Line 10:9]:The identifier "i2" is already defined in scope "internal".  [Line 20:2]:Type "internal" is not defined yet.  [Line 57:12]:The array length must be an integer number. "Boolean" is provided.  [Line 58:18]:Constant type "Char" cannot be assigned to type "short".  [Line 58:22]:Constant type "Float" cannot be assigned to type "short".  [Line 58:27]:Constant type "String" cannot be assigned to type "short".  Completed. |

根据以上信息，可以判断出错误的发生位置，以及获得错误的内容信息。可以看到，所有的错误都被检测了出来。

# C++代码生成

将上方的测试用例中的错误删除后，再进行C++代码生成，可以得到以下结果：

|  |
| --- |
| typedef const char\* string;  namespace A  {  namespace innerModule  {  typedef struct internal  {  short i2 = 10;  }internal;  }  typedef struct innerStruct  {  int i3 = 100;  int i4 = 100;  innerModule::internal test1;  }innerStruct;  }  namespace B  {  namespace middle  {  namespace inner  {  typedef struct bottom  {  unsigned short i7 = 10;  unsigned short i8 = 10;  unsigned int i10 = 100;  long long i5 = 1000;  long long i6 = 1000;  unsigned int i9 = 100;  }bottom;  }  }  }  typedef struct C  {  unsigned long i11 = 1000;  unsigned long i12 = 100;  char c0 = 'a';  string c1 = "abc";  bool c2 = true;  float c3 = 10.901f;  double c4 = 23.234d;  long double c5 = 12.23456432235d;  short arr[10] = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9];  C testBool1;  A::innerModule::internal testBool2;  short testInt = 2 + 5 \* 2 / 3;  float testFloat = 1.2 \* 3.0 - 2.0 % 1.0;  float testFloat2 = ~2.0;  }C; |