

Институт Космических и информационных технологий
институт
Кафедра «Информатика»
кафедра

Преподаватель	<u></u> подпись, дата	И.В. Якимов инициалы, фамилия
Студент <u>КИ17-176/2 031722054</u>	<u></u> подпись, дата	<u>А.И. Гарбарт</u> инициалы, фамилия
номер группы, зачетной книжки		
Студент <u>КИ17-176/2 031723205</u>	<u></u> подпись, дата	<u>Л.С. Кислов</u> инициалы, фамилия
номер группы, зачетной книжки		

Красноярск 2019

Цель

Изучение основных методов организации таблиц идентификаторов в трансляторах языков программирования и методов синтаксического и контекстного анализа с их программной реализацией.

Задачи

1. Изучение теоретического материала об основных методах организации таблиц имен;
2. изучение теоретического материала по организации синтаксического и контекстного анализа языков программирования;
3. составление формального описания процессов синтаксического и контекстного анализа;
4. программная реализация по формальному описанию.

Основные конструкции языка

Объявление переменных

Состоит из 5 частей:

```
<type> <name> = <value>;
```

Где <type> – тип переменной (int, float, bool или char), <name> – имя переменной и <value> – значение. Опустить присваивание значения нельзя.

Объявление массивов

Состоит из 7 частей:

```
<type> <name> = <value>[<размерность>;
```

Где <type> – тип переменной (int, float, bool или char), <name> – имя переменной, <value> – значение и <размерность> – размерность объявляемого массива. Опустить присваивание значения нельзя.

Присваивание значений

Состоит из 4 частей

```
<name> = <value>;
```

Где <name> – имя переменной и <value> – значение.

Объявление условного оператора

Может быть двух видов:

```
if (<условие>) { # тело }
```

Либо:

```
if (<условие>) { # тело }
```

```
else { # тело }
```

Где <условие> – условие выполнения тела внутри первого блока.

Объявление циклов

Может быть трех видов:

```
while (условие) { # тело }
```

Либо:

```
do { # тело } while (условие);
```

Либо:

```
for (<выражение>; <условие>; < выражение>) { # тело }
```

Объявление выражений

Выражения могут быть самыми различными. Например:

- $a \geq b$
- $10 < c$
- $(10 + 5)$
- -1
- $(10 + a) / c * b$
- 10
- a
- 10.123

И так далее.

Объявление операций ввода и вывода

Для ввода используется следующая конструкция:

```
vox (<переменная>) ;
```

Для вывода используется следующая конструкция:

```
wouf (<выражения через запятую>) ;
```

Объявление функций

Функции объявляются по следующему шаблону:

```
func    <имя    функции> (<входные    параметры>)    ->  
<возвращаемый тип> {    # тело }
```

Входные параметры могут являться пустотой.

Обрабатываемые ошибки

Компилятор предупреждает о некоторых типах ошибок. При этом явно указывается строка, где была допущена ошибка.

Несоответствие типов

```
[~/Development/GNU/lab_2 » cat test03.simpl  
int a = 10.1;  
-----  
[~/Development/GNU/lab_2 » ./parser test03.simpl  
test03.simpl: 1.13: warning - types incompatible in assignment  
0
```

Рисунок 1 – Ошибка несоответствия типов

```
[~/Development/GNU/lab_2 » cat test03.simpl  
int a = 1;  
float b = a;  
-----  
[~/Development/GNU/lab_2 » ./parser test03.simpl  
test03.simpl: 2.12: warning - types incompatible in assignment  
0
```

Рисунок 2 – Еще одна ошибка несоответствия типов

Переменная не объявлена

```
[~/Development/GNU/lab_2 » cat test03.simpl  
b = 10;  
-----  
[~/Development/GNU/lab_2 » ./parser test03.simpl  
test03.simpl: 1.7: warning - Variable b isn't declared  
0
```

