

БНФ и примеры в Pascal

Терминальный словарь Бэкусовой Нормальной Формы (БНФ) содержит четыре символа:

< – открывающая угловая скобка;

> – закрывающая угловая скобка;

/ – знак альтернативы (союз “или”);

::= – “равно по определению” или “это есть”.

Правило грамматики в языке Бэкуса задается в виде, похожем на алгебраическое выражение и представляет собой формулу.левой частью выражения является определяемый нетерминальный символ (закрываемый в угловые скобки), после чего следует символ “равно по определению”. В правой части содержится цепочка символов терминального и нетерминального словарей того языка, для которого определяется правило грамматики. Нетерминальные символы, входящие в цепочку заключаются в угловые скобки. При этом предполагается, что эти символы ранее определены соответствующими правилами или будут определяться ниже. Возможные альтернативы в правой части разделяются знаком “или”.

При разборе текста (любой цепочки терминальных символов) можно выделить синтаксические конструкции, получив при этом новую цепочку нетерминальных и, возможно, “оставшихся” от предыдущего шага разбора терминальных символов с учетом альтернатив. Затем нужно повторить эту процедуру и т.д. Результатом грамматического разбора будет цель грамматики или некоторое правило, если анализируется не вся цепочка.

Программа

```
<программа> ::= <заголовок программы>;  
               <блок>.  
  
<заголовок программы> ::= program<имя программы>  
<имя программы> ::= <идентификатор>  
  
<блок> ::= <список разделов описаний>;  
          <раздел операторов>.  
<список разделов описаний> ::= <раздел> /  
          <список разделов описаний>:  
          <раздел>  
<раздел> ::= <раздел меток> /  
          <раздел констант> /  
          <раздел типов> /  
          <раздел переменных> /  
          <раздел процедур и функций> /  
  
<раздел переменных> ::= var <список описаний переменных>  
<список описаний переменных> ::= <описание переменных> /  
          <список описаний переменных>;  
          <описание переменных>  
<описание переменных> ::= <список имен переменных> :  
          <тип>  
<список имен переменных> ::= <имя переменной> /  
          <список имен переменных>,  
          <имя переменной>  
<тип> ::= Integer / Real / Boolean / Char  
<имя переменной> ::= <идентификатор>
```

```

<раздел операторов> ::= begin
                        <список операторов>
                        end
<список операторов> ::= <оператор> /
                        <список операторов>;
<оператор> ::= <оператор>
<оператор> ::= <оператор присваивания> /
<оператор> ::= <составной оператор> /
<оператор> ::= <оператор выбора> /
<оператор> ::= <оператор цикла> /
<оператор> ::= <оператор перехода> /
<оператор> ::= <оператор присоединения> /
<оператор> ::= <оператор процедуры>

```

Операторы

```

<оператор присваивания> ::= < левая часть >:=< правая часть >
< левая часть> ::= < имя переменной>
< правая часть> ::= <выражение>
<выражение> ::= <терм> /
+<терм> /
-<терм> /
not <терм> /
<выражение>
<аддитивная операция>
<терм >
<терм> ::= <множитель> /
<простое выражение>
<мультипликативная операция>
<множитель>
<множитель> ::= <переменная> /
<константа без знака> /
(<выражение>) /
<переменная> ::= <имя переменной>
<константа без знака> ::= <число без знака>
<аддитивная операция> ::= + / - / or
<мультипликативная операция> ::= * / / div / mod / and

<составной оператор> ::= begin <список операторов> end

<блок> ::= <список разделов описаний><составной оператор>

<оператор выбора>:= <условный оператор> /
<оператор варианта>

<условный оператор> ::= if <условие> then <оператор>
else <оператор> /
if <условие> then <оператор>

<условие> ::= <выражение> /
<выражение><операция отношения>
<выражение>

<оператор варианта> ::= case <выражение> of <список ветвлений>
end

<список ветвлений> ::= <вариант> /
<список ветвлений>;<вариант>

<вариант> ::= <список меток > : <оператор>
<список меток > ::= <метка > /<список меток >,<метка>
<метка> ::= <константа скалярного типа >

