Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное
учреждение высшего профессионального образования
«МАТИ — Российский государственный технологический университет
имени К. Э. Циолковского» (МАТИ)

Кафедра «Моделирование систем и информационные технологии»

ИСПОЛЬЗОВАНИЕ СОСТАВНЫХ ТИПОВ ДАННЫХ ПАСКАЛЯ. ЧАСТЬ 1. СТРОКИ И МНОЖЕСТВА

Методические указания к лабораторной работе по курсу
«Алгоритмические языки и программирование»

Составитель В. В. Лидовский

Москва 2013

Л Использование составных типов данных паскаля. Часть 1. Строки и множества: методические указания к лабораторной работе по курсу «Алгоритмические языки и программирование». / сост. В. В. Лидовский. — М.: МАТИ, 2013. — 12 с.

ВВЕДЕНИЕ

Настоящие методические указания предназначены для обеспечения учебного процесса студентов дневной формы обучения по направлению подготовки 230100.62 “Информатика и вычислительная техника” по профилю подготовки 230102.62 “Автоматизированные системы обработки информации и управления” при выполнении лабораторной работы по предмету “Алгоритмические языки и программирование”.

Цель лабораторной работы: изучить методы программирования на языке паскаль для работы со строковым и множественным типами.

1. СИСТЕМНЫЕ ТРЕБОВАНИЯ

Для выполнения лабораторной работы необходимы следующие системные компоненты:

а) компьютер, работающий под управлением операционной системы Linux;

б) компилятор Free Pascal.

2. СТРОКИ

Строки — это (упакованные) массивы символов. Строковый тип соответствует регулярному типу array[0..n]of char, где n — максимальная длина строки, n<256. Элемент такого массива с индексом 0 хранит длину строки, а сама строка хранится, начиная с индекса 1. Строковый тип описывается служебным словом string, например, тип string[n] соответствует приведенному ранее. Для строкового типа существует множество стандартных операций, что отличает строки от одномерных массивов символов.

Литералы-строки — это идущие подряд литералы-символы без смежных апострофов. Например, ’XYZ’ — строка из 3 символов, #8#8 — строка из 2 символов, ’A’#$D#10’A’’BC’ — строка из 8 символов.

К строкам применимы обычные операции сравнения: =, <, >, <=, >=, <>. Строки сравниваются сначала посимвольно, а затем по длинам. Например, верно, что ’FIVE’ < ’TEN’ и ’AB’ < ’ABC’. Такое сравнение называют словарным или лексикографическим.

К строкам также применимы следующие операции:

+ — склейка строк, например, ’ABC’+’CDE’=’ABCDE’;

length(s) — возвращает длину строки s, например, length(’ABCDEF’)=6. Верно, что ord(s[0])=length(s);

pos(s1,s2) (position — позиция) — поиск строки s1 в строке s2: если поиск успешен, то эта функция возвращает позицию s1 в s2, если же поиск неуспешен, то функция возвращает 0, — например, pos(’AB’,’RABABR’)=2, pos(’A’,’BC’)=0;

copy(s,p,n) — выделение подстроки длины n с позиции p строки s, например, значением функции copy('UVWXYZ',2,4) будет строка 'VWXY';

low(s) — для любых строки или строкового типа s возвращает 0;

high(s) — для строки или строкового типа s возвращает максимальную возможную длину для строк данного типа.

Для работы со строками можно испльзовать процедуры:

insert(s1,s2,p) — вставка строки s1 в строку s2 с позиции p, например, если s='ABC', то после insert('YZ',s,2) s будет равно 'AYZBC';

delete(s,p,l) — удаление подстроки длины l с позиции p из строки s, например, если s='ABCDE', то после delete(s,3,2) s будет равна 'ABE';

val(s,n,p) (value — значение) — преобразование строки s в число n, например, val('123',n,p) установит n=123. Если s соответствует число, то p=0, иначе p установится равным позиции, с которой преобразование строки в число стало невозможным, например, вызов val('12x3',n,p) установит p=3. Параметр n может быть любого числового типа, а параметр p должен быть типа integer;

str(i[:w[:d]],s) (string — строка) — преобразование числа i (любого числового типа) в строку s. Опциональные параметры w и d служат для форматирования: w — это количество позиций для числа, а d — количество цифр после десятичной точки, например, после вызова str(10,s) установится s='10', после str(10:4,s) — s=' 10', после str(sqrt(2):6:4,s) — s='1.4142'.

Строки частично совместимы с символьным типом. Символ в строковых операциях ведет себя как строка с константной длиной 1.

