## «Разработка проекта ВАШ ВЕС»

## Постановка задачи

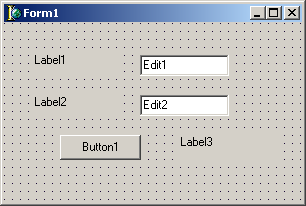
Пусть оптимальный вес человека определяется как рост минус 100. Если фактический вес человека меньше оптимального, то будем считать его худым, если больше, то полным.

|  |  |
| --- | --- |
| 1 | Создать программу, с помощью которой можно выполнить:  Введя свой рост и фактический вес, и нажав кнопку **Расчет**, учащийся может определить, худой ты или полный и насколько надо поправиться или похудеть. |

|  |
| --- |
| Новым в этой работе являются:   * использование типов переменных - целочисленного и действительного (**integer** и **real**); * преобразование строковых данных в числовые и числовые в строковые с помощью функций **StrToInt**; **StrToFloat**, **IntToStr** **FloatToStr** ; **Format()**; * обработка исключительных ситуаций с помощью оператора  **Try – except – end;** * использование процедуры **ShowMessage** для вывода сообщения в отдельном окне. |

## План разработки программы, №6

1. Открыть новый проект.
2. Разместить в форме экземпляры компонентов в соответствии с рисунком. В **Edit1** будем вводить вес в кг, а в **Edit2** – рост в см.



1. Сохранить код программы и проект под именами, например, **Unit5.pas** и **Pr5.dpr**.
2. Ввести в раздел **VAR** переменные для сохранения значений:

фактического веса (faktW),   
оптимального веса (optW),   
роста (Rost)   
разницы между оптимальным весом и фактическим (Delta).

В начале будем считать, что все данные у нас целые числа:

VAR  
factW, optW, Rost, Delta : integer