

Занятие по дисциплине «Основы алгоритмизации и программирование» на тему: «Строковый тип данных в Turbo Pascal».

Тип занятия: урок изучения и первичного закрепления нового материала

Цели занятия:

1. Изучить строковые константы и переменные, процедуры, функции и операции над строками; закрепить изученный материал при решении задач на строковый тип данных.
2. Развивать алгоритмическое мышление и познавательную деятельность студентов; интерес к программированию; внимательность и самостоятельность при решении задач на ПК.
3. Воспитывать у студентов ответственное отношение к учебному труду, аккуратность, дисциплинированность, усидчивость.

Методы обучения: словесный, наглядный, практический.

Формы обучения: коллективная, индивидуальная.

ТСО: интерактивная доска, ПК, мультимедиа проектор.

Программное обеспечение:

Операционная система Microsoft Windows XP Professional.

Пакет офисных приложений Microsoft Office.

Turbo Pascal.

Структура занятия:

1. Постановка темы и целей урока (1 минута).
2. Проверка усвоения изученного ранее материала (4 минуты).
3. Изучение нового материала. Первичная проверка усвоения знаний (10 минут).
4. Закрепление знаний (6 минут).
5. Контроль и самопроверка знаний (6 минут).
6. Подведение итогов урока (2 минуты).
7. Домашнее задание (1 минута).

ХОД ЗАНЯТИЯ:

1. Постановка темы и целей урока (1 минута).

Здравствуйте. Ребята, мы продолжаем с Вами изучать язык программирования Turbo Pascal. Сегодня на занятии мы вспомним, простые и структурированные типы данных, изученные ранее, структуру программы в Turbo Pascal, ввод – вывод данных, типы алгоритмов. А также познакомимся ещё с одним структурированным типом данных - строки (строковый тип данных). Изучим, процедуры и функции, которые можно выполнять над строками.

Слайд 1 (цели занятия)

2. Проверка усвоения изученного ранее материала (4 минуты).

У вас на столах лежат листочки, возьмите, пожалуйста, первый листочек, на нем написано «Задание 1». Сверху напишите свою фамилию. Ваша задача в течение трех минут найти ошибки в программе – это и будет повторение ранее пройденного материала. Обращайте внимание на правильность написания операторов, алгоритмических структур, объявления переменных, а также на структуру самой программы. Небольшая подсказка, ошибок 10. Начали.

А теперь берем следующий листочек, переворачиваем его. Вы видите все ошибки, которые были допущены при написании программы. Сверяем свое решение и ставим знак «+», если у вас такая ошибка отмечена.

Считаем количество плюсов. На экране вы видите критерии оценки. Ставим себе оценку и откладываем оба листочка в сторону. Пятерки есть?

Слайд 2 (критерии оценки)

3. Изучение нового материала. Первичная проверка усвоения знаний (15 минут).

Молодцы. Итак, в начале занятия я назвала вам тему, которую мы будем сегодня изучать. Какая тема? Правильно. «Строковый тип данных в Turbo Pascal».

Слайд 3 (тема)

Откройте тетради и запишите. Изучение новой темы мне хотелось бы начать с китайской поговорки: «Скажи мне и я забуду, покажи мне, и я запомню, дай мне действовать самому и я научусь».

Слайд 4 (поговорка)

Как вы ее понимаете? Да, конечно же. При изучении новой темы мы используем зрительную память для наглядности, а потом закрепляем полученные знания, решая задачи по теме.

У вас на столах, у каждого лежит опорный конспект занятия. Мы будем изучать тему вместе, используя этот конспект. Все примеры, какие-либо пояснения для себя, вы можете дописывать в этот конспект.

Слайд 5-22 (слайды по конспекту, ответы к заданиям)

«Строковый тип данных в Turbo Pascal».

Строка представляет собой последовательность символов. Для ее хранения можно использовать массив элементов символьного типа. Однако, многие действия со строками в этом случае (сравнение, ввод и т. д.), будут неудобны. Поэтому в языке Pascal существует строковый тип данных – String, который относится к стандартным структурированным типам данных.

Значение типа string представляет собой строку длиной до 255 символов, при этом ее нулевой элемент содержит информацию о ее текущей длине. При объявлении переменной, в квадратных скобках указывается максимальный размер строки. Если он не указан, то максимальная длина – 255 символов.

Переменная типа String объявляется в разделе переменных следующим образом:

- имя переменной: String;
- имя переменной: String[длина];

Примеры объявления переменной строкового типа:

- h: String;
- d: String[15];

Задание:

1. Переменная f – строка, состоящая из 28 символов.
2. Переменная g – строка неизвестной длины.

В тексте программы последовательность символов, являющейся строкой заключают в апострофы.

Например:

- parol:='Большой секрет';
- chislo:='123456';
- simvol:='&%45#@';

Поскольку строку можно рассматривать как одномерный массив символов, то к любому ее символу можно обращаться по его номеру.