Типичная задача при работе со строками — это поиск с заменой. Наличие такой операции — это неприменный атрибут практически любого текстового редактора. Некоторые языки программирования, в частности, перл, рубин, питон и оук, поддерживают эту операцию непосредственно. Наиболее проста реализация этой операции при помощи вспомогательной строки, в которой будет собираться результат замен.

Итак, имеем три строки исходных данных: 1) строку, в которой нужно провести замену; 2) заменяемую строку; 3) новую строку, заменяющую вторую. Вначале инициализируем вспомогательную строку-результат пустым значением. Затем ищем вхождение 2-й строки в 1-ю. Если вхождение есть, то берем часть первой строки до найденной позиции вхождения и переносим ее в конец вспомогательной строки. Добовляем к результату 3-ю строку. Удаляем из 1-й строки часть до конца найденного вхождения 2-й. Повторяем поиск. Если при поиске вхождения 2-й строки в 1-ю нет, то добавляем содержимое 1-й строки к результату и заканчиваем работу.

Рассмотрим пример. Пусть 1-я строка s1 равна 'ABCCCBCF', 2-я s2 — 'BC', 3-я s3 — 'CB'. Установим результат r в ''. Тогда при поиске с позиции 2 будет обнаружено вхождение s2 строки в s1. Добавим 'ACB' к

результату и удалим три первых символа s1, что установит s1 в 'CCBCF'. При следующем поиске s2 будет найдена с позиции 3. Добавим 'CCCB' к r — это установит r в 'ACBCCCB'. Удалим затем четыре первых символа s1. Теперь значение s1 — 'F'. Следующий поиск не обнаружит искомого вхождения. Добавим 'F' к r и получим тем самым окончательный результат в r — 'ACBCCCBF'. Процесс выполнения замены можно представить в следующем пошаговом отчете.

шаг	s1	pos(s2, s1)	r
0	'ABCCCBCF'	2	''
1	'CCBCF'	3	'ACB'
2	'F'	0	'ACBCCCB'
3			'ACBCCCBF'

Стандартная функция поиска pos имеет несколько ограничений: ищет всегда с первой позиции, не может искать с конца строки.

Рассмотрим как можно написать функцию xpos с тремя аргументами, реализующую поиск с заданной позиции. Нужно, чтобы, например, xpos('AB','ABAB',2)=3. Очевидно, что xpos(s1,s2,1)=pos(s1,s2).

```
function xpos(s1, s2: string; p: byte): byte;
(* ищет s1 в s2 с позиции p *)
   begin
      xpos := pos(s1, copy(s2, p, length(s2) - p + 1));
      if xpos > 0 then
         xpos := xpos + p - 1
   end;
```

А так можно реализовать функцию rpos для поиска справа налево. Должно получиться, что, например, rpos('AB','ABAB')=3.

```
function rpos(s1, s2: string): byte;
(* поиск s1 в s2 с конца s2 *)
   begin
      for rpos := length(s2) - length(s1) + 1 downto 1 do
         if s1 = copy(s2, rpos, length(s1)) then
            exit;
      rpos := 0
   end;
```

В случае, если нужно искать в строке справа налево только отдельный символ, то в rpos вместо вызова функции copy можно обращаться к элементу массива-строки и использовать 1 вместо вызовов length.

Другая типичная задача для строк — это разбор строки на отдельные компоненты. В сложном случае, например, разбор фразы естественного

языка, решение такой задачи требует привлечения средств, относящихся к области Искусственного интеллекта. Более простые случаи разбора, например, арифметического выражения или фразы на языке программирования, относятся к Теории компиляции, к, как правило, детерминированному лексическому и синтаксическому анализу. В простейших случаях такой разбор может производиться несложными алгоритмами, реализация которых прямо опирается на стандартные функции для работы со строками.

Рассмотрим разбор полного имени интернет-ресурса. Такое имя состоит из четырх компонент*, именующих: 1) протокол; 2) сервер; 3) каталог; 4) ресурс. Например, имя http://www.yandex.ru/index.html состоит из названия протокола http**, имени сервера www.yandex.ru, имени каталога / (корневой каталог) и имени ресурса index.html.

Название протокола — это подстрока от начала до двоеточия. Имя сервера начинается после двух знаков / и заканчивается первым следующим /. Имя каталога — это подстрока от первого одиночного знака / до последнего, включая эти знаки. Имя ресурса — это концевая подстрока от последнего /.