```

<оператор повтора>	::= <оператор for > / <оператор repeat > / <оператор while >
<оператор <i>for</i> >	::= for <параметр цикла> := <список цикла> do <оператор>
<список цикла>	::= <начальное значение> to <конечное значение> <начальное значение> downto <конечное значение>
<начальное значение>	::= <выражение>
<конечное значение>	::= <выражение>
<параметр цикла>	::= <имя>
<оператор <i>repeat</i> >	::= repeat <список операторов> until <условие>
<оператор <i>while</i> >	::= while <условие> do <оператор>
<оператор <i>goto</i> >	::= goto <метка>
<метка>	::= <целое без знака>
<помеченный оператор>	::= <метка> : <оператор>
<раздел меток>	::= label <список меток>
<список меток>	::= <метка> / <список меток>, <метка>

Концепция типов

```

<раздел типов> ::= type <список описаний типов>
<список описаний типов> ::= <описание типа> /
<список описаний типов>;
<описание типа>
<описание типа> ::= <имя типа>=<тип>
<имя типа> ::= <идентификатор>
<тип> ::= <простой тип> /
<простой структурный тип> /
<ссылочный тип>

<простой тип> ::= <перечисляемый тип> /
<предопределенный тип> /
<ограниченный тип>

<перечисляемый тип> ::= (<список значений>)
<список значений> ::= <значение> /
<список значений>,<значение>
<значение> ::= <идентификатор> /
<целое >

<предопределенный тип> ::= Integer / Real / Boolean / Char

<ограниченный тип> ::= <минимальное значение> ..
<максимальное значение>
<минимальное значение> ::= <константа базового типа>
<максимальное значение> ::= <константа базового типа>

<предопределенный тип> ::= <подмножество целых типов > /
<подмножество вещественных типов > /
<подмножество булевских типов > / char

```

```

<раздел констант> ::= const<список описаний констант>
<список описаний констант> ::= <описание константы> /
<список описаний констант>;
<описание константы>
<описание константы> ::= <имя>=<константа> /
<константа> ::= <число> / <имя константы> / <строка>

<условие> ::= <выражение> /
<выражение><операция отношения> <выражение>

<арифметическое выражение> ::= <терм> /
+<терм> /
-<терм> /
<арифметическое выражение>
<аддитивная операция>
<терм>
<терм> ::= <операнд> /
<терм>
<мультипликативная операция>
<операнд>
<операнд> ::= <переменная> /
<константа без знака> /
(<арифметическое выражение>)
<переменная> ::= <имя переменной>
<константа без знака> ::= <число без знака>
<аддитивная операция> ::= + / -
<мультипликативная операция> ::= * / div / mod

<логическое выражение> ::= <логический терм> /
not <логический терм> /
<логическое выражение> or
<логический терм>
<логический терм> ::= <логический множитель> /
<логический терм> and
<логический множитель>
<логический множитель> ::= <логическая переменная> /
(<логическое выражение>)
<логическая переменная> ::= <имя переменной> /
(<арифметическое выражение>
<операция отношения>
<арифметическое выражение>)

<выражение> ::= <арифметическое выражение> /
<логическое выражение>
<условие> ::= <логическое выражение>

<простой структурированный тип> ::= <тип array> /
<тип record> /
<тип set> /
<тип file>

<тип array> ::= array [<тип индексов>] of <тип компонент>
<тип компонент> ::= <тип>

<тип record> ::= record <описание полей> end
<описание полей> ::= <фиксированная часть> /
<вариантная часть> /
<фиксированная часть>;
<вариантная часть>
<фиксированная часть> ::= <список полей>

```

<список полей> ::= <секция записи> /
 <секция записи> ::= <список полей> ; <секция записи>
 <список имен полей> ::= <список имен полей> : <тип>
 ::= <имя поля> /
 <список имен полей>, <имя поля>

 <вариантная часть> ::= case <поле признака> : <скалярный тип>
 of <список вариантов>
 <список вариантов> ::= <вариант> /
 <список вариантов>; <вариант>
 <вариант> ::= <список меток варианта>:
 (<список полей>) / <пусто>
 <список меток варианта> ::= <метка варианта> /
 <список меток варианта>,
 <метка варианта>
 <метка варианта> ::= <значение скалярного типа>
 <поле признака> ::= <имя> / <пусто>

