**Строковые типы**

Строковый тип данных - один из самых часто используемых в программах тип. Действительно, без него не обходится практически ни один алгоритм. Даже программы, выполняющие исключительно математические операции, порой, написаны с использованием строковых типов данных.  
  
Юникод-символы - многопользовательские сервера, мультиязыковые приложения  
  
Для большинства целей подходит тип AnsiString (иногда называется Long String).  
  
Стандартные функции обработки строк:  
  
1) Функция Length(Str: String) - возвращает длину строки (количество символов). Пример:

|  |  |
| --- | --- |
| 1  2  3  4  5 | var     Str: String; L: Integer;  { ... }  Str:='Hello!';  L:=Length(Str);  { L = 6 } |

2) Функция SetLength(Str: String; NewLength: Integer) позволяет изменить длину строки. Если строка содержала большее количество символов, чем задано в функции, то "лишние" символы обрезаются. Пример:

|  |  |
| --- | --- |
| 1  2  3  4 | var Str: String;  { ... }  Str:='Hello, world!';  SetLength(Str, 5); { Str = "Hello" } |

3) Функция Pos(SubStr, Str: String) - возвращает позицию подстроки в строке. Нумерация символов начинается с единицы (1). В случае отсутствия подстроки в строке возращается 0. Пример:

|  |  |
| --- | --- |
| 1  2  3  4  5 | var Str1, Str2: String; P: Integer;  { ... }  Str1:='Hi! How do you do?';  Str2:='do';  P:=Pos(Str2, Str1);  { P = 9 } |

4) Функция Copy(Str: String; Start, Length: Integer) - возвращает часть строки Str, начиная с символа Start длиной Length. Ограничений на Length нет - если оно превышает количество символов от Start до конца строки, то строка будет скопирована до конца. Пример:

|  |  |
| --- | --- |
| 1  2  3  4 | var Str1, Str2: String;  { ... }  Str1:='This is a test for Copy() function.';  Str2:=Copy(Str1, 11, 4); { Str2 = "test" } |

5) Процедура Delete(Str: String; Start, Length: Integer) - удаляет из строки Str символы, начиная с позиции Start длиной Length. Пример:

|  |  |
| --- | --- |
| 1  2  3  4 | var Str1: String;  { ... }  Str1:='Hello, world!';  Delete(Str1, 6, 7); { Str1 = "Hello!" } |

6) Процедура Insert(SubStr: String; Str: String; Pos: Integer) - вставляет в строку Str подстроку SubStr в позицию Pos. Пример:

|  |  |
| --- | --- |
| 1  2  3  4 | var Str: String;  { ... }  Str:='Hello, world!';  Insert('my ',Str, 8); { Str1 = "Hello, my world!" } |

7) Функции UpperCase(Str: String) и LowerCase(Str: String) преобразуют строку соответственно в верхний и нижний регистры:

|  |  |
| --- | --- |
| 1  2  3  4  5 | var Str1, Str2, Str3: String;  { ... }  Str1:='hELLo';  Str2:=UpperCase(Str1); { Str2 = "HELLO" }  Str3:=LowerCase(Str1); { Str3 = "hello" } |

Если строки полностью идентичны, логическое выражение станет равным True.  
  
Дополнительные функции обработки строк:  
  
В модуле StrUtils.pas содержатся полезные функции для обработки строковых переменных. Чтобы подключить этот модуль к программе, нужно добавить его имя (StrUtils) в раздел Uses.  
  
