Лекция 1. Строки в Delphi.

В Delphi есть три группы строк:

Типы строк (Строка - последовательность символов)

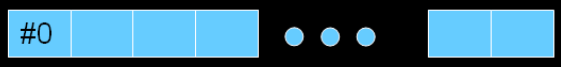
- Статические короткие строки
  - » ShortString, String[40]
- Динамические длинные строки
  - » ANSIString, WideString
- Строки с завершающим #0-символом (для совместимости с функциями, написанными на Си)
  - » PChar, PANSIChar, PWideChar

Статические короткие строки

- Не более 255 символов
  - » Var s1: ShortString; // 255 символов + 1

0 1 2 255

» Var s2: String[40]; // 40 символов + 1



0 1 2 40

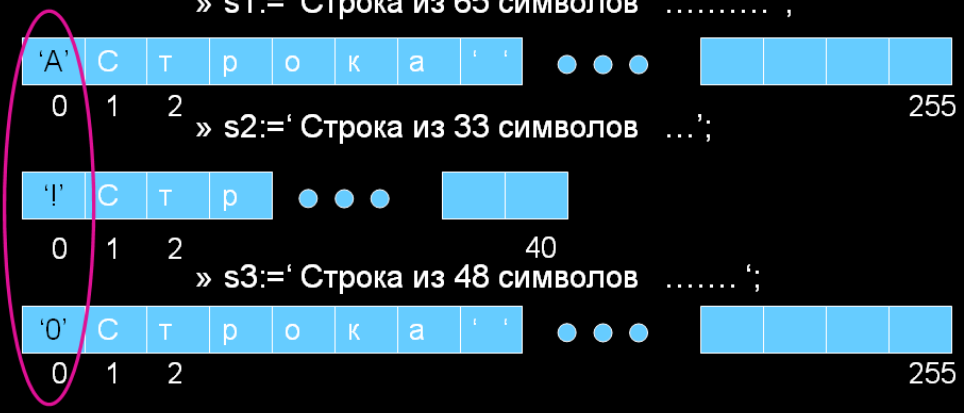
» {$H-} Var s3: String; // =ShortString при {$H-}

0 1 2 255

Статические короткие строки

- Нулевой символ – длина строки

» s1:=‘ Строка из 65 символов …… ‘;



» <Длина> :=Length(s2) либо <Длина> := ORD(s2[0])

Динамические длинные строки

- Тип с управляемым временем жизни (Скрытый указатель, к которому применяется технология “сборки мусора”)

NiL

NiL

» {$H+} Var s4: String; // =ANSIString при {$H+}

» Var s5: ANSIString;

КУЧА

Динамические длинные строки

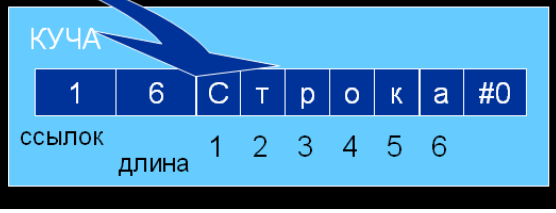
- Тип с управляемым временем жизни (Скрытый указатель, к которому применяется технология “сборки мусора”)