 <оператор присоединения> ::= **with** <список имен полей> **do**
 <оператор>
 <список имен полей> ::= <имя поля> /
 <список имен полей>, <имя поля>

Процедуры

<процедура> ::= <заголовок процедуры>;
 <блок>
 <заголовок процедуры> ::= **procedure** <имя процедуры> /
procedure <имя процедуры>
 (<список формальных параметров>)
 <блок> ::= <список разделов описаний>:
 <раздел операторов>
 <список разделов описаний> ::= <раздел> /
 <список разделов описаний>:
 <раздел>
 <раздел> ::= <раздел меток> /
 <раздел констант> /
 <раздел типов> /
 <раздел переменных> /
 <раздел процедур и функций> /
 <раздел операторов> ::= **begin** <список операторов> **end**
 <список операторов> ::= <оператор> /
 <список операторов>; <оператор>

 <раздел процедур и функций> ::= <процедура или функция> /
 <раздел процедур и функций>;
 <процедура или функция>
 <процедура или функция> ::= <процедура> /
 <функция>

 <оператор процедуры> ::= <имя процедуры> /
 <имя процедуры>
 (<список фактических параметров>

 <список формальных параметров> ::= <формальный параметр> /
 <список формальных параметров>;
 <формальный параметр>
 <формальный параметр> ::= <параметр-переменная> /
 <параметр-значение> /
 <параметр-процедура> /
 <параметр-функция>
 <параметр-переменная> ::= **var** <список имен> : <тип>

<параметр-значение> ::= <список имен>:<тип>
 <параметр-процедура> ::= **procedure**<имя процедуры>
 <параметр-функция> ::= **function**<имя функции>:<тип>
 <список имен> ::= <имя переменной> /
 <список имен>,
 <имя переменной>

<заголовок функции > ::= **function** <имя>:<тип результата> /
function <имя>
 (<список формальных параметров>):
 <тип результата>
 <тип результата> ::= <простой тип> / <ссылочный тип>
 <список операторов> ::= <оператор> / <список операторов>;
 <оператор>.

Модули

<модуль > ::= <заголовок модуля >
 <интерфейсная секция>
 <секция реализации>
 <секция инициализации >
 <заголовок модуля> ::= **unit**! <имя модуля >
 <имя модуля > ::= <идентификатор>
 <предложение использования > ::= **uses**<список имен модулей>
 <список имен модулей> ::= <имя модуля> /
 <список имен модулей>,
 <имя модуля>
 <интерфейсная секция> ::= **Interface** <список разделов> /
Interface
 <предложение использования >
 <список разделов>
 <список разделов> ::= <раздел > /
 <список разделов>;<раздел>
 <раздел> ::= <раздел констант> /
 <раздел типов > /
 <раздел переменных> /
 <раздел заголовков>
 <раздел заголовков> ::= <заголовок> /
 <раздел заголовков>;<заголовок>
 <заголовок> ::= <заголовок процедуры> /
 <заголовок функции>
 <Секция реализации> ::= **Implementation**
 <список разделов описаний> /
Implementation
 <предложение использования>
 <список разделов описаний>
 <секция инициализации> ::= **end** /
 <составной оператор>

Операции с файлами

<имя физического файла> ::= <имя устройства>:\<имя файла> /
 <имя устройства>:\<путь>\<имя файла>
 <путь> ::= <имя подкаталога> /
 <путь>\<имя подкаталога>
 <имя файла> ::= <имя> .<расширение имени>

<расширение имени>	::= <идентификатор>
<имя устройства>	::= <идентификатор>
<имя подкаталога>	::= <идентификатор>

Объектно-ориентированное программирование

<объект>	::= object <список полей>;<список методов>; end / object (<имя предка>) <список полей>;<список методов>; end
<список методов>	::= <метод>; / <список методов>; <метод>;
<метод>	::= <заголовок процедуры> / <заголовок функции>
<заголовок метода>	::= <имя объекта>.<заголовок процедуры>; / <имя объекта>.<заголовок функции>