Рисунок 3 – Ошибка необъявленной переменной

Размерность массива не целочисленная константа

```
[~/Development/GNU/lab_2 » cat test03.simpl
int arr = int[1.23];

[~/Development/GNU/lab_2 » ./parser test03.simpl
test03.simpl: 1.20: warning – size of array must be const int
0
```

Рисунок 4 – Ошибка размерности массива

Типы несравнимы

```
[~/Development/GNU/lab_2 » cat test03.simpl
int arr = int[123];
int a = 123;
if (a > arr)
{
    wouf(123);
}

[~/Development/GNU/lab_2 » ./parser test03.simpl
test03.simpl: 3.12: warning – types in relop incompatible
0
```

Рисунок 5 – Ошибка несравнимых типов

Такие же ошибки существуют и для арифметических операций.

Синтаксические ошибки

```
[~/Development/GNU/lab_2 » cat test03.simpl
int arr = int[123]

[~/Development/GNU/lab_2 » ./parser test03.simpl
test03.simpl: test03.simpl:1.1: syntax error, unexpected end of file, expecting ;
```

Рисунок 6 – Отсутствующая точка с запятой

```
[~/Development/GNU/lab_2 » cat test03.simpl
int arr = int(123);

[~/Development/GNU/lab_2 » ./parser test03.simpl
test03.simpl: test03.simpl:1.1: syntax error, unexpected (, expecting [
```

Рисунок 7 – При объявлении массива использована '(', но ожидается '['

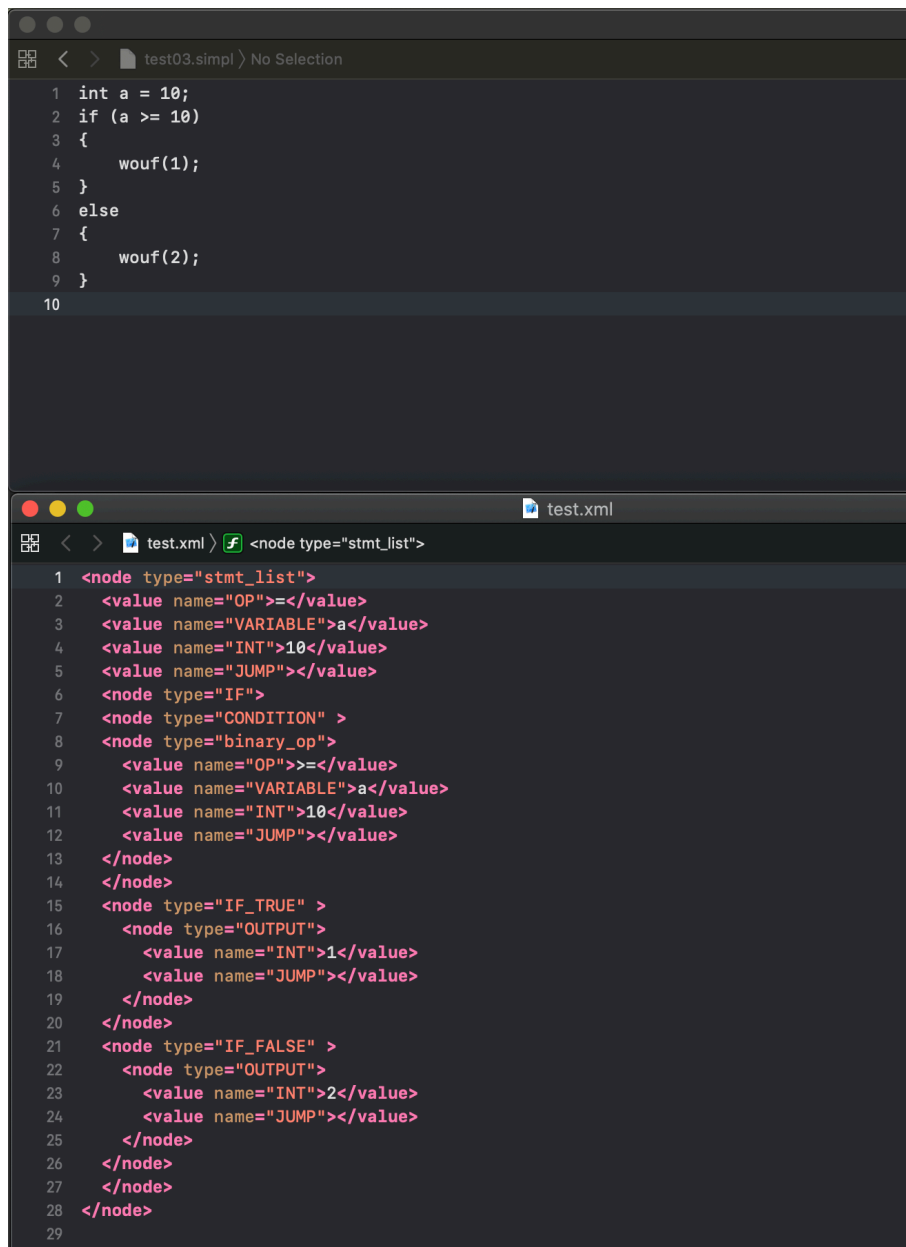
```
[~/Development/GNU/lab_2 » cat test03.simpl
while 0 == 0
{
    wouf(1);
}

[~/Development/GNU/lab_2 » ./parser test03.simpl
test03.simpl: test03.simpl:1.1: syntax error, unexpected integer, expecting (
```

