Урок информатики в 10б классе

Тема: Практическая работа по теме «Типы данных. Преобразование типов»

*Учитель информатики Сарсембаева Д.К.*

КГУ «Колутонская СШ»

2017-2018 учебный год

Тема урока: Практическая работа «Типы данных. Преобразование типов»

Цель урока: Закрепить на практике тему «Типы данных. Преобразование типов».

Задачи:

1. Учащиеся должны знать основные элементы языка программирования: переменные, их типы, основные операции и функции.
2. Учащиеся должны уметь применить теоретический материал на практике. Создать простейший калькулятор, который будет выполнять сложение, вычитание, произведение, деление целых чисел, введенных с клавиатуры.
3. Уметь самостоятельно работать.

Ход урока

1. Организационный момент.
2. Выявление имеющихся знаний
3. Практическая работа
4. Обобщение
5. Домашнее задание. (Повторить с курса 9 класса процедуры ввода-вывода).
6. Итоги

2. Выявление имеющихся знаний

* Перечислите известные вам типы данных.
* В чем заключается отличие переменных от констант?
* Как описывают переменные?
* Операторы DIV и MOD.
* Определите значение переменной х после выполнения операций:

1. Х:=8+16 MOD 3 (ответ 9)
2. Х:= Sqr(5 DIV 2/2) ответ 1

**Целочисленный тип данных**

Целочисленные типы данных могут занимать в памяти компьютера один, два, четыре или 8 байтов.

|  |  |  |
| --- | --- | --- |
| **Тип** | **Диапазон** | **Размер (байт)** |
| Shortint | -128 .. 127 | 1 |
| Integer | -32768 .. 32767 | 4 |
| Longint | -2147483648 .. 2147483647 | 4 |
| Byte | 0 .. 255 | 1 |
| Word | 0 .. 65535 | 2 |

**Вещественный тип данных**

Внутренне представление вещественного числа в памяти компьютера отличается от представления целого числа. Вещественное число представлено в экспоненциальной форме mE&plusmn;p, где m – мантисса (целое или дробное число с десятичной точкой), p – порядок (целое число).Чтобы перейти от экспоненциальной формы к обычному представлению числа, необходимо мантиссу умножить на десять в степени порядок.

Вещественное число может занимать от 4 до 10 байтов.

|  |  |  |  |
| --- | --- | --- | --- |
| **Вещественные типы** | **Диапазон** | **Кол-во значащих цифр** | **Размер, байт** |
| Real | 2.9e-39 ..1.7e+38 | 11-12 | 8 |
| Single | 1.5-e45 .. 3.4e+38 | 7-8 | 4 |
| Double | 5.0e-324.. 1.7e308 | 15-16 | 8 |
| Extended | 3.4e-4932 ..1.1e4932 | 19-20 | 8 |

Пример описания вещественных переменных:

Var

r1, r2: real; d: double;

**Основные алгебраические операторы языка программирования Free Pascal.**

|  |  |
| --- | --- |
| **Оператор** | **Действие** |
| + | Сложение |
| - | Вычитание |
| \* | Умножение |
| / | Деление |
| DIV | Целочисленное деление |
| MOD | Вычисление остатка от деления |

Операторы целочисленной арифметики DIV и MOD применяются только к целочисленным операндам.

DIV позволяет получать целую часть результата деления одного числа на другое.

Например, 15 DIV 7 = 2.

Оператор  MOD  получает остаток от деления одного числа на другое.

Например, 15 MOD 7 = 1, Для задания нужного порядка выполнения операций в выражении можно использовать скобки.

**Стандартные функции**

В языке определены стандартные функции. Некоторые арифметические функции представлены в таблице ниже.

|  |  |
| --- | --- |
| **Обозначение** | **Действие** |
| Abs(*n*) | *Абсолютное значение n.* |
| Sqrt(*n*) | *Квадратный корень из n.* |
| Sqr(n) | *Квадрат n.* |
| Exp(*n*) | *Экспонента n*. |
| Ln(n) | *Натуральный логарифм n.* |
| Random(n) | *Случайное целое число в диапазоне от 0 до n-1. (перед первым обращением к функции необходимо вызвать функцию*Randomize*, которая выполнит инициализацию программного генератора случайных чисел)* |
| Sin() | *Синус выраженного в радианах угла* |
| Cos() | *Косинус выраженного в радианах угла* |
| Arctan() | *Арктангенс выраженного в радианах угла* |

Величина угла тригонометрических функций должна быть выражена в радианах. Для преобразования величины угла из градусов в радианы используется формула:

(α \* 3.1415256 ) / 180,

где α – величина угла в градусах, 3.1415256 –число π.

