



Упражнения

- 1 Измените константу в программе примера 6.10 на свое имя. Используя команды графической библиотеки для работы с текстом, измените шрифт, размер символов, фон, начертание букв.
- 2 Измените программу примера 6.11 так, чтобы выполнялись указанные условия.
 1. Каждое слово должно выводиться случайным цветом.
 - 2*. Расстояние между словами должно быть 50 пикселей.
 - 3*. Каждое новое слово должно выводиться шрифтом на 5 пунктов больше, чем предыдущее. Отрегулируйте расстояние между словами так, чтобы слова при выводе не перекрывали друг друга.
- 3 Запустите программу из примера 6.12 несколько раз. Какие результаты получили?
- 4 Измените программу из примера 6.12 так, чтобы случайным образом выбиралось одно из трех слов. Выводите каждое слово своим цветом (например, первое — красным, второе — синим, третье — зеленым).

§ 7. Строковые величины

В первых языках программирования строкового типа данных не было; программист должен был сам строить функции для работы со строками.

В 1962 г. был разработан язык SNOBOL (StriNg Oriented symBolic Language), ориентированный на работу со строками. В конце 60-х гг. XX в. строковые типы данных появились в языках Algol и Fortran.

Две строки, в отличие от двух чисел, нельзя прочитать с помощью одной команды `read`, поскольку пробел для строк не разделитель, а такой же символ, как и все остальные. Необходимо использовать две команды `readln`.

Если использовать две команды `read`, то первая строка будет считана так, как нужно, а вторая строка будет пустой (она не будет вводиться). Это происходит потому, что первая команда `read` считывает данные до нажатия клавиши Enter. Вторая команда `read` прочитает один символ — символ нажатия клавиши Enter.

7.1. Ввод, вывод, присваивание строковых величин

Современные компьютеры способны обрабатывать данные, представленные различными способами: числа, тексты, графику, звуки. Вы уже знаете, как на языке программирования Pascal можно работать с целыми и вещественными числами, выполнять простейшие графические построения. Обработка текстовых данных является сегодня наиболее актуальной — это обработка различных поисковых запросов в Интернете, перевод текстов с одного языка на другой, озвучивание компьютером печатного текста и др.

В языке Pascal для работы с текстовыми данными используется тип **string (строка)**. Строки состоят из набора последовательно расположенных символов и используются для хранения текста. Они могут иметь произ-

вольную длину. Строка, в которой нет ни одного символа, называется **пустой**.

Строка описывается следующим образом:

```
var <имя строки>: string;
```

Для ввода и вывода строки используются те же команды, что и для ввода и вывода чисел: `read` (`readln`) и `write` (`writeln`) (пример 7.1). Ввод данных всегда заканчивается нажатием клавиши `Enter`, которой соответствует специальный символ: `¶`. Команда `read` считывает символы в строку до тех пор, пока не встретится этот специальный символ. Команда `readln` отличается тем, что считывает не только данные в строку, но и символ нажатия `Enter`. Сам символ `¶` к строке не приписывается.

Переменной строкового типа можно присвоить значение с помощью команды присваивания. Значение строковой величины записывается в апострофах. Пустая строка задается следующим образом: `S := ''`;

Запись поясняющего текста при выводе в команде `write` является строковой константой. К символам в строке можно обращаться, используя индекс. Нумерация символов начинается с единицы, `s[i]` соответствует `i`-му символу в строке `s` (пример 7.2).

Пример 7.3. Написать программу, которая спросит имя пользователя и выведет приветствие, обращаясь по имени.

Этапы выполнения задания

I. Исходные данные: `imja` — строка, в которой будет храниться введенное имя.

Пример 7.1. Введем строку `s` и выведем ее значение.

```
var s: string;
begin
  writeln('Введи строку');
  readln(s);
  writeln('Ввели строку: ', s);
end.
```

Результат работы:

```
Окно вывода
Введи строку
Pascal
Ввели строку: Pascal
```

Пример 7.2. Вывести третий символ строки «Информатика».

```
var s: string;
begin
  s := 'Информатика';
  writeln('3-й символ - ', s[3]);
end.
```

Результат работы:

```
Окно вывода
3-й символ - ф
```

Пример 7.3.

