

Алгоритмизация и программирование

ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ 5. РЕКУРСИЯ. ПОЯСНЕНИЯ

•

Форма Бэкуса-Наура - способ задания синтаксиса современных языков программирования

Разработан для описания Алгола-60.

При записи грамматики в форме Бэкуса-Наура используются два типа объектов:

- основные символы (или *терминальные символы*, в частности, ключевые слова)
- металингвистические переменные (или *нетерминальные символы*), значениями которых являются цепочки основных символов описываемого языка. Металингвистические переменные изображаются словами (русскими или английскими), заключенными в угловые скобки (*<...>*)
- металингвистические связки (*::=, |*)

Форма Бэкуса-Наура - способ задания синтаксиса современных языков программирования .

$\langle \text{целое} \rangle ::= \langle \text{целое без знака} \rangle \mid + \langle \text{целое без знака} \rangle \mid - \langle \text{целое без знака} \rangle$

$\langle \text{целое без знака} \rangle ::= \langle \text{цифра} \rangle \mid \langle \text{цифра} \rangle \langle \text{целое без знака} \rangle$

$\langle \text{цифра} \rangle ::= 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$

1) 123

2) *12

3) 1.2

4) -1

5) +12.3

Задание

Понятие вещественное определено следующим образом:

$\langle \text{вещественное} \rangle ::= \langle \text{мантисса} \rangle \langle \text{порядок} \rangle \mid \langle \text{знак} \rangle \langle \text{мантисса} \rangle \langle \text{порядок} \rangle$

$\langle \text{мантисса} \rangle ::= \text{.} \langle \text{целое без знака} \rangle \mid \langle \text{целое без знака} \rangle \text{.} \langle \text{целое без знака} \rangle$

$\langle \text{порядок} \rangle ::= \text{E} \langle \text{знак} \rangle \langle \text{целое без знака} \rangle$

$\langle \text{целое без знака} \rangle ::= \langle \text{цифра} \rangle \mid \langle \text{цифра} \rangle \langle \text{целое без знака} \rangle$

$\langle \text{цифра} \rangle ::= 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$

$\langle \text{знак} \rangle ::= + \mid -$

- 1) 1 – false
- 2) +1 – false
- 3) .5 - false
- 4) 5.5a – false
- 5) 5.5e+2 - false
- 6) 5.5E2 – false
- 7) 5.5E+2 – true
- 8) -5.123E-123 – true
- 9) 12.3E-12a - false

•

```
// <знак> ::= + | -
```

```
bool isSign (char * str)
{  if (*str == '+' || *str == '-') {
    str++;
    return true;
  }
  return false;
}
```

```
bool isSign (const char *& str) ...
```

•

//<вещественное> ::= <мантисса> <порядок> | <знак> <мантисса> <порядок>

```
bool isRealNumber (const char *& str)
{
    isSign(str);
    if (isMantissa(str) && isExhibitor(str) && *str == '\0') {
        return true;
    } else {
        return false;
    }
}
```

•

//<вещественное> ::= <мантисса> <порядок> | <знак> <мантисса> <порядок>

```
bool isRealNumber(const char *& str)
{
    isSign(str);
    if (isMantissa(str) && isExhibitor(str) && *str == '\\0') {
        return true;
    } else {
        return false;
    }
}
return isMantissa(str) && isExhibitor(str) && *str == '\\0';
```

```

// <вещественное> ::= <мантисса><порядок>
bool isRealNumber(const char **expression)
{
    if (isMantissa(expression)) {
        if (isExhibitor(expression)) {
            return (**expression == '\0'); // вернуть true, если число состоит
                                            // только из мантиссы и экспоненты
                                            //(после экспоненты больше ничего нет)
        }
    } else {
        return false;
    }
}

```


int main()

{

// 1 - false

char* str = "1";

if (isRealNumber(str) == false) {

std::cout << "test - ok"<< '\n';

}

else {

std::cout << "test - NOT ok"<< '\n';

}

+1 - false