» {$H+} Var s4: String; // =ANSIString при {$H+}

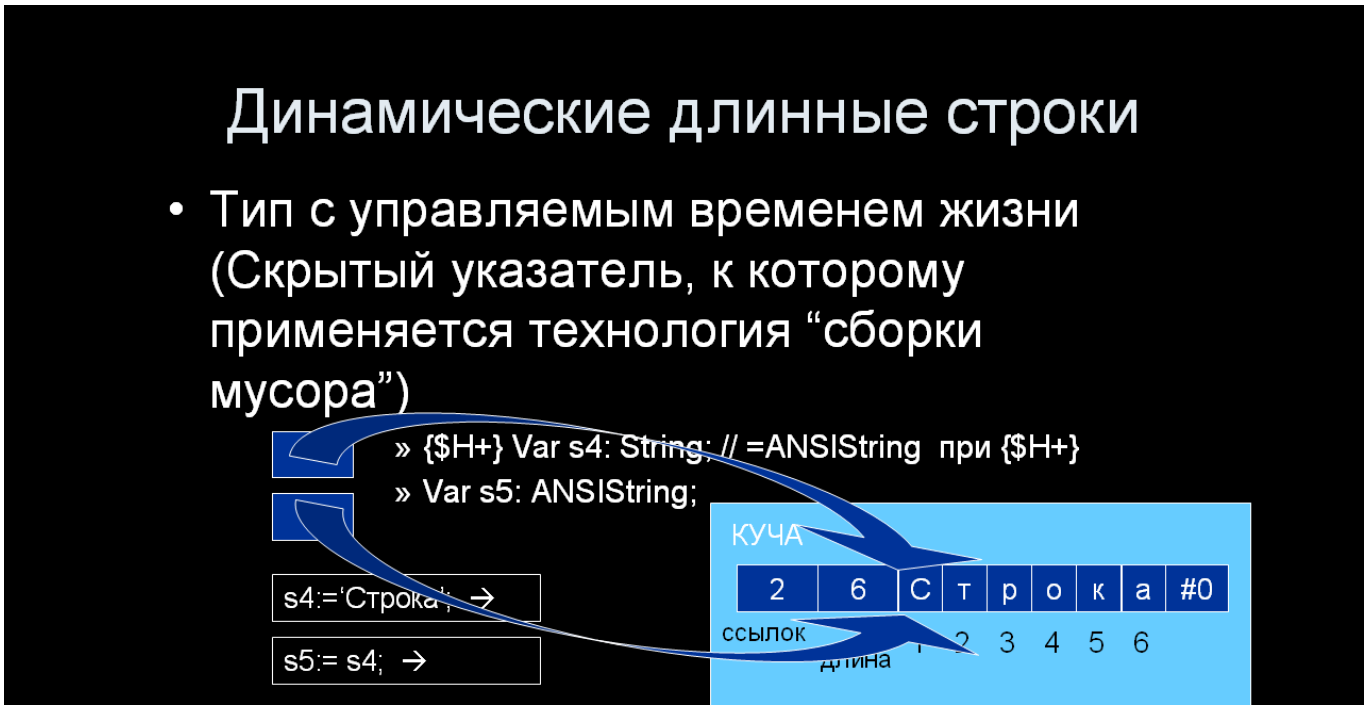
» Var s5: ANSIString;

NiL

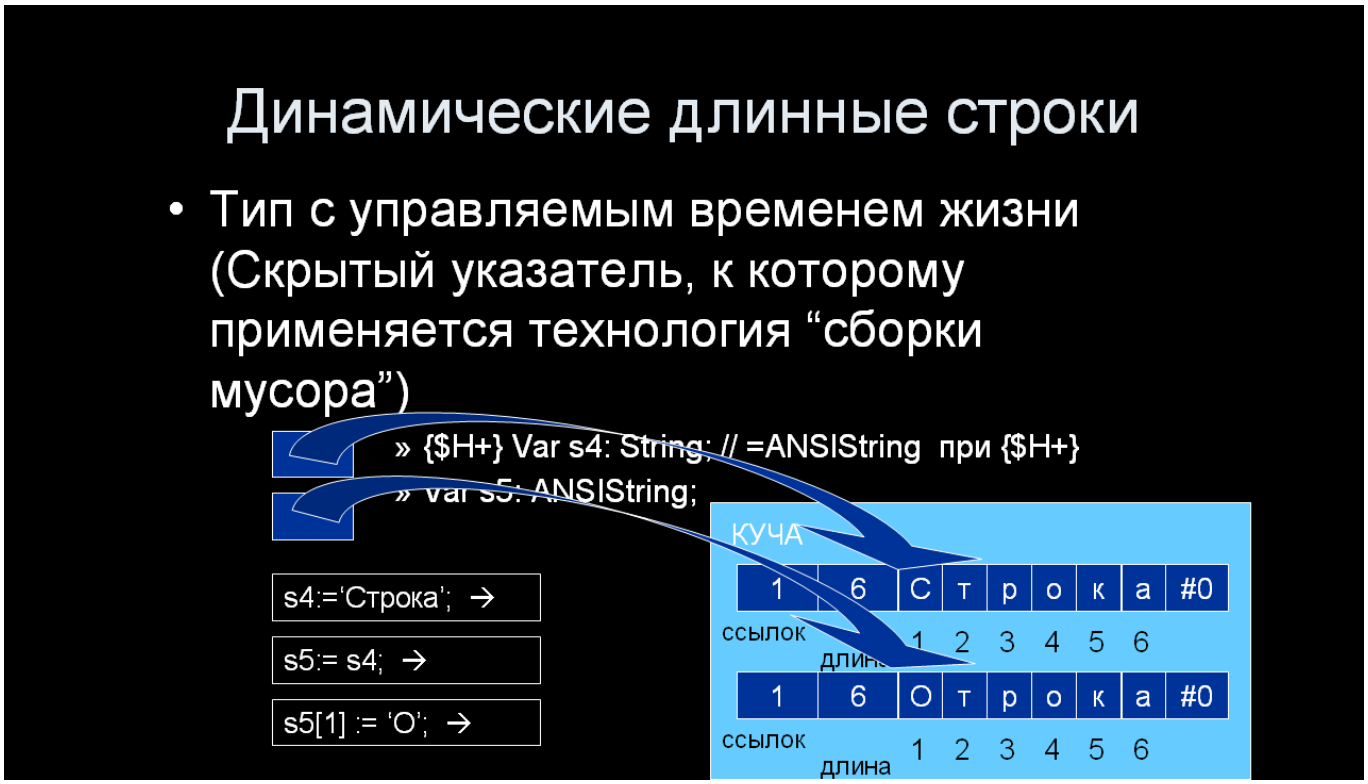
s4:=‘Строка’; →



При присваивании s5:=s4 строка не копируется, а увеличивается счетчик ссылок::



