1. Ввод слова или знака пунктуации
2. Выведение в единственном компоненте вывода

Нажатие кнопки «Количество символов» -вывод количество символов если введено слово

Нажатие кнопки «Количество символов» -вывод уведомления если введено знак пунктуации

Нажатие кнопки «Наличие слова» -вывод даты если введен знак пунктуации

Нажатие кнопки «Наличие слова» -вывод уведомления что такое слово находится в стихе

Требования

1. Программа должна отличать слово от знака пунктуации и других символов
2. Программа должна найти слово в стихе
3. Выводить дату
4. Выводить уведомления в случаи некорректного ввода

Вариант № ll

единственный компонент вывода:

■ за нажатием кнопки "Количество символов" - количество введенных пользователем букв, если пользователем введено слово; сообщение "Введен знак пунктуации", если введено знак пунктуации;

■ по нажатию кнопки "ДатаУНаявнисть слова" - текущую дату, если пользователем введено знак пунктуации; иначе - сообщение, есть ли введено слово в следующей части стихотворения Анатолия Тарана "Удачи в Вашей комнате, мама":

Удачи в Вашей комнате, мама, всем, кто переступит Ваш порог: Добрым людям, и птицам так же, и коту, что в теплые прилег;

Удачи кожненький древесине, что в дом ветви Притула,

Удачи маленькой Травин И ветерка, прилет издали.

- Задача 1.2 -

Пользователь вводит числовые значения х, у, г. Вычислить и вывести Q ".

| \ Интерфейсом (визуальное программирование).

112 ^ ■ Язык интерфейса приложения, запросов и сообщений - украинский.

■ Программный реализацию задач .следует осуществить с использованием стандартных процедур и функций (при необходимости подключить соответствующие модули в разделе uses и компоненты).

■ Для организации ввода исходных данных рекомендуется использовать визуальный компонент Edit (свойство / поле / Text). \*

■ Для вывода результатов работы приложения рекомендуется использовать компонент Label (свойство / поле / Caption).

■ Для преобразования типов следует применять стандартные функции преобразования типов модуля SysUtils: StrToInt () ■, IntToStr (,) - ь

и / St ^ TpEloat (jgl FloatToStr () TextToFloat () ;, y \* J | oatToText (ijj

| Upate | pStEs ()., TimeJCoStr () DateTimeToStr () и ДР.

■ Значение локальной даты, времени ОС можно получить с помощью .стандартних функций модуля SysUtils date, time или now (тип результата - TDateTime).

Рекомендательные инструкции к визуального программирования в Lazarus, Delphi

■ Процесс создания графического приложения состоит из: 1) проектирование интерфейса приложения (формы и визуальных компонентов, которые на ней располагаются) 2) написание программного кода обработки событий.

■ Для создания графического приложения в Lazarus / Delphi следует: в с помощью испекшра объектов заполнишь поле Caption свойств

главной формы Forml: добавить название программы, ГПБ разработчика и тому подобное; в с панели компонентов добавить на форму необходимые визуальные компонента: надписи Label, кнопки Button, поля ввода Edi t, Memo и тому подобное; в соответствии с требованиями по интерфейса, должным расположить добавлены объекты на форме, установишь их размеры тошо; в заполнишь соответствующие поля свойств добавленных на форму объектов (название, текст, цвет, шрифт тошо).

в если действие обработки данных необходимы дополнительные переменные, их следует объяснишь: глобальные - в разделе описания глобальных переменных var главного модуля (создается автоматически EDE), локальные - добавить раздел var в соответствующей процедуре или функции;

в доступ к свойствам (атрибутов) объектов Label, Memo, Edit осуществляется следующим образом:

// Процедура обработки события нажатия кнопки Buttonl procedure TForml. ButtonlClick (Sender: TObject);

var

str: string;

// Вывод строки в объект Labell

Label 1. Caption if 'КНТУ | Кафедра ПЗИ ';

Labell-Caption = Editl.Text;

// Вывод строки в объект Editl Editl.Text = 'Модульное программирование ";

Editl.Text = Labell-Caption;

// Чтение строки из объекта Editl

И / чтения всего текста с объектами объекта Мешои в переменную

str = Memol.Lines.GetText;

Editl.Text; = Memol-Lines-GetText;

// Чтение с Memol 1-й строки

str; = MemolI Lines [OJ;

Labell.Caption = Memol-Lines [0];

// Запись строки в 2-ю строчку Memol

Memol.Lines [1]; = Editl.Text;

// Определение количества имеющихся строк в Memol Memol.Lines [0] Щ IntToStr (Memol.Lines-Count)

// Добавление строки в конец имеющегося текста в Memol

fMemol.Lines.Add ( 'Новая строка ... ")

Memol I Lines .Add (DateTimeToStr (now)) // Вывод лок. даты и времени ос ирешОИ и Lines .Add И DateToStr (date)) // Вывод локальной дать ос

Memol. Lines - Add (TimeToStr (time)) // Вывод локального времени ос

// Вставка строки в Memol

Memol.Lines.Insert (2, 'Эта строка будет третьим.');

end;