Рассмотрим еще разбор полного имени файла командной строки Microsoft Windows или DOS. Здесь полное имя также состоит из четырех компонент, именующих: 1) устройство внешней памяти, как правило, дисковый накопитель или flash-память; 2) каталог; 3) собственно имя файла; 4) расширение имени файла (опционально). Например, имя C:\CONFIG.SYS состоит из названия диска — C, каталога — \, имени — CONFIG и расширения — SYS. Другой пример, строка C:\WINDOWS\SYSTEM32\COMMAND.COM разбивается на части C, \WINDOWS\SYSTEM32\, COMMAND и COM.

Имя диска — это первый символ полного имени файла. Имя каталога заключено между знаками \ (позиция первого \ фиксирована — она всегда равна 3). Концевая подстрока от последней точки в имени файла — это расширение. Точка, выделяющая расширение, не включается ни в имя, ни в расширение. Если точка является последним символом в имени файла, то считается, что расширение отсутствует. При выделении имен каталога и расширения удобно пользоваться функцией с возможностями rpos.

3. МНОЖЕСТВА

Множество — это чрезвычайно общее понятие. В компьютере можно представлять лишь множества, на которые наложены ряд ограничений.

---

* В адресах интернет-ресурса могут быть и другие компоненты, например, установка значений переменных, имя пользователя, номер порта и др.

** Типичные протоколы, помимо http, — это https, ftp и file.

Множества могут быть только конечными (до 256 элементов) и должны состоять из однотипных элементов дискретного типа. Тип множества описывается при помощи служебного слова set.

Примеры:

```
set of (David, Helena, Diana, Ann, Robert) (* множество имен *)
set of 'A'..'Z'       (* множество из заглавных букв английскиго
                                                        алфавита *)
set of char           (* множество всех символов *)
```

Литералы-множества записываются перечислением своих элементов в квадратных скобках через запятую. В таком списке можно использовать диапазоны, первый элемент которых должен отделяться от второго двумя точкам. Примеры: [0..9] — десятичные цифры, ['0'..'9','A'..'F','a'..'f'] — множество шестнадцатеричных символов-цифр.

К данным множественного типа применимы операции сравнения: = и <> ($\neq$)*. К ним также применимы операции включения <= ($\subset$) и >= ($\supset$), например, если A и B множества, то A<=B означает A подмножество B ($A \subset B$) и, аналогично, A>=B означает, что B подмножество A ($B \subset A$). Например, верно, что [1,2]<=[0..5] ($\{1,2\} \subset \{0,\ldots,5\}$), и неверно, что [2,5,8]<=[3..4]($\{2,5,8\} \subset \{3,\ldots,4\}$).

К множествам применимы еще следующие операции:

\+ — объединение множеств ($\cup$);

− — вычитание множеств ($\setminus$ или $-$);

* — пересечение множеств ($\cap$);

in — проверка на вхождение элемента в множество, например, если s — множество, состоящее из чисел 1, 4 и 5, то значение выражения 2 in s — false, а 5 in s — true ($\in$).

Примеры: [1..3]+[1,4,5]=[1..5], [1..3]-[1,4,5]=[2,3], [1..3]*[1,4,5]=[1], 3 in [1..8] = true.

Стандартная процедура writeln не может распечатывать данные множественного типа. Поэтому для распечатки множеств необходимо создавать специальную процедуру.

4. ПОРЯДОК ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ

Необходимо создать три программы, решающие задачи согласно установленному варианту задания.

5. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Почему максимальная длина строки типа string ограничена 255 знаками?

---

* Здесь и далее в скобках приводится эквивалентное обозначение, принятое при записи математических формул.

2. Что можно сказать о результате операции соединения двух строк в паскале, если сумма их длин превышает 255?
3. Чему равен результат функции length при ее применении к величине типа char?
4. Можно ли поиск с заменой реализовать без использования четвертой строки?
5. Как на паскале записывается пустое множество?
6. Можно ли на паскале записать литерал, соответствующий множеству всех подмножеств заданного множества?
7. Как на паскале выразить операцию симметрическая разность множеств?
8. Сколько байт необходимо для хранения данных типа множество из 256 элементов?
9. Какой оператор паскаля можно записать только через условные операторы и логические выражения с операцией in?
10. Зависит ли количество бит для представления элемента множества в паскале от типа элемента множества?
11. Зависит ли время операции с множествами от количества элементов в множествах-операндах в паскале?