V. Программа:

```
var imja: string;
begin
  writeln('Как тебя зовут?');
  readln(imja);
  writeln('Привет, ', imja);
end.
```

VI. Тестирование.

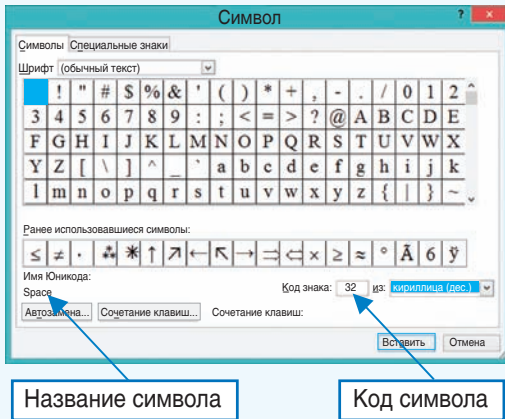
Запустить программу и ввести значение «Катя». Результат:

```
Окно вывода
Как тебя зовут?
Катя
Привет, Катя
```

Пример 7.4. Порядок расположения групп символов в таблице символов.

1. Пробел.
2. Символы цифр.
3. Заглавные латинские буквы.
4. Строчные латинские буквы.
5. Заглавные русские буквы.
6. Строчные русские буквы.

В нижеприведенной таблице символов выделен пробел, имеющий код 32. Далее видно расположение цифр и латинских букв.



Пример 7.5. Примеры сравнения строк.

S1	S2	Результат
'string'	'char'	S1 > S2
'строка'	'Строка'	S1 > S2
'книга'	'Журнал'	S1 > S2
'мебель'	'стол'	S1 < S2
'липа'	'лист'	S1 < S2
'леска'	'лес'	S1 > S2
'112'	'7'	S1 < S2
'123'	'11111111'	S1 > S2
'123'	'123'	S1 = S2

- II. Результат: строка с приветствием.
- III. Алгоритм решения задачи.

1. Ввод имени.

2. Вывод строки с приветствием.

IV. Описание переменных: `imja` — `string`.

7.2. Сравнение и сложение строковых величин

Так же как и для других типов данных, для строк определены свои операции — действия, которые можно выполнять с данными типа `string`. Для строкового типа такими операциями являются сравнение и сложение.

Для сравнения строк нужно уметь сравнивать символы. Все символы записаны в таблице символов (в документе Word эту таблицу вы использовали для вставки символа, отсутствующего на клавиатуре). Каждый символ в таблице имеет свой код (номер), и при сравнении символов сравниваются их коды — номера в таблице символов.

В таблице символов, которую использует PascalABC, 65536 символов (стандарт Юникод — англ. *Unicode*¹). Запомнить, в каком порядке записаны символы, невозможно. Достаточно знать, как расположены группы наиболее часто употребляемых символов. В примере 7.4 приведены некоторые группы символов в порядке их расположения в таблице.

Строки сравниваются посимвольно. Сначала сравниваются первые символы двух строк. Если символы различны, то больше та строка, символ которой имеет больший номер. Если

¹ <https://ru.wikipedia.org/wiki/Юникод> (дата доступа: 05.01.2019).

символы одинаковые, то переходят к сравнению следующих символов. Сравнение заканчивается, когда найдены различные символы или в одной из строк закончились символы — в этом случае больше та строка, в которой символы остались. Если при сравнении символов различия не найдены и строки закончились одновременно, то они равны (пример 7.5). Порядок, в котором меньшая строка предшествует большей, называют **лексикографическим**. Это название он получил по аналогии с размещением по алфавиту в словаре.

При сравнении строк следует помнить, что заглавные и строчные буквы — это разные буквы, поскольку имеют различные номера в таблице символов. Поэтому строки 'мама' и 'Мама' будут различными, а неравенство 'мама' > 'Мама' будет верным (у буквы 'м' код 236, а у 'М' — 204). Для проверки правильности сравнения строк можно воспользоваться программой из примера 7.6.

Пример 7.7. Написать программу, которая спросит имя пользователя, его пол, а затем поздоровается с ним, выводя красным цветом женские имена, а синим — мужские.

