## Министерство образование и науки Российской Федерации

## Федеральное государственное бюджетное образовательное учреждение высшего

## профессионального образования

**«Московский государственный технический университет имени Н.Э.Баумана»**

**(МГТУ им. Н.Э.Баумана)**

## Отчет

## по педагогической практике:

**Тема:**

«Фунции, процедуры и записи в Delphi»

Выполнил студент: Перов Д.Ю. ИУ7-17

Москва, 2013

**Цели**

Ознакомление с функциями, процедурами и записями в языке Delphi.

Изучение вариантов использования

**Функции**

Описание функции состоит из заголовка и блока. Заголовок включает ключевое слово

Function, имя функции, список формальных параметров и тип возвращаемый функцией значения.

Заголовок имеет формат:

Function <имя> (формальные параметры): <Тип результата>

Переменная Result используется, чтобы хранить значение результата, возвращенное функцией.

Присваивая переменной Result значение необходимо учитывать тип результата объявленной функции.

Рассмотрим пример функции, которая будет возвращать куб числа, переданного как аргумент.

function cube(a: integer) : integer;

begin

result := a \* a \* a;

end;

И так чтобы вызвать функцию в теле основной программы необходимо указать имя и параметры нашей функции.

var

x: integer;

begin

x := cube(2);

WriteLn(x); // результат выполнения 8

**Процедуры**

Заголовок состоит из ключевого слова procedure, за которым следует имя процедуры и, при необходимости, список параметров, заключенных в круглые скобки:

procedure <Имя> [(параметры)];

Рассмотрим пример процедуры, которая будет выполнять печать строки.

procedure Print(str: string);

begin

writeln('"'+str+'"');

end;

Для использования данной процедуры в любом месте программы достаточно написать инструкцию вызова процедуры, состоящую из имени процедуры и списка аргументов, например:

Print('Hello'); //результат “Hello”

**Использование параметров**

Параметры в процедурах и функциях могут применяться не только по своему прямому предназначению - для передачи данных подпрограмме, но так же могут быть использованы для возвращения значений. При этом объявления параметров будет несколько отличатся, для того чтобы передать параметр, как переменную необходимо использовать ключевое слово var.

Рассмотрим пример процедуры, которая будет возводить в куб аргумент

procedure cube(var a: integer);

begin

a := a \* a \* a;

end;

Пример вызова:

var

x: integer;

begin

x:=2;

cube(x);

WriteLn(x); // результат выполнения 8

Контрольные вопросы:

1. Перечислите отличие функций от процедур?
2. Как изменить значение аргумента процедуры?
3. Приведите пример процедуры без параметров.

**Записи**

Запись - это составной тип данных, состоящий из фиксированного числа элементов одного или нескольких типов. Описание типа записи начинается словом record и заканчивается словом end. Между ними заключен список элементов, называемых полями, с указанием идентификаторов полей и типа каждого поля:

<название записи> = record

<имя поля 1> : <тип поля 1>;

...

<имя поля N> : <тип поля N>;

end;

Идентификаторы полей должны быть уникальными только в пределах записи. Допускается вложение записей друг в друга, т.е. поле записи может быть в свою очередь тоже записью.   
Чтобы получить в программе реальную запись, нужно создать переменную соответствующего типа.

type

TStudent = record

Name: string[35];

BirthYear: Integer;

end;

var

Stud: TStudent;

begin

Stud.Name := 'Ivanov';

Stud.BirthYear :=1995;

end;

Обращение к полям записи имеет довольно неудобный вид, поэтому довольно часто используется

оператор with, который имеет формат:

with <запись> do

<оператор>;

Например:

with Stud do

begin

Name := 'Ivanov';

BirthYear :=1995;

end;

Допускается применение оператора присваивания для всей записи в целом, при условии, что они имеют один и тот же тип.

Стоит заметить, что использование одной записи не целесообразно и зачастую применяют массив записей

type

arr = array [1..4] of TStudent

var

a: arr;

В данном случае каждый элемент массива будет иметь два ранее описанных поля: Name, BirthYear.

Для обращения к элементу массива записей необходимо указывать: номер элемента массива и ячейку записи.

a[1].Name := 'Ivanov';

a[1].BirthYear :=1995;

Контрольные вопросы:

1. Как объявить запись?
2. Возможно ли присвоить одной записи другую?
3. Как применять оператор with?