6. ВАРИАНТЫ РАБОТ

Лабораторная работа по теме “Использование составных типов данных паскаля. Часть 1. Строки и множества” имеет 12 вариантов заданий. Вариант задания определяется вариантом задачи, решаемой каждой из трех программ.

Первая программа должна быть представлена в одном из трех вариантов:

1) для проведения замены всех вхождений заданной строки на заданную строку-замену в исходной строке;
2) для проведения замены первого вхождения заданной строки с заданной позиции на заданную строку-замену в исходной строке;
3) для проведения замены последнего вхождений заданной строки на заданную строку-замену в исходной строке.

В каждом из вариантов необходимо обеспечить интерактивный ввод трех строк данных. Во втором варианте нужно еще обеспечить интерактивный ввод позиции начала поиска. В качестве результата должна быть распечатана строка-результат замены. В 1-м варианте нужно еще распечатать количество проведенных замен.

Вторая программа должна быть представлена в одном из двух вариантов:

1) для разбора полного имени файла Microsoft Windows;
2) для разбора адреса интернет-ресурса.

Программа должна обеспечить интерактивный ввод разбираемой строки и распечатку результатов разбора, где должны быть представлены пары из названия компоненты и ее значения.

В обоих программах для работы со строками на этапе подготовки вместо интерактивного ввода данных для ускорения работы лучше использовать установленные присваиваниями значения строк-данных.

Третья программа должна создать и распечатывать два множества из случайных чисел из заданного вариантом диапазона, 1-е из 12 чисел, а 2-е из 24. Вывести на экран дисплея их объединение, разности и пересечение, а также ответ на вопрос: “Входит ли 1-е множество во 2-е?” Диапазон 1-го варианта — это числа от 12 до 124, диапазон 2-го — от 16 до 64. Программа должна содержать процедуру для распечатки множества.

Один из 12 вариантов задания выбирается согласно таблице.

Вариант задания	Вариант программы №1	Вариант программы №2	Вариант программа №3
1	1	1	1
2	2	1	1
3	3	1	1
4	1	2	1
5	2	2	1
6	3	2	1
7	1	1	2
8	2	1	2
9	3	1	2
10	1	2	2
11	2	2	2
12	3	2	2

ЛИТЕРАТУРА

1. *Зуев Е.А.* Язык программирования Turbo Pascal 6.0 — М.: Унитех, 1992. — 298 с.
2. *Немнюгин С. А.* Turbo Pascal. Программирование на языке высокого уровня — СПб.: Питер, 2008. — 544 с.
3. *Нефедов В. Н., Осипова В. А.* Курс дискретной математики — М.: МАИ, 1992. — 264 с.
4. *Петерсен Р.* Linux: полное руководство — Киев: “Ирина” BHV, 2000. — 642 с.
5. *Фараонов В. В.* Turbo Pascal: учебное пособие — СПб.: Питер, 2010. — 368 с.

ПРИЛОЖЕНИЕ

ПРИМЕРЫ РАБОТЫ С ВТОРОЙ ПРОГРАММОЙ

```
Введите имя интернет-ресурса: http://www.linux.org/tutorial
Протокол: http
Сервер: www.linux.org
Каталог: /
Ресурс: tutorial

Введите имя интернет-ресурса: ftp://ftp.funet.fi/pub/README
Протокол: ftp
Сервер: ftp.funet.fi
Каталог: /pub/
Ресурс: README

Введите полное имя DOS: d:\dbs\data\table
Диск: d
Каталог: \dbs\data\
Имя: table
Расширение отсутствует

Введите полное имя DOS: c:\command.com
Диск: c
Каталог: \
Имя: command
Расширение: com

Введите полное имя DOS: e:\dir\subdir\dir.ext\fn.
Диск: e
Каталог: \dir\subdir\dir.ext\
Имя: fn
Расширение отсутствует
```

СОДЕРЖАНИЕ

ИСПОЛЬЗОВАНИЕ СОСТАВНЫХ ТИПОВ ДАННЫХ ПАСКАЛЯ. ЧАСТЬ 1. СТРОКИ И МНОЖЕСТВА

Методические указания к лабораторной работе по курсу
«Алгоритмические языки и программирование»

Составитель:
Владимир Викторович **Лидовский**

Технический редактор *Е. Е. Костылева*

Оригинал-макет подготовлен в пакете Plain-TEX

Подписано в печать _._.2013
Усл. печ. л. 1,86. Формат 60×84 $^1/_{16}$.
Тираж 70 экз. Заказ № _
Издательский-типографский центр МАТИ
109383, Москва, ул. Полбина, 45