Этапы выполнения задания

I. Исходные данные: переменные `imja` (имя пользователя) и `pol` (пол пользователя).

II. Результат: строка с приветствием.

III. Алгоритм решения задачи.

Частично задача совпадает с примером 7.3, поэтому возьмем решение

Пример 7.6. Сравнение строк.

```
var S1,S2: string;
begin
  writeln('Введите первую строку');
  readln(S1);
  writeln('Введите вторую строку');
  readln(S2);
  if S1>S2 then
    writeln('S1 > S2')
  else
    if S1<S2 then
      writeln('S1 < S2')
    else
      writeln('S1 = S2')
end.
```

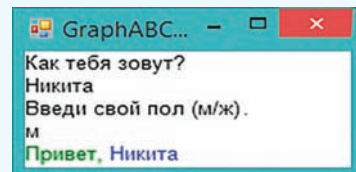
В PascalABC есть тип `char`, который позволяет хранить один символ. Над символами определены операции сравнения «<», «>», «<=», «>=», «=», «<>», которые сравнивают коды символов. Функция `ChrUnicode(n)` возвращает символ с кодом `n` (тип `char`), а функция `OrdUnicode(c)` — код символа (тип `byte`).

Пример 7.7.

```
V. Программа:
uses GraphABC;
var imja, pol: string;
begin
  writeln('Как тебя зовут?');
  readln(imja); writeln(imja);
  writeln('Введи свой пол (м/ж).');
  readln(pol); writeln(pol);
  SetFontColor(clgreen);
  write('Привет, ');
  if pol = 'ж' then
    SetFontColor(clred)
  else
    SetFontColor(clblue);
  writeln(imja);
end.
```

VI. Тестирование.

Результат для имени Никита:



Пример 7.8. Примеры сложения строк.

```
var s1, s2, s3, s4: string;
begin
  s1 := 'Петя';
  s2 := 'Иванов';
  s3 := s1 + ' ' + s2;
  s4 := s2 + ' ' + s1;
  writeln(s3);
  writeln(s4);
end.
```

Результат:

Окно вывода
Петя Иванов
Иванов Петя

Пример 7.9.

V. Программа:

```
var str, gor, s: string;
begin
  writeln('Введите название госу-
дарства');
  readln(str);
  writeln('Введите название
столицы');
  readln(gor);
  s := 'Столица государства '
+ str + ' — город ' + gor;
  writeln(s);
end.
```

VI. Тестирование.

Запустить программу и ввести значения Республика Беларусь и Минск.
Результат:

Окно вывода
Введите название государства
Республика Беларусь
Введите название столицы
Минск
Столица государства Республика Беларусь —
город Минск

из этого примера и изменим его. Цвет для текста можно задавать только в графическом окне, поэтому подключим графический режим.

1. Ввод исходных данных.

2. Сравнение переменной `pol` со значением 'ж'. Если результат сравнения — истина, то устанавливаем красный цвет, иначе — синий.

3. Вывод строки.

IV. Описание переменных: `imja`, `pol` — string.

Для строк определена операция *сложение* (конкатенация). Обозначается операция знаком «+». В результате сложения двух строк получается новая строка, в которой после символов первой строки будут записаны символы второй строки. Результат данной операции зависит от порядка слагаемых (пример 7.8).

Пример 7.9. Написать программу, которая просит ввести название государства и его столицу, затем выводит сообщение:

Столица государства ... — город ...

Вместо многоточия должны быть выведены соответствующие значения.

Этапы выполнения задания

I. Исходные данные: переменные `str` (название государства) и `gor` (название города).

II. Результат: переменная `s` (итоговая строка).

III. Алгоритм решения задачи.

1. Ввод исходных данных.

2. Создание итоговой строки.

3. Вывод строки.

IV. Описание переменных: `str`, `gor`, `s` — string.

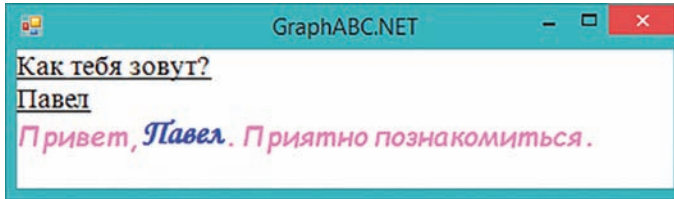


1. Как описываются строковые переменные?
2. Из чего состоят строки?
3. Какие операции возможны над строками?
4. Как упорядочены символы в таблице символов?
5. Как сравниваются строки?
6. Что является результатом сложения двух строк?