Вместо константы 3.1415256 можно использовать стандартную именованную константу PI.

**Функции преобразования**

Функции преобразования типов часто используются при  вводе и выводе информации

Например, для того чтобы вывести в поле вывода (компонент Label) диалогового окна значение переменной Real, необходимо преобразовать число в строку символов, изображающую данное число. Это можно сделать при помощи функции FloatToStr, которая возвращает строковое представление значения выражения, указанного в качестве параметра функции.

Основные функции преобразования типов

|  |  |
| --- | --- |
| **Обозначение** | **Действие** |
| Chr(n) | *Символ, код которого равен n.* |
| IntToStr(k) | *Строка, являющаяся изображением целого k.* |
| FloatToStr(n) | *Строка, являющаяся изображением вещественного n.* |
| FloatToStrF(n,f,k,m) | *Строка, являющаяся изображением вещественного n. При вызове функции указывают:*f *- формат;*k *– точностью (общее количество цифр);*m *- количество цифр после десятичной точки.*Возможны следующие значения параметра Format: ffGeneral – общий числовой формат; ffFixed – фиксированный формат; ffCurrency – денежный формат. |
| StrToInt(s) | *Целое, изображением которого является строка*s*.* |
| StrToFloat(*s*) | *Вещественное, изображением, которого является строка*s*.* |
| Round(*n*) | *Целое, полученное путем округления*n *по известным правилам.* |
| Trunc(n) | *Целое, полученное путем отбрасывания дробной части*n*.* |
| Frac(n) | *Дробное, представляющее собой дробную часть вещественного*n*.* |
| Int(*n*) | *Дробное, представляющее собой целую часть вещественного*n*.* |

Наиболее часто программа может получать исходные данные из окна ввода или из поля редактирования (компонент Edit). Для преобразования данных в числовой тип используют соответствующую функцию.

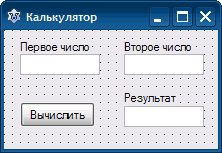
Например, для преобразования в вещественный тип используется оператор:

Funt:= StrToFloat(Edit1.Text);

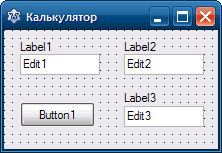
Основные сведения об переменных,  числовых типах данных и функциях преобразования типов мы повторили. Теперь переходим к практической части нашего занятия.

**Проект «Калькулятор»**

**Задание.** Создать приложение для вычисления суммы двух чисел. Окно программы должно выглядеть так, как показано на рисунке.



**Ход выполнения**

1. Создайте приложение. Сохраните проект в папке Калькулятор.
2. Создайте интерфейс по образцу.  
   
3. Настройте свойства объектов в соответствии с таблицей.

|  |  |  |
| --- | --- | --- |
| **Компонент** | **Свойство** | **Значение** |
| Form1 | Caption | Калькулятор |
| Label1 | Caption | Первое число |
| Label2 | Caption | Второе число |
| Label3 | Caption | Результат |
| Edit1 | Text  Name | Пусто  EditNum1 |
| Edit2 | Text  Name | Пусто  EditNum2 |
| Edit3 | Text  Name | Пусто  EditResult |
| Button1 | Coption | Вычислить |

1. Написать процедуру обработки щелчка на кнопке Вычислить (btnMath).Для этого выполните двойной щелчок на кнопке Вычислить. Это приведет к созданию  процедуры TForm1.Button1Clic в разделе implementation:
2. procedure TForm1.Button1Click(Sender: TObject);
3. begin

end;

Понятно, что созданная процедура не содержит ни одной команды. Ваша задача – заполнить шаблон операторами. Все команды, указанные в процедуре между begin и end, будут выполнены при щелчке на кнопке Выполнить.

В нашем случае процедура TForm1 . But ton1Cl ick будет иметь вид:

procedure TForm1.Button1Click(Sender: TObject);

var num1, num2, result: integer;

begin

num1:=StrToInt(editNum1.text);

num2:=StrToInt(editNum2.text);

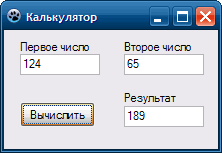
result:=num1+num2;

EditResult.text:=IntToStr(result);

end;

Обратите внимание, что были написаны всего 5 команд, предназначенных для решения поставленной задачи. Остальной текст в окне редактора создается автоматически

1. Сохранить проект. Проверьте работу приложения. Введите в первые два поля  целые числа, нажмите кнопку Вычислить. В поле результат должна отобразиться сумма этих чисел.



Задание 2.

Создать калькулятор выполняющий +,-,\*,/.