**Информатика 10 класс**

### Конспект урока "Символьный тип данных" (на основе данного текста составьте краткий конспект на 1-1,5 стр максимум)

Ссылка на дополнительные материалы <https://youtu.be/ls7kcqU3l1A>

Вопрос:

· Символьный тип данных и его значения.

· Функции обработки символьных данных.

· Принцип последовательного кодирования алфавитов.

Величина символьного типа – char в языке Pascal может хранить любой символ компьютерного алфавита. В среде Pascal ABC – это символы кодовой таблицы Unicode-16. Так как это 16-битная кодировка, то переменная символьного типа в среде Pascal ABC занимает 2 байта оперативной памяти.

Символьный тип данных является порядковым. Это значит, что переменная символьного типа может принимать конечное упорядоченное множество значений. У каждого символа есть порядковый номер, то есть код этого символа в кодовой таблице от 0 до 65 535:

· 0..31 – управляющие символы, которые не отображаются;

· 32 – пробел;

· 33..127 – буквы английского алфавита, цифры, знаки препинания и знаки основных арифметических операций;

· 128..65 535 – символы различных региональных алфавитов, в том числе и русского.

Задача: В файле input.txt в первых двух строках записаны 2 слова одинаковой длины до 20 символов. Вывести на экран расстояние между ними. Расстоянием между двумя словами одинаковой длины называется количество знаковых позиций, в которых эти слова имеют отличия. Например, расстояние между словами Казак и Радар равно 3.

|  |  |
| --- | --- |
| пояснение | Код программы |
| Напишем программу для решения задачи. Назовём её rasstoyanie. Для работы программы нам потребуется файловая переменная – f, 2 массива из 20 символов, в которых мы будем хранить буквы, из которых состоят слова, назовём их соответственно a и b. Также нам потребуются переменные для хранения длины слов, а также для хранения номера текущего символа и расстояния между словами. Назовём их соответственно i, n и s. По условию задачи они будут принадлежать к целочисленному типу byte.  Напишем логические скобки. Тело программы будет начинаться с операторов assign, который будет назначать переменной f файл input.txt, и reset, который будет открывать назначенный файл для чтения. Дальше присвоим переменной n значение 0, так как длина слов нам пока неизвестна. Дальше напишем цикл для считывания из файла первого слова в массив и подсчёта его длины. Это будет цикл с предусловием. Условием продолжения его работы будет то, что не достигнут конец строки, то есть not (eoln (f)). В логических скобках запишем тело цикла. Оно будет содержать оператор, увеличивающий n на 1 и оператор read, считывающий из файла значение n-ного элемента массива a. По завершении работы цикла мы считаем первое слово из файла, а также рассчитаем его длину в переменной n. Для того, чтобы перевести курсор считывания на следующую строку файла, напишем оператор readln (f). Дальше запишем цикл для считывания из файла второго слова в массив b. Так как длина слова нам уже известна это будет цикл для i, изменяющегося от 1 до n. Он будет содержать оператор read, считывающий из файла значение i-того элемента массива b. После того, как мы считали слова из файла, закроем его с помощью оператора close.  Теперь запишем оператор присваивания переменной s значения 0, так как мы ещё не рассчитывали расстояние между словами. Запишем цикл для расчёта расстояния между словами. В нём будут последовательно сравниваться соответствующие знаковые позиции в обоих словах. Это будет цикл для i, изменяющегося от 1 до n. Он будет содержать условный оператор, условием которого будет то, что a[i] ≠ b[i]. После служебного слова then в нём будет следовать оператор увеличивающий значение s на 1. После цикла нам остаётся с помощью оператора write вывести на экран сообщение о том, что расстояние между словами, содержащимися в файле, равно значению переменной s. | program rasstoyanie;  var  f: text;  a, b: array [1..20] of char;  i, n, s: byte;  begin  assign (f, 'input.txt');  reset (f);  n:=0;  while not (eoln (f)) do    begin  n:=n+1;  read (f, a[n]);  end;  readln (f);  for i:=1 to n do  read (f, b[i]);  close (f);  s:=0;  for i:=1 to n do  if a[i]<>b[i]  then s:=s+1;  write ('Расстояние между словами, содержащимися в файле: ', s);  end. |

Создадим пустой файл и запишем в нём два слова одинаковой длины, например, «кошка» и «мышка». Сохраним этот файл в директории по умолчанию под именем input.txt, после чего закроем его.

Запустим программу на выполнение. Расстояние между словами «кошка» и «мышка» действительно равно двум. Программа работает правильно. Задача решена.