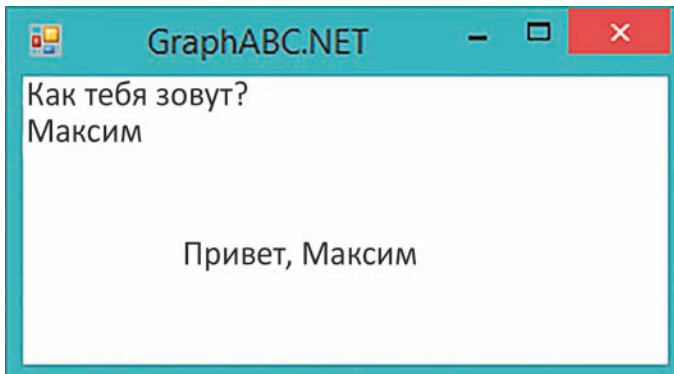


Упражнения

- 1 Внесите указанные изменения в программу из примера 7.3.
 1. После имени пользователя выводить восклицательный знак.
 2. Измените программу так, чтобы выводился текст:
Привет, Ольга. Приятно познакомиться.
 3. Подключите графический режим. Задайте параметры шрифта для текста «Как тебя зовут?» и выводимого текста. Например, так:



- 2 Выполните следующие задания, изменив программу примера 7.3.
 1. Выведите приветствие в графическом окне в точке с координатами (70; 70).



2. Задайте координаты вывода случайным образом.
3. Измените программу так, чтобы координаты месторасположения текста можно было вводить.
4. Задайте параметры шрифта для выводимых сообщений.

3 Используя программу из примера 7.6, проведите исследование по сравнению символов.

1. Совместно заполните таблицы.
 2. Добавьте в каждую таблицу 2—3 строки с другими символами в соответствии с правилом сравнения.
 3. Проверьте некоторые из результатов, используя таблицу символов (можно открыть в Word).
- Сделайте выводы.

Сравнение символов русского алфавита, набранных в одном регистре		
S1	S2	Результат
ф	а	
С	Я	
я	у	

Сравнение символов латинского алфавита, набранных в одном регистре		
S1	S2	Результат
r	z	
W	J	
q	d	

Сравнение символов русского алфавита, набранных в разных регистрах		
S1	S2	Результат
ы	Ы	
а	Я	
я	А	

Сравнение символов латинского алфавита, набранных в разных регистрах		
S1	S2	Результат
Q	q	
Z	a	
A	z	

Сравнение символов цифр		
S1	S2	Результат
1	3	
7	9	
4	4	

Сравнение цифр и букв		
S1	S2	Результат
Q	1	
2	б	
9	ю	

Сравнение русских и латинских букв		
S1	S2	Результат
Q	л	
w	б	
Ч	k	

Сравнение знаков и цифр		
S1	S2	Результат
,	5	
7	!	
9	>	

Сравнение знаков и букв		
S1	S2	Результат
пробел	f	
W	{	
№	Я	

Сравнение знаков		
S1	S2	Результат
*	+	
.	,	
(	)	

4 Используя программу из примера 7.6, сравните строки. Откройте файл с таблицей и запишите результаты.

S1	S2	Результат сравнения
'string'	'String'	
'строка'	'символ'	
'alinéa'	'caractère'	
'Zeile'	'Zeichen'	
'cuerda'	'simbolo'	
'778'	'8'	
'876'	'55555555'	
'2 + 2'	'4'	

5 Какое приветствие выведет программа из примера 7.7, если, указывая пол, пользователь введет символ, отличный от «м» и «ж»? Внесите в программу изменения так, чтобы в этом случае вместо приветствия выводилось сообщение «Ошибка ввода».

6 Напишите программу, которая организует диалог с пользователем по следующему шаблону:

Как тебя зовут?
Ваня
Привет, Ваня. А ты любишь читать? (д/н)
д
Хорошо! Книга – источник знаний.

Как тебя зовут?
Петя
Привет, Петя. А ты любишь читать? (д/н)
н
Плохо, из книг можно узнать много интересного.

Как тебя зовут?
Леша
Привет, Леша. А ты любишь читать? (д/н)
v
Леша, ты ответил на вопрос некорректно.

Добавьте в программу свои вопросы и сообщения.

7 Напишите программу для решения следующей задачи: пользователь вводит свою фамилию, класс и учебное заведение, программа должна вывести сообщение «Учащийся ... учится в ... классе ГУО “...”». Вместо многоточия должны быть выведены соответствующие значения.