1) PosEx(SubStr, Str: String; Offset: Integer) - функция аналогична функции Pos(), но позволяет задать отступ от начала строки для поиска. Если значение Offset задано (оно не является обязательным), то поиск начинается с символа Offset в строке. Если Offset больше длины строки Str, то функция возратит 0. Также 0 возвращается, если подстрока не найдена в строке. Пример:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | uses StrUtils;  { ... }  var Str1, Str2: String; P1, P2: Integer;  { ... }  Str1:='Hello! How do you do?';  Str2:='do';  P1:=PosEx(Str2, Str1, 1); { P1 = 12 }  P2:=PosEx(Str2, Str1, 15); { P2 = 19 } |

2) Функция AnsiReplaceStr(Str, FromText, ToText: String) - производит замену выражения FromText на выражение ToText в строке Str. Поиск осуществляется с учётом регистра символов. Следует учитывать, что функция НЕ изменяет самой строки Str, а только возвращает строку с произведёнными заменами. Пример:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | uses StrUtils;  { ... }  var Str1, Str2, Str3, Str4: String;  { ... }  Str1:='ABCabcAaBbCc';  Str2:='abc';  Str3:='123';  Str4:=AnsiReplaceStr(Str1, Str2, Str3); { Str4 = "ABC123AaBbCc" } |

3) Функция AnsiReplaceText(Str, FromText, ToText: String) - выполняет то же самое действие, что и AnsiReplaceStr(), но с одним исключением - замена производится без учёта регистра. Пример:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | uses StrUtils;  { ... }  var Str1, Str2, Str3, Str4: String;  { ... }  Str1:='ABCabcAaBbCc';  Str2:='abc';  Str3:='123';  Str4:=AnsiReplaceText(Str1, Str2, Str3); { Str4 = "123123AaBbCc" } |

4) Функция DupeString(Str: String; Count: Integer) - возвращает строку, образовавшуюся из строки Str её копированием Count раз. Пример:

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | uses StrUtils;  { ... }  var Str1, Str2: String;  { ... }  Str1:='123';  Str2:=DupeString(Str1, 5); { Str2 = "123123123123123" } |

5) Функции ReverseString(Str: String) и AnsiReverseString(Str: AnsiString) - инвертируют строку, т.е. располагают её символы в обратном порядке. Пример:

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | uses StrUtils;  { ... }  var Str1: String;  { ... }  Str1:='0123456789';  Str1:=ReverseString(Str1); { Str1 = "9876543210" } |

6) Функция IfThen(Value: Boolean; ATrue, AFalse: String) - возвращает строку ATrue, если Value = True и строку AFalse если Value = False. Параметр AFalse является необязательным - в случае его отсутствия возвращается пустая строка.

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | uses StrUtils;  { ... }  var Str1, Str2: String;  { ... }  Str1:=IfThen(True, 'Yes'); { Str1 = "Yes" }  Str2:=IfThen(False, 'Yes', 'No'); { Str2 = "No" } |

Мы рассмотрели функции, позволяющие выполнять со строками практически любые манипуляции. Как правило, вместо строки с указанным типом данных, можно использовать и другой тип - всё воспринимается одинаково. Но иногда требуются преобразования. Например, многие методы компонент требуют параметр типа PChar, получить который можно из обычного типа String функцией PChar(Str: String):

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | uses ShellAPI;  { ... }  var FileName: String;  { ... }  FileName:='C:\WINDOWS\notepad.exe';  ShellExecute(0, 'open', PChar(FileName), '', '', SW\_SHOWNORMAL); |

Тип Char представляет собой один-единственный символ. Работать с ним можно как и со строковым типом. Для работы с символами также существует несколько функций:  
  
Chr(Code: Byte) - возвращает символ с указанным кодом (по стандарту ASCII):

|  |  |
| --- | --- |
| 1  2  3 | var A: Char;  { ... }  A:=Chr(69); { A = "E" } |

Ord(X: Ordinal) - возвращает код указанного символа, т.е. выполняет противоположное действие функции Chr():

|  |  |
| --- | --- |
| 1  2  3 | var X: Integer;  { ... }  X:=Ord('F'); { X = 70 } |

Из строки можно получить любой её символ - следует рассматривать строку как массив. Например:

|  |  |
| --- | --- |
| 1  2  3  4  5 | var Str, S: String; P: Char;  { ... }  Str:='Hello!';  S:=Str[2]; { S = "e" }  P:=Str[5]; { P = "o" } |

Список Литературы

<https://yadro.top/index.php?route=news/view&news_id=8079&page=7>