Расширенная БНФ и примеры в С

Расширенная форма Бэкуса-Наура (БНФ) состоит из ряда следующих обозначений и правил:

- символы в кавычках переносятся в конструкцию языка так, как они записаны. Кавычки при этом отбрасываются. Например, "while" означает, что в конструкции языка присутствует while;
- имена, записанные слитно русскими и латинскими буквами, обозначают различные конструкции языка. Например, оператор_цикла;
- квадратные скобки охватывают элементы языка, которые могут повторяться 0 или 1 раз. Например, "AB"["C"] означает, что в конструкции языка может присутствовать или AB или ABC;
- фигурные скобки охватывают элементы языка, которые могут повторяться 0 или много раз. Например, "AB" { "C" } означает, что в конструкции языка может присутствовать или AB, или ABC, или ABCC и т.д.;
- символ | обозначает или, то есть используется для задания альтернативных значений, из списка элементов, разделенных знаком |. Например, "AB"|"C"|"ff" означает, что в конструкции языка может присутствовать или AB или C или ff;
- круглые скобки используются для группировки. Например, "A"("B"|"C")"D" означает, что в конструкции языка может присутствовать или ABD или ACD;
- многоточие используется для обозначения очевидных пропущенных значений в перечислении;
- символ = обозначает - слово есть. Например, буква = "A"|"B"|"C".

1. Идентификаторы языка Си

```
имя = ( буква | "_" ) { буква | цифра | "_" }
          буква = |"A"|"B"|...|"Y"|"Z"|"a"|"b"|...|"y"|"z"
          цифра = "0"|"1"|...|"9"
```

1.1. Основные типы данных языка Си

```
описание_типа = ["const "] имя_типа " " имя ["=" константа]
                  { "," имя ["=" константа] } ";"
имя_типа = "int" | "short" | "long" | "char" | "float" | "double"
```

1.2. Правила записи констант различных типов

```
целая_константа =( десятичная | восьмеричная | шестнадцатеричная ) ["l"|"L"]

    десятичная         =     цифра {цифра}
    восьмеричная       = "0" цифра8 {цифра8}
    шестнадцатеричная = "0" ( "x" | "X" ) цифра16 {цифра16}
    цифра8             = "0"|"1"|...|"7"
    цифра16            = "0"|"1"|...|"9"|"A"|"B"|"C"|"D"|"E"|"F"

    вещ_константа = цифра {цифра} "." {цифра} [ "e" [ "+" | "-" ] цифра {цифра} ]
```

1.3. Беззнаковый тип для целых данных

```
имя_беззнакового_типа = "unsigned "("int"|"long"|"short"|"char" )
```

1.4. Символьные строки

```
описание_массива_символов = "char " имя "["размер"]" { "," имя "["размер"]" } ";"
```

2. Понятие функции

```
вызов_функции = имя_ функции "(" [ аргумент { "," аргумент } ] ")"
```

2.1. Стандартная функция printf

```
"printf" "(" формат { "," аргумент } ")"
модификатор = ["-"] {цифра1} [ "." {цифра2} ][l]
```

2.2. Стандартная функция scanf

```
"scanf" "(" формат { "," аргумент } ")"
```

3. Операции и выражения

3.1. Оператор-выражение

```
оператор = выражение ";"
```

3.2. Операции преобразования типов

```
преобразование_типа = "("имя_типа ")" выражение
```

3.3. Простейшие функции, определяемые программистом

```
заголовок_функции = тип имя_функции"("[тип параметр {"," тип параметр}]")"
"return " [выражение];
```

3.4. Операция определения размера данных

```
"sizeof(" тип ")"
"sizeof " имя_данного
```

4. Операторы языка Си и приемы программирования

```
составной_оператор = "{" оператор { оператор } "}"
```

4.1. Оператор цикла while

```
"while" "(" выражение ")" оператор
```

4.2. Условный оператор и условная операция

```
условный_оператор = "if" "("выражение")" оператор_1 ["else" оператор_2]
условное_операция = выр_0 "?" выр_1 ":" выр_2
```

4.3. Множественный выбор. Оператор переключения

```
оператор_переключения =
            "switch" "(" выражение ")"
            "{"
                  "case" константа ":" { оператор }
```

```
                "case" константа ":" { оператор }
              [ "default"        ":" { оператор } ]
          "}"
```

4.4. Оператор цикла do-while

```
цикл_do-while = "do" оператор "while" "("выражение")" ";"
```

4.5. Перечисления

```
перечисление =
             "enum" [ имя_перечисления ]
                  "{"
                            имя_конст [ "=" конст_выр ]
                        { "," имя_конст [ "=" конст_выр ] }
                  "};"
```