Но при изменении строки s5, происходит их разделение:

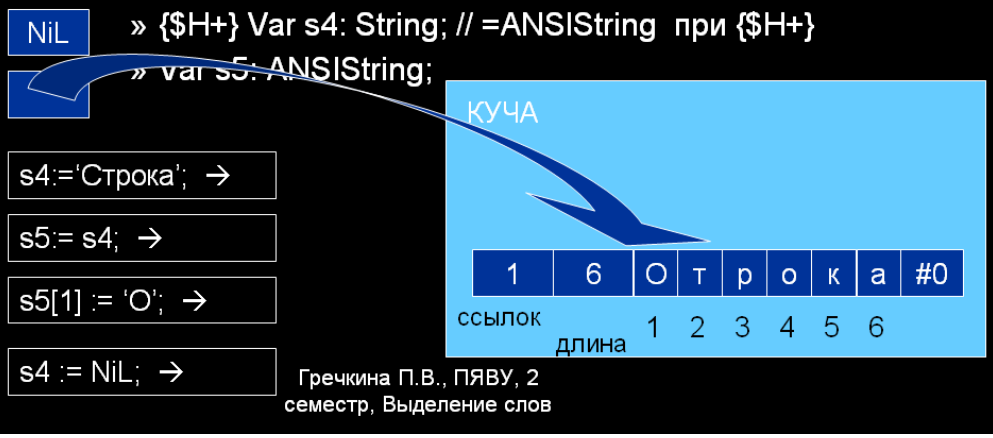


Для освобождения памяти, выделенной для строки (если счетчик ссылок =1) достаточно присвоить строке значение Nil (нулевой указатель), Например, s4:=Nil;

При этом во время трансляции в код будет включен код освобождения памяти (технология «сборки мусора»):

Динамические длинные строки

- Тип с управляемым временем жизни (Скрытый указатель, к которому применяется технология “сборки мусора”)



Гречкина П.В., ПЯВУ, 2 семестр, Выделение слов

– Var s6: WideString; тоже, но из символов WideChar (2 байта на символ, 16-битный Юникод)

На следующем слайде наглядно видны отличия трех групп строк:

Типы строк (Строка - последовательность символов)

- Статические короткие строки
  - ShortString, String[40]
- Динамические длинные строки
  - ANSIString, String
- Строки с завершающим #0-символом
  - PChar, PANSIChar, PWideChar

Гречкина П.В., Программирование, семестр2, Строки в Delphi

Действия над строками

- Посимвольное обращение по индексу с 1
  s[1]
- Сравнение строк
  'ABCD' < 'ABD' 'AB' < 'ABC'
- Присваивание
  s1:=''; s2:=' '; s3:= 'Строка';
  s4:=nil; setlength(s, 7);// динамическая
- Конкатенация строк s1:='Ещё' + ' слово';
  s1:=concat('Ещё',' слово');

Процедуры и функции для работы со строками Delphi

Length(строка) → Длина строки = для статич. Ord(строка[0])

Pos(подстрока, строка) → Номер первого вхождения

PosEx(подстрока, строка, откуда начать) → Номер первого вхождения, начиная с указанного места