Рисунок 8 – При объявлении цикла не указаны скобки

Дополнительные функции

В качестве дополнительной функции была реализована возможность экспортировать дерево программы в XML.



```
test03.simpl
1 int a = 10;
2 if (a >= 10)
3 {
4     wouf(1);
5 }
6 else
7 {
8     wouf(2);
9 }
10

test.xml
1 <node type="stmt_list">
2   <value name="OP">=</value>
3   <value name="VARIABLE">a</value>
4   <value name="INT">10</value>
5   <value name="JUMP"></value>
6   <node type="IF">
7     <node type="CONDITION" >
8       <node type="binary_op">
9         <value name="OP">=</value>
10        <value name="VARIABLE">a</value>
11        <value name="INT">10</value>
12        <value name="JUMP"></value>
13      </node>
14    </node>
15    <node type="IF_TRUE" >
16      <node type="OUTPUT">
17        <value name="INT">1</value>
18        <value name="JUMP"></value>
19      </node>
20    </node>
21    <node type="IF_FALSE" >
22      <node type="OUTPUT">
23        <value name="INT">2</value>
24        <value name="JUMP"></value>
25      </node>
26    </node>
27  </node>
28 </node>
29
```

Рисунок 9 – Исходный код программы и XML-дерево для нее

Примеры работы

Приедем несколько примеров обработки программ.

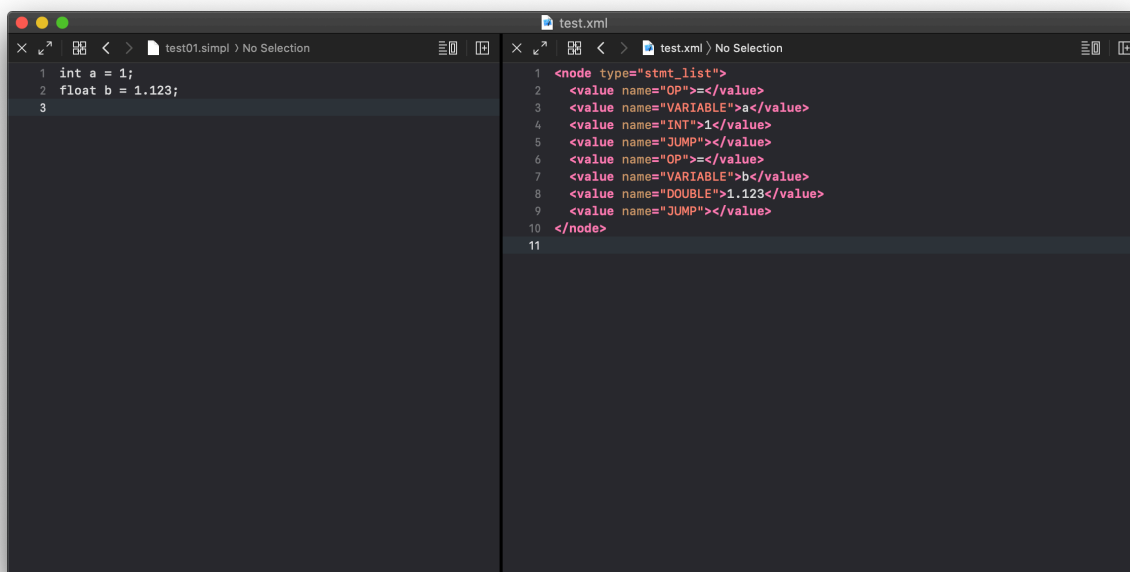


Рисунок 10 – Пример 1

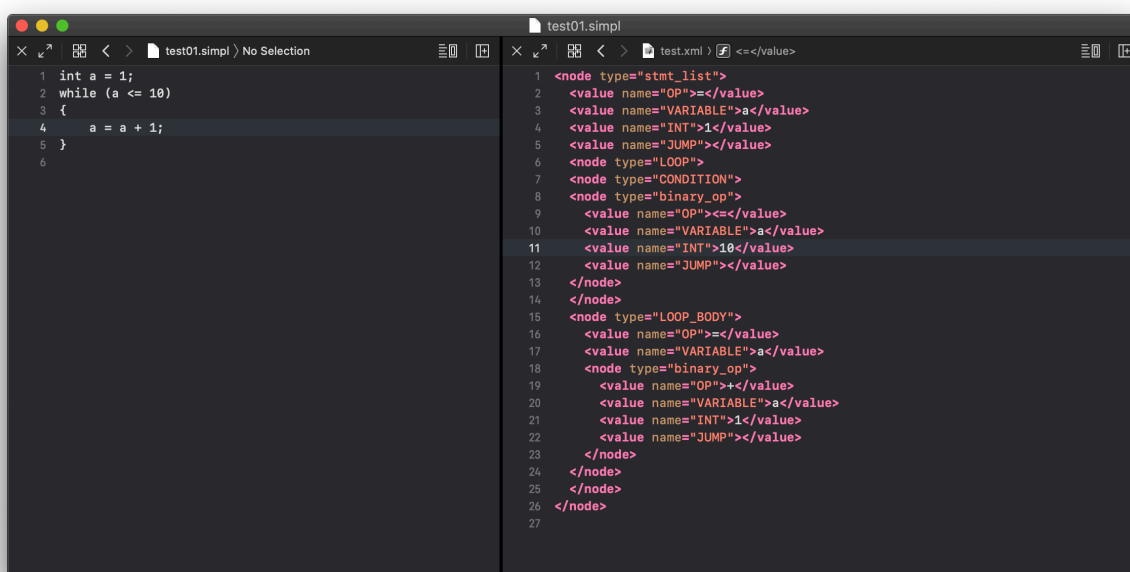


Рисунок 11 – Пример 2

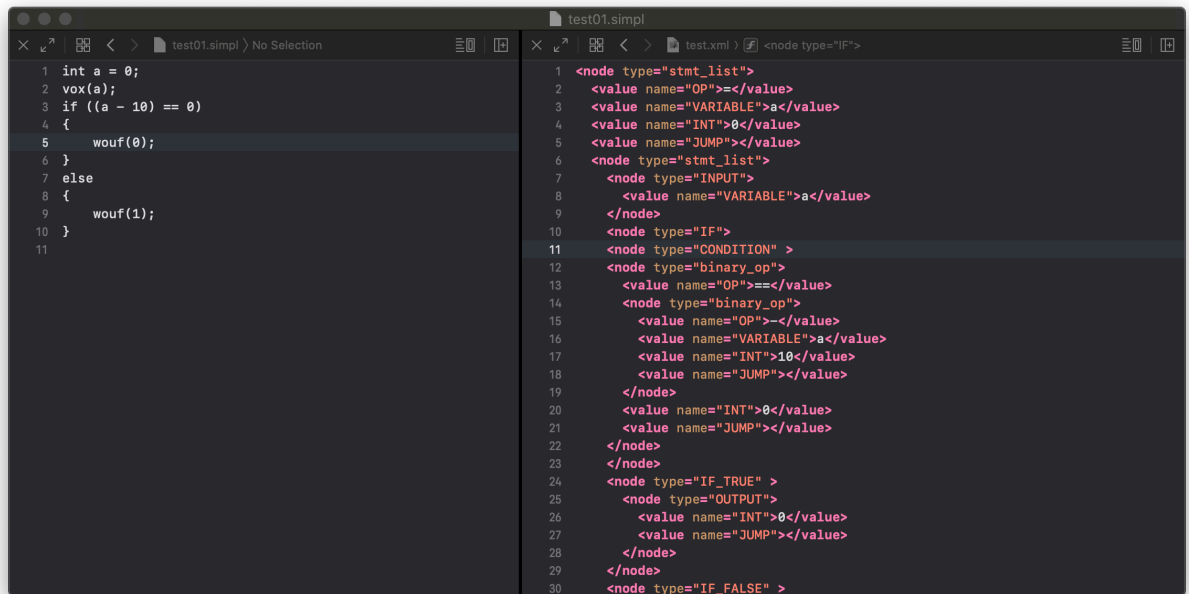


Рисунок 12 – Пример 3

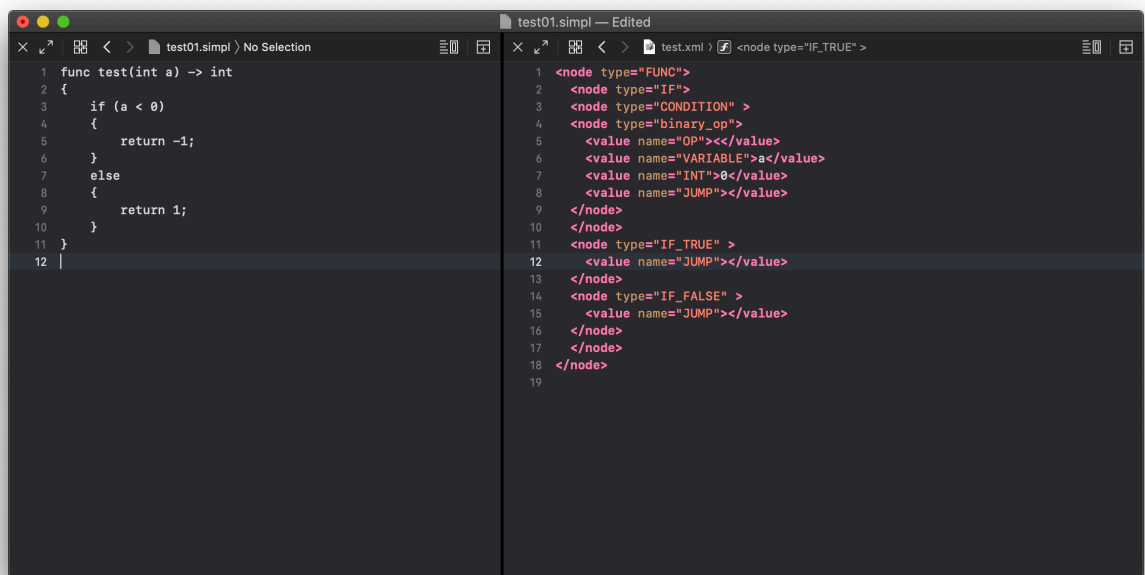


Рисунок 13 – Пример 4