**Донецкий национальный университет**

**Кафедра компьютерных технологий**

**Отчет**

по лабораторной работе №2

**«Строки. Логическая структура строк. Операции над элементами строк»**

по курсу “ ***Структуры и организация данных в ЭВМ*** ”

Выполнил студент

Группы: ИСПР А1

**Присяжный М.Б.**

Преподаватель:

**Демин В.А.**

Донецк - 2011

**Цель работы**

Приобретение и закрепление знаний о представлении данных типа **строка символов** и о **строковых операциях**.

**Темы для предварительной проработки**

Строки: представления строк, операции над строками. Средства обработки строк в языках программирования.

1. **Описание алгоритма.**

Программа была реализована в среде Turbo Pascal 7.0

В качестве индивидуального задания была представлена следующая задача:

**Разработать процедуру или функцию, выполняющую заданную операцию над данными типа string языка Pascal. Предусмотреть и описать реакцию на некорректное задание параметров.**

**Right(s,l) -** выравнивание строки **s** справа до длины **l**

**Суть алгоритма:**

В коде используется 2 вспомогательные строки, одна из них собирает все пробелы, а вторая – значащие символы и единичные пробелы между словами. В коде для определения длины строки я использовал процедуру Ord(s[0]), обойдясь без функции length (стандартной процедуры обработки строк).

1. **Описание программных элементов.**

**Было разработано 2 процедуры:**

* **Procedure Right(s:string;l:integer);** - процедура осуществляет выравни вание текста по правой границе.

Procedure Right(var s:string;l:byte);

var t1,t2:string;

begin

osc:=true;

t1:='';

t2:='';

For i:=1 to (maxlen-l) do t1:=t1+' ';

For i:=1 to l do

if (s[i]=' ') and (s[i+1]=' ') then t1:=t1+' ' else

begin

osc :=false;

t2:=t2+s[i];

end;

s:=t1+t2;

end;

s – текущая строка, i – счетчик, t1, t2 – вспомагательные строки, maxlen – длина макс. строки = 80, l – длина текущей строки.

* **Procedure DrawLine; -** процедура рисует вспомогательную линию.

В коде используются процедуры из модуля crt, но они, как и процедура **DrawLine** служат лишь для удобного диалога с пользователем и улучшенного интерфейса. В частности в коде можно заметить некоторые циклические конструкций, обработанных для организации повторного ввода данных.

**Листинг программы**

Program sod2;

uses crt;

const maxlen=80;

var s1:string;

i,lengths,k:byte;

p:char;

osc:boolean;

Procedure DrawLine;

begin

for i:=1 to maxlen do

write('\_');

end;

Procedure Right(var s:string;l:byte);

var t1,t2:string;

begin

osc:=true;

t1:='';

t2:='';

For i:=1 to (maxlen-l) do t1:=t1+' ';

For i:=1 to l do

if (s[i]=' ') and (s[i+1]=' ') then t1:=t1+' ' else

begin

osc :=false;

t2:=t2+s[i];

end;

s:=t1+t2;

end;

Begin

TextColor(Black);

TextBackground(White);

repeat

clrscr;

s1:='';

WriteLn('Input STRING:');

WriteLn;

ReadLn(s1);

lengths:=Ord(s1[0]);

DrawLine;

If lengths>maxlen then WriteLn('Large stiring!!!') else

begin

Right(s1,lengths);

If lengths=0 then WriteLn('Empty STRING!!!') else

If osc then WriteLn('Only SPACE characters!!!')

else

begin

WriteLn('New string:');

Write(s1);

end;

end;

WriteLn('Try Again?');

WriteLn('[Y/N]');

repeat

p:=readkey;

until (p='n') or (p='y') or (p=#27) ;

until (p='n') or (p=#27);

end.

1. **Описание результатов контрольного тестирования задачи.**

Ввод строки со значащими символами:

**Input STRING:**

**435435 3 rfg rege rgfhg**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**New string:**

**435435 3 rfg rege rgfhg**

**Try Again?**

**[Y/N]**

Ввод пустой строки:

**Input STRING:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Empty STRING!!!**

**Try Again?**

**[Y/N]**

Ввод строки с пробелами:

**Input STRING:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Only SPACE characters!!!**

**Try Again?**

**[Y/N]**

Ввод «Длинной» строки:

**Input STRING:**

**fdsgfkfhkhfdjhfjf777777777777777777777777777777777777777777777777777777777777ddd**

**dd**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Large stiring!!!**

**Try Again?**

**[Y/N]**

1. **Выводы.**

В результатах тестирования программы были представлены следующие пункты:

* Ввод строки с значащими символами
* ввод пустой строки
* ввод «длинной» строки
* ввод строки с пробелами

Обобщив результаты, полученные после контрольного тестирования программы индивидуального задания (Вариант №9), можно свидетельствовать, что:

* строка выравнивается верно
* выдача сообщений об ошибках организована корректно

За время написания программы:

* приобрел и закрепил навыки работы со строками
* закрепил знания о модульности программы с точки зрения обрабатываемых данных
* повторил основы организации циклов, условий и других конструкций языка Pascal
* повторил работу с подпрограммами модуля crt.