Copy(строка, откуда, сколько) → Часть строки

Delete(строка, откуда, сколько); Удаляет часть строки

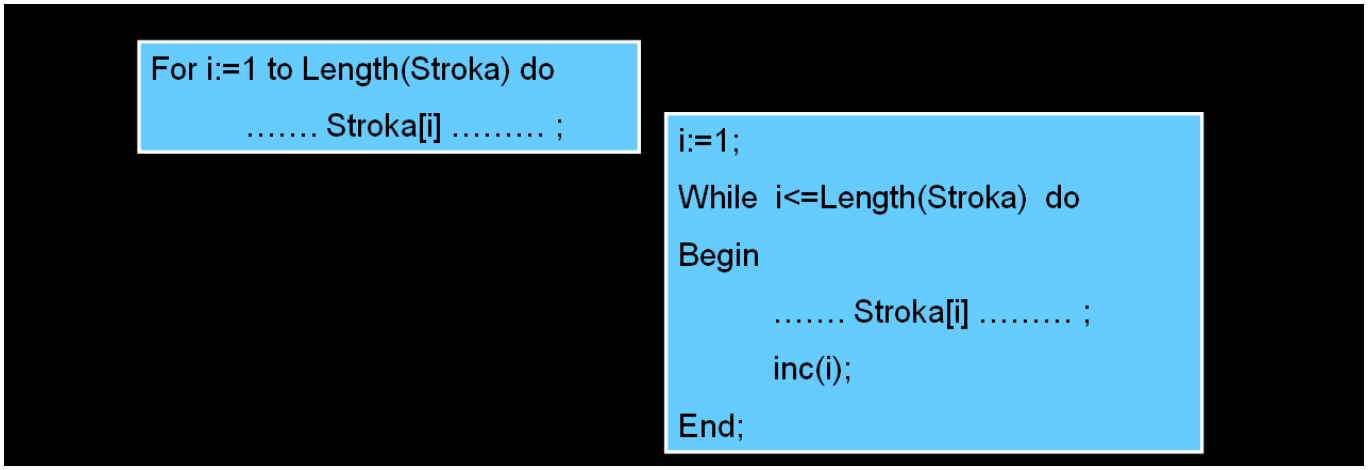
Insert(подстрока, строка, куда) → Вставляет подстроку в строку с указанной позиции, «раздвигая» символы

Изменение регистра ANSILowerCase, ANSIUpperCase, UpperCase, LowerCase, UpCase…
'a' > 'Z' ANSIUpperCase('a') < ANSIUpperCase('Z')

Есть и другие - Trim, TrimLeft, Str, Val, IntToStr, StrToFloatF, …

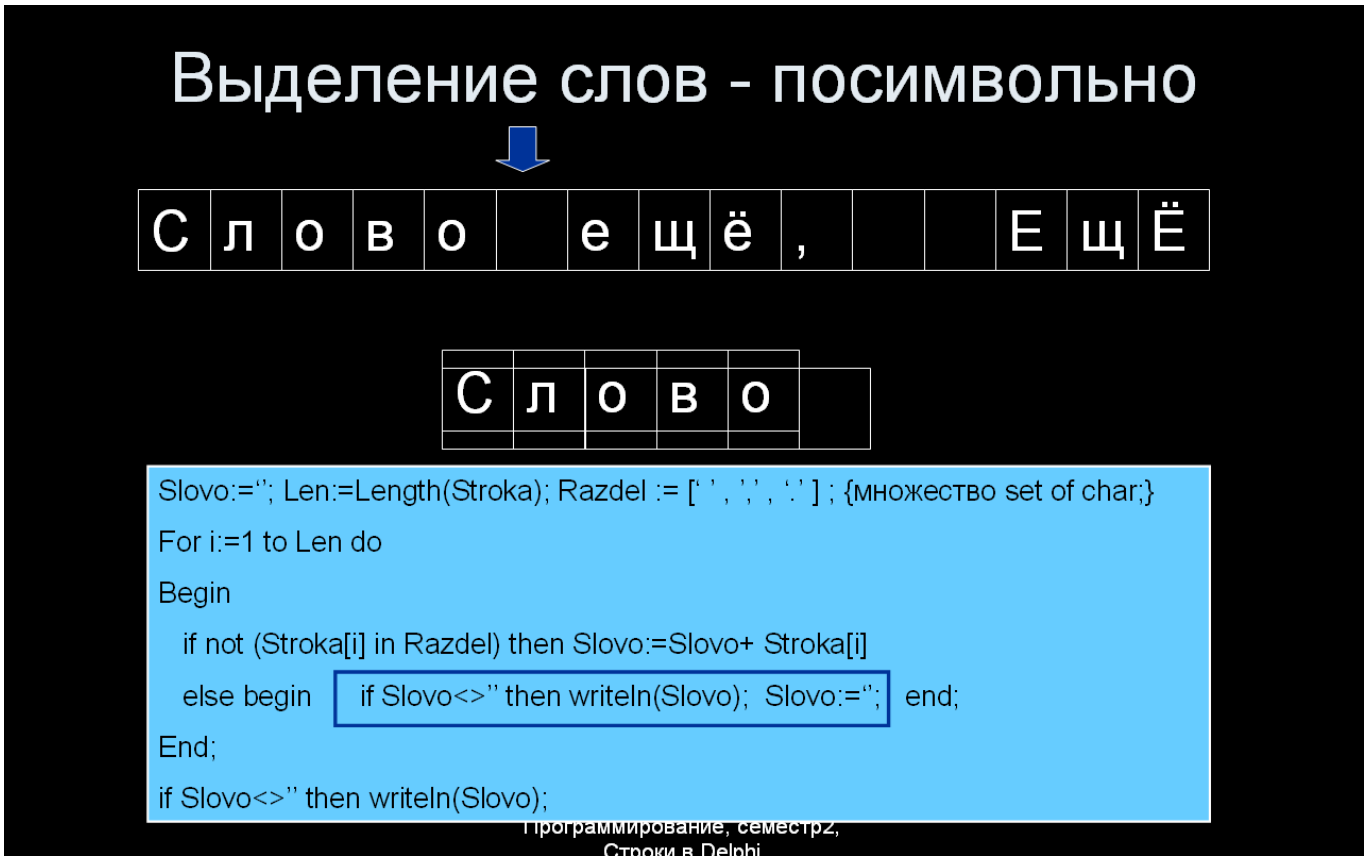
Работать со строкой можно посимвольно, как с массивом.

