Лабораторная работа №6

Тема: Тип данных — строка

Задание: Обработать введенную пользователем строку в соответствии с индивидуальным заданием. Каждое задание обработки должно осуществляться при нажатии соответствующей кнопки. Вывод должен производиться с возможностью копирования и без потери введенной пользователем строки.

Строка — это просто некоторая конечная последовательность символов. Вполне правомерно рассматривать строку как массив символов, но этот подход не учитывает некоторые особенности строк. Прежде всего, значения строк, в отличие от значений массивов, часто используются повторно без изменений. Во-вторых, возможный диапазон значений элемента строки (символа), как правило, невелик. В-третьих, необходимость перестановки отдельных элементов (символов) в строке возникает исключительно редко — если строку отсортировать, скажем, по возрастанию кода символ, весь ее смысл скорее всего потеряется. При сортировке массивов смысл обычно не теряется (там не важны положения элементов относительно друг друга). В-четвертых, для строк вполне определена операция сложения, называемая в данном случае конкатенация.

По этим причинам во многих языках программирования строки выносятся в отдельный тип данных со специальной обработкой. Чаще всего к символам в строке можно обращаться как к элементам массива, например

Var

S: String;

Begin

…

S := ‘Вося’;

S[2] := ‘а’;

…

End;

При этом важно помнить, что индексация строк в языке Pascal всегда начинается с 1 (у динамических массивов с 0).

Также нужно помнить, что в FreePascal все строки по-умолчанию представлены в кодировке UTF-8, в которой на один символ может приходиться как один элемент, так и несколько. Для работы с отдельными символами в строке следует использовать строки в кодировке UTF-16 (тип WideString).

Для строк определены следующие основные операции:

1. Операция объединения (конкатенация) строк. Используется стандартная языковая операция сложения:

Var S1, S2: string;

S1 := S1 + S2 + ‘ конец’;

1. Операция вставки подстроки

Var S1: string;

S1 := ‘Вася=???’;

Insert(‘+Маша’, S1, 6);

1. Операция удаления подстроки

Var S1: string;

S1 := ‘Вася – нехороший человек’;

Delete(S1, 8, 2);

1. Операция поиска подстроки

function Pos(const substr: string; const str: string): Integer;

которая возвращает номер символа, с которого начинается искомая подстрока, если подстроки в строке нет, возвращает 0.

1. Копирование элемента строки

Var S1, S2: string;

S1 := ‘Вася – хороший человек’;

S2 := Copy(S1, 8, 7);

1. Операции сравнения = <> > <

Во многих случаях необходимо по тому или иному признаку выделить подстроку из строки. Для этого необходимо либо запомнить начало подстроки при разборе строки:

Var S1: string;

I, bi: Integer;

For I := 1 to Length(S1) do

If S1[i] = ‘,’ then

Bi := I;

Copy(S1, bi…

Либо одновременно копировать символы в некоторую буферную переменную:

Var S1, SB: string;

I: Integer;

SB := ‘’;

For I := 1 to Length(S1) do

If S1[i] = ‘,’ then

SB := SB+S1[i];

Оба этих подхода имеют преимущества и недостатки, первый быстрее работает, второй проще программировать.

Также часто возникает задача проверить символ строки на вхождение в определенный набор. Удобнее всего это сделать с помощью множеств:

For I := 1 to Length(S1) do

If S1[i] in [‘0’..’9’] then

…

Но в таком случае следует либо использовать однобайтные строки (тип AnsiString), либо воспользоваться специальной функцией CharInSet.

Вариант №1

1. Подсчитать количество символов, не являющихся буквами
2. Заменить все запятые на точку с запятой
3. Подсчитать сумму всех чисел в строке
4. Заменить каждую четную букву ‘b’ на ‘б’

Вариант №2

1. Подсчитать количество русских букв
2. Заменить все последовательности ‘abc’ на ‘ПРЕВЕД’
3. Подсчитать количество чисел в строке
4. Заменить каждую 3-ю точку на двоеточие

Вариант №3

1. Подсчитать количество латинских букв
2. Заменить все пробелы на символы подчеркивания
3. Подсчитать среднее арифметическое чисел в строке
4. Заменить каждый второй разделитель на пробел

Вариант №4

1. Подсчитать количество цифр
2. Заменить все последовательности \*\_\* на ^\_^
3. Удвоить все числа в строке