Например:

- str1[i] – обращение к i-му элементу строки str1
- parol[4] – обращение к 4 элементу строки parol.

Задание:

1. Переменная `chislo` – обратиться к символу 5.
2. Переменная `simvolu` – обратиться к символу %.

Действия, процедуры и функции, которые можно производить над строками:

1. Ввод и вывод

Значение переменных строкового типа можно вводить с клавиатуры и отображать на экране с помощью операторов: `Write`, `Writeln`, `Read`, `Readln`.

Например:

```
Write ('введите строку ');  
Readln (s);
```

} Ввод строки с клавиатуры

```
Writeln (s);
```

} Вывод строки на экран

2. Присваивание значения

Присвоить значение переменной строкового типа можно с помощью соответствующего оператора присваивания - `:=`.

Например: `s:='Строка';`
`k:='Кот'; k[2]:='и';`

3. Объединение (сцепление)

В результате объединения нескольких строк, получается новая строка.

Например: `a:='Это строка, '+'которая '+'служит примером.';`
`a:= 'Это строка, ';`
`b:= 'которая ';`
`c:= 'служит примером.';`
`a:=a+b+c;`

4. Сравнение

При сравнении используются операторы сравнения: `=`, `<`, `>`, `<=`, `>=`, `<>`. Строки сравниваются посимвольно, начиная с первого символа. Если в одинаковых позициях находятся разные символы, большей считается та строка, у которой код символа больше. Если все символы сравниваемых строк одинаковы, то такие строки считаются равными.

Например:

- `'Balkon' < 'balkon';`
- `'кит' < 'кот';`
- `'символ' < 'символ.';`
- `'строка' = 'строка';`

5. Функция **Length**

Позволяет определить длину строки. У функции один параметр – переменная строкового типа.

Значение функции (целое число) – количество символов, из которых состоит строка.

Например: `n:=Length('Иванов');` `n=6`

Задание:

- `m:=Length(' Невский проспект '); m=`
- `k:=Length('gh yt hj kih!');` `k=`

6. Функция **Concat**

Объединяет несколько строк в одну.

Например: `s1:='Привет! ';`
`s2:='Как дела?';`
`s:=Concat(s1,s2);`

7. Функция **Copy**

Позволяет скопировать фрагмент строки. В общем виде обращение к функции `copy` выглядит так:

Переменная:= Copy (строка, p, n);

- строка — переменная строкового типа, содержащая строку, фрагмент которой надо получить;
- p — номер первого символа в строке, с которого начинается копирование подстроки;

n — длина выделяемой подстроки.

Например: st:='Приехали';
fam:=Copy(st,4,4);

8. Процедура **Delete**

Позволяет удалить часть строки. В общем виде обращение к процедуре выглядит так:

Delete (строка, p, n);

строка — переменная строкового типа;

p — номер символа, с которого начинается удаляемая подстрока;

n — длина удаляемой подстроки.

Например: s:='Город Каменск-Уральский'; Delete(s,14,10);

9. Процедура **Insert**

Позволяет вставить в строку подстроку. В общем виде обращение к процедуре выглядит так:

Insert (s1, s2, n);

s1 — переменная строкового типа (строка, которую нужно вставить);

s2 — переменная строкового типа (строка, в которую добавляется вставка);

n — позиция в строке начиная с которой делается вставка.

Например: s:='Добавление в строку';
s1:='подстроки';
Insert(s1,s,12);

10. Функция **Pos**

Позволяет определить положение подстроки в строке. Если в строке нет искомой подстроки, то функция **Pos** возвращает нуль. В общем виде обращение к функции выглядит так:

Переменная:=Pos(s1, s);

s1 — подстрока, которую надо найти в строке s.

Например: p:=Pos('ри','электричество'); p=6

Задание:

- n:=Pos('мен','Я супермен!'); n=
- k:=Pos('ро','умопомрачительный'); k=

4. Закрепление знаний (4 минуты).

Итак, мы разобрали основные процедуры и функции, которые можно выполнять со строковым типом данных. Взяли листочки с задачами. Необходимо вычислить результат выполнения программы.

Слайд 23-30 (слайды с задачами, ответы к задачам)

Как только вы определили результат, поднимаем руку.

5. Контроль и самопроверка знаний (5 минут).

Молодцы. Ну и чтобы окончательно убедиться в том, что вы усвоили тему, выполним тестовые задания. Подпишите, пожалуйста, листочки. Начали.

Слайд 31 (ответы к тестам)

6. Подведение итогов занятия (1 минута).

Давайте подведем итог занятия. Чему вы сегодня научились? А где в работе на ПК применяется строковый тип данных? Правильно, при проверке орфографии в любом редакторе. И закончить наше занятие я хочу словами Л.Н.Толстого «Ничто так глубоко и надолго не остается в памяти человека, как то, что приобретено усилием мысли».

Слайд 32 (девиз)

Спасибо за урок. Все свободны.