Если длина строки НЕ изменяется в процессе , то подойдет цикл for, иначе – While^

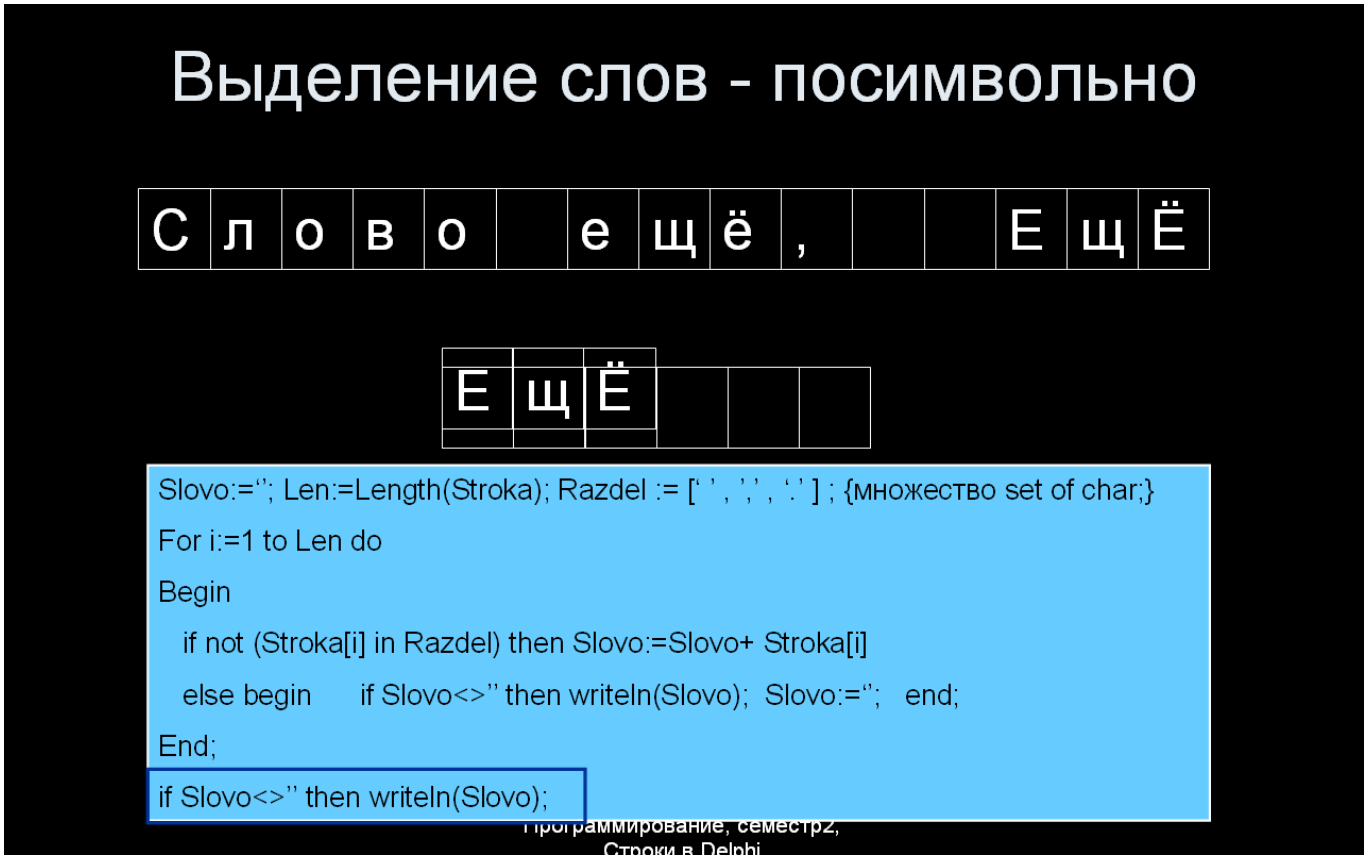


Например, рассмотрим задачу выделения слов из строки. В качестве разделителей возьмем пробел, запятую и точку.

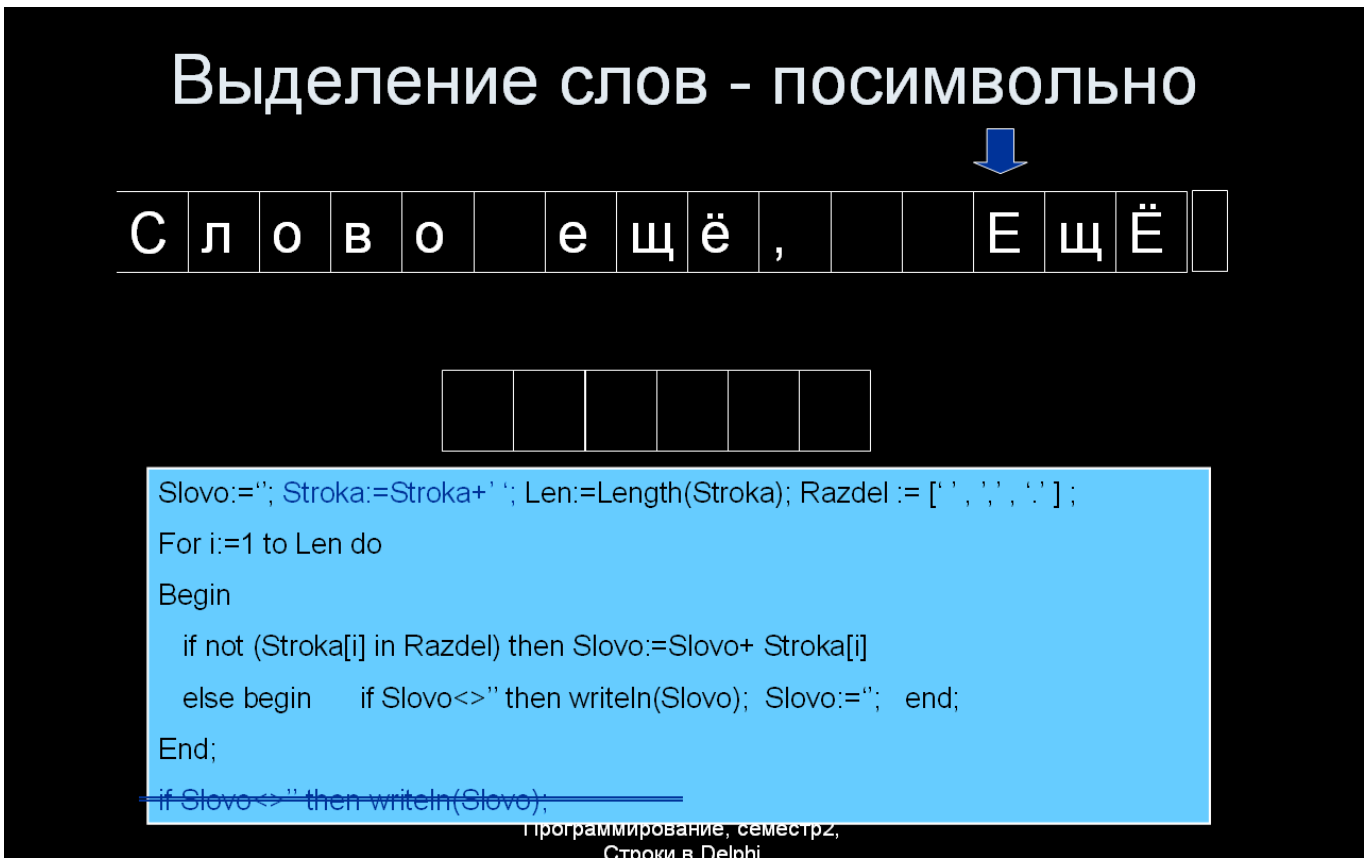
Пока не встретился разделитель, собираем буквы в слово, а когда дойдем до разделителя, выводим найденное слово (если оно не пустое, при двух разделителях подряд или разделителе в начале строки) и опустошаем слово:



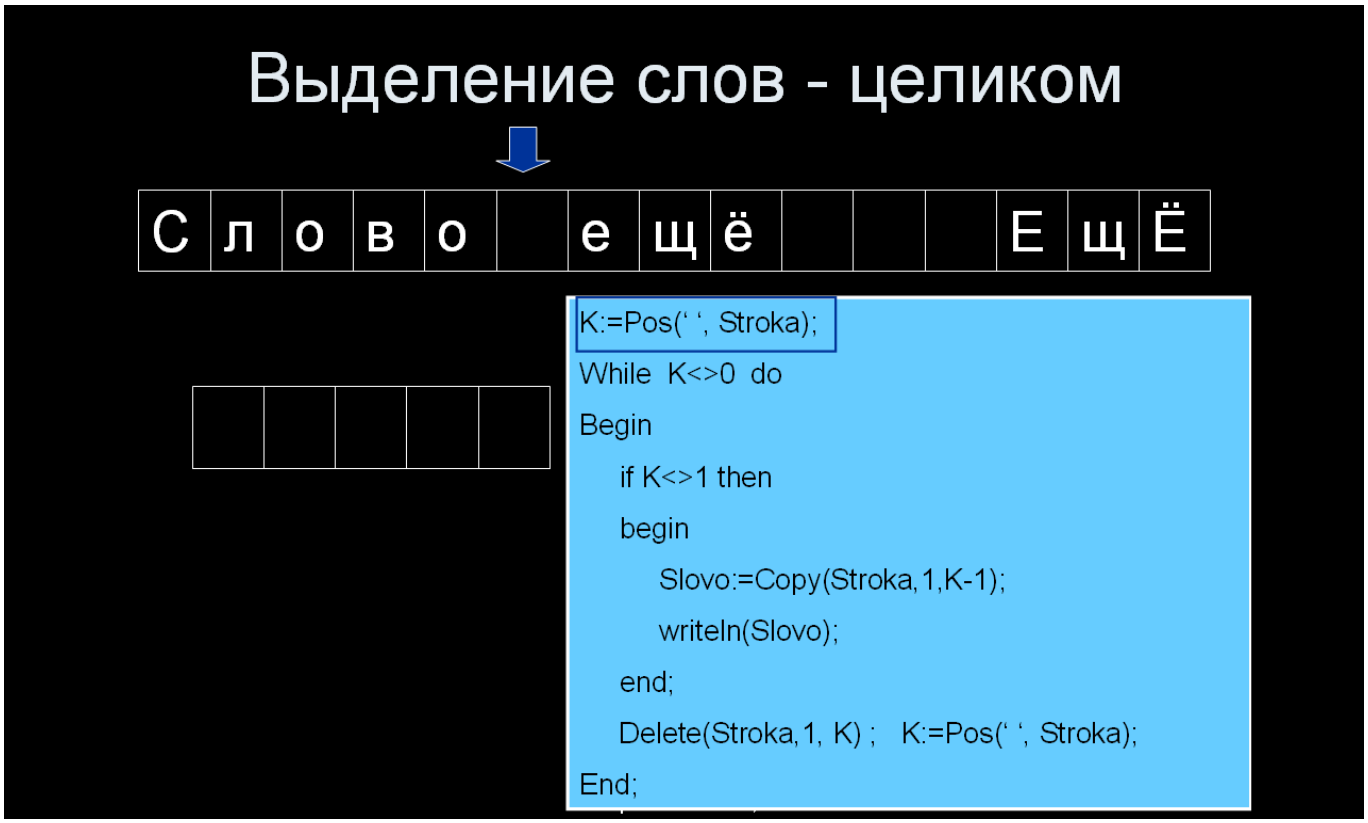
Если мы дошли до конца строки, и в конце нет разделителя, то последнее слово останется не выведенным: Для этого после цикла есть проверка:



Можно с последним словом поступить иначе, добавив разделитель к исходной строке в самом начале (хотя он может не уместится в строке, если она полностью заполнена…)

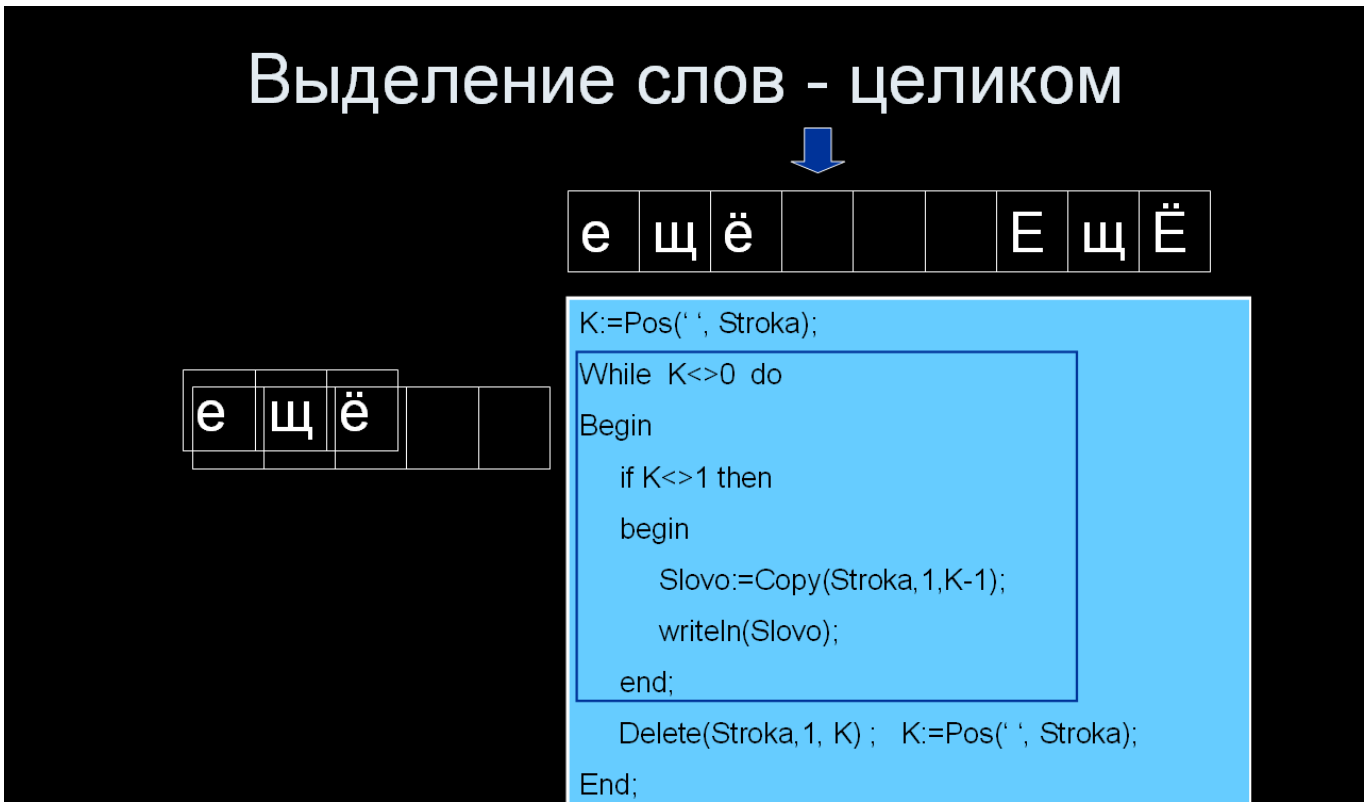


Слова можно выделять с помощью процедур работы со строками:



Но это удобно делать в Delphi только если разделитель один единственный, например, пробел, позицию первого разделителя находим с помощью функции Pos и выделяем слово до него.

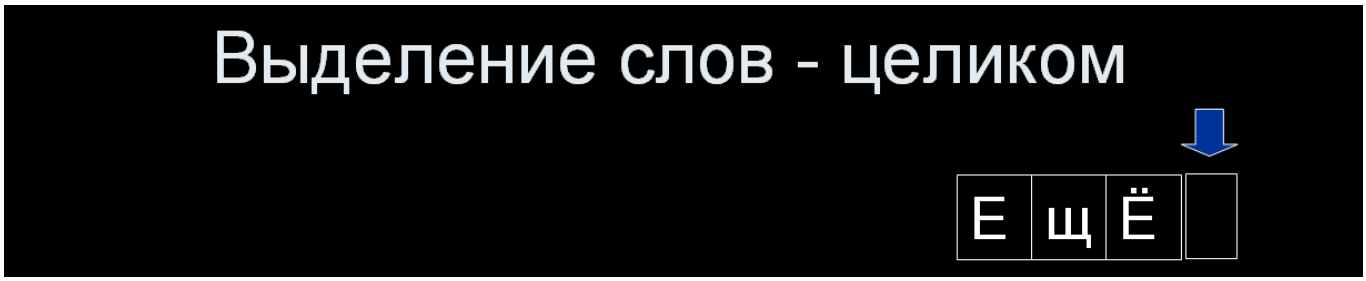
После вывода слова, удаляем его вместе с разделителем из строки. И снова ищем слово, пока K не станет равным нулю, то есть разделители в строке не закончатся.



Когда разделители закончатся, строка будет пустой или содержать последнее слово.

Выводим его:

Или опять же,



можно добавить с самого начала в конец строки разделитель:

После изучения динамических открытых массивов (ещё одного типа с управляемым временем жизни) будет рассмотрены особенности передачи динамических строк и динамических массивов в качестве параметров процедур.

Правила передачи статических строк, такие же как у статических массивов, и других пользовательских типов данных. изученных нами в прошлом семестре. Единственное исключение - тип *ShortString* - это базовый тип..