Рассмотрим функции для работы с символьными значениями. Первая функция преобразования символа в его код. Она записывается служебным словом ord, после которого в скобках указывается величина символьного типа. Функция возвращает целое число – код символа в кодовой таблице. Есть и противоположная функция, преобразующая код в соответствующий символ. Она записывается служебным слово chr, после которого в скобках указывается целое число – код символа в кодовой таблице. Функция возвращает символ, соответствующий заданному коду.

|  |  |
| --- | --- |
| пояснение | Код программы |
| Задача: Написать программу, которая с использованием изученной функции преобразования кода в символ, выводит на экран коды с 32 по 127 и соответствующие им символы.  Начнём написание программы. Назовём её simvoly. Для работы программы нам потребуется всего одна переменная, которая будет хранить код текущего символа. Назовём её i. По условию задачи она будет принадлежать к целочисленному типу byte. Напишем логические скобки. Тело программы будет содержать цикл для i, изменяющегося от 33 до 127. Это будет цикл, перебирающий коды элементов. В нём будет всего один оператор вывода writeln, который будет выводить на экран значение кода – i, и через тире после него соответствующий ему символ – chr (i). Чтобы выровнять колонки укажем формат вывода кода. Выделим для него 3 знаковые позиции. | program simvoly;  var  i: byte;  begin  for i:=33 to 127 do  writeln (i:3, ' - ', chr (i));  end. |

Запустим программу на выполнение. Она вывела на экран коды и соответствующие им символы.

Просматривая полученную таблицу кодов, обратим внимание, что цифры в ней расположены одна за другой по возрастанию. Точно так же расположены буквы английского алфавита. Это учитывается при обработке символьной информации и называется принципом последовательного кодирования алфавитов.

При выполнении операций сравнения символов происходит сравнение их кодов. Также благодаря принципу последовательного кодирования алфавитов, можно определить является ли символьная величина с цифрой. с является цифрой в том случае, если её значение больше или равно символу ноль, а также меньше или равно символу девять. Аналогично можно определить является ли величина с заглавной или прописной буквой английского алфавита.

‘0’ < с < ‘9’

Признак того, что символ с – цифра

‘a’ < с < ‘z’ или ‘A’ < с < ‘Z’

Признак того, что символ с – буква английского алфавита

Задача: Написать программу, которая заполняет массив из n элементов случайными строчными буквами английского алфавита, после чего сортирует его элементы по невозрастанию. Программа должна вывести на экран как исходный, так и отсортированный массив. n вводится с клавиатуры и находится на промежутке от 1 до 30.

|  |  |
| --- | --- |
| пояснение | Код программы |
| Начнём написание программы для решения задачи. Назовём её sortirovka simvolov. Для работы программы нам потребуется массив s из 30 элементов символьного типа char. Также нам потребуются переменные для хранения размерности массива и номера текущего элемента. Назовём их соответственно n и i. По условию задачи они будут целочисленного типа byte.  Напишем логические скобки. Тело программы будет начинаться с оператора writeln, который выводит на экран сообщение о том, что это программа, которая генерирует массив из n случайных букв английского алфавита и сортирует этот массив по невозрастанию. Дальше будет следовать оператор write, который выводит на экран запрос на ввод n, а также оператор readln, считывающий значение n. Дальше напишем цикл для заполнения первых n элементов массива s случайными буквами английского алфавита. Это будет цикл с параметром i, изменяющимся от 1 до n. Он будет содержать оператор присваивания i-тому элементу массива s значения функции chr, с аргументом равным сумме кода прописной буквы a и целого случайного числа до 25, которое генерируется функцией random (26), так как в английском алфавите 26 букв.  Напишем оператор write, выводящий на экран сообщение о том, что это исходный массив и двоеточие. После него запишем цикл вывода элементов массива. Это будет цикл для i, изменяющегося от 1 до n. Он будет содержать оператор write, который будет выводить на экран значение i-того элемента массива s. После цикла для перехода на следующую строку напишем оператор writeln без параметров.  Дальше вызовем процедуру для сортировки элементов массива – sort без параметров, которую мы опишем позже. После этого напишем оператор write, выводящий на экран поясняющее сообщение о том, что это отсортированный массив и двоеточие. Дальше должен следовать цикл вывода на экран элементов массива s. Просто скопируем и вставим предыдущий. На этом описание основной программы завершено. | program sortirovka\_simvolov;  var  s: array [1..30] of char;  n, i: byte;  begin  writeln ('Программа, которая генерирует массив из n случайных букв английского алфавита и сортирует его по невозрастанию.');  write ('n=');  readln (n);  for i:=1 to n do  s[i]:=chr (ord ('a')+random (26));    write ('Исходный массив: ');  for i:=1 to n do  write (s[i]);  writeln ();  sort ();  write ('Отсортированный массив: ');  for i:=1 to n do  write (s[i]);  end. |

Важно запомнить:

· Мы узнали, что значениями символьного типа char являются любые символы компьютерного алфавита.

· Переменная символьного типа в среде Pascal ABC занимает 2 байта оперативной памяти.

· Мы изучили функцию преобразования символа в его код – ord и обратную ей функцию chr.

· Мы узнали, что во всех кодовых таблицах выполняется принцип последовательного кодирования английского алфавита и алфавита десятичных цифр.

**Домашняя работа: написать в тетради программу для данной задачи, написав пояснения что выполняется в каждой строке программы**

**Задача:** Задача: В файле input.txt в первых трех строках записаны 3 слова одинаковой длины до 20 символов. Вывести на экран расстояние между ними. Пример молоко, дерево, погода