

**Лабораторная работа 17-1 (2 часа)**  
**Конструирование программного обеспечения**

**Разработка лексического распознавателя (III часть)**

1. Используйте материалы лекции № 9-13.
2. Используйте результаты лабораторных работ № 13-16.
3. Создайте проект (VS201X, C++, консольное приложение) с именем **SELab17**.
4. Ознакомьтесь со спецификацией языка SVV-2015.

Компоненты языка	Описание
Символы	Windows-1251
Символы-сепараторы	пробел – допускается везде кроме идентификаторов и ключевых слов; ;(точка с запятой) – разделитель инструкций; { } – программный блок; ( ) – параметры; ( ) – приоритетность операций.
Идентификаторы	только малые буквы, от 1 до 5 букв; идентификатор не может совпадать с ключевыми словами; максимальное количество идентификаторов $2^{16}$
Типы данных	<b>integer</b> – целочисленные данные (четыре байта; диапазон от $-2^{31}$ до $2^{31}-1$ ); автоматическая инициализация 0; LE; <b>string</b> – строка, любые символы, (максимально 255 символов, первый байт – длина строки), автоматическая инициализация строкой длины 0
Операции с данными	+ – бинарный, суммирование, (integer, integer); + – бинарный, конкатенация, (string, string); - – бинарный, вычитание, (integer, integer); * – бинарный, умножение, (integer, integer); / – бинарный, деление, (integer, integer)
Программные конструкции	Главная функция (точка входа):  main { ... return <integer-идентификатор>   <integer-литерал>; }

	<p>Функции:</p> <pre>&lt;тип данных&gt; function&lt;идентификатор&gt; (&lt;тип данных&gt; &lt;идентификатор&gt;, ...) { ... return &lt;идентификатор&gt;; }</pre> <p>Параметры: передаются по значению.</p>
Литералы	<p>числа в диапазоне от <math>-2^{31}</math> до <math>2^{31}-1</math>, интерпретируются как integer, могут быть только rvalue;</p> <p>строки, символы, заключенные в '...' (одинарные кавычки), могут быть только rvalue</p>
Выражения	<p>арифметические с применением +, -, /, *, (); строковые с применением +,()</p>
Инструкции	<p>declare &lt;тип данных&gt; идентификатор; – объявление переменных;</p> <p>declare &lt;тип данных&gt; function идентификатор (&lt;тип данных&gt; идентификатор,... (&lt;тип данных&gt; идентификатор )); – объявление внешних функций;</p> <p>= присвоение значения;</p> <p>print &lt;идентификатор или литерал&gt;; – вывод в стандартный поток вывода</p>
Область видимости	<p>сверху вниз (по принципу C++); параметры – только внутри функции; объявление внутри функции – видно только внутри функции</p>
Память	Все переменные размещаются в стеке
Стандартная библиотека	<p>integer strlen (string) – длина строки;</p> <p>string substr (string, integer, integer) – подстрока;</p>
Максимальное количество функций	$2^{16}$

Максимальное количество идентификаторов в функции	$2^{16}$
---	----------

5. Текст программы на языке SVV-2015 для первоначального тестирования (контрольный пример):

```
integer function fi(integer x, integer y)
```

```
{
```

```
declare integer z;
```

```
z = x*(x+y);
```

```
return z;
```

```
};
```

```
string function fs (string a, string b)
```

```
{
```

```
declare string c;
```

```
declare string function substr(string a, integer p,  
integer n);
```

```
c = substr(a, 1,3)+ b;
```

```
return c;
```

```
};
```

```
main
```

```
{
```

```
declare integer x;
```

```
declare integer y;
```

```
declare integer z;
```

```
declare string sa;
```

```
declare string sb;
```

```
declare string sc;
```

```
declare integer function strlen(string p);
```

```
x = 1;
```

```
y = 5;
```

```
sa = '1234567890';
```

```
sb = '1234567890';
```

```
z = fi(x,y);
```

```
sc = fs(sa,sb);
```

```
print 'контрольный пример';
```

```
print z;
```

```
print sc;  
print strlen(sc);  
return 0;  
};
```

6. Разработайте приложение, которое выполняет следующее:

- вводит текст программы из входного файла (заданного параметром `–in:`);
- проверяет входные символы на допустимость, в соответствии с таблицей символов (лабораторные 14, 15);
- удаляет лишние пробелы и добавляет сепаратор для вычисления номера строки для каждой лексемы;
- выводит преобразованный текст программы в выходной файл (заданный параметром `–out:`) (лабораторная 15);
- формирует протокол работы (лабораторная 15);
- в том случае, если строка ошибочна, в протокол выводится номер этой строки, позиция в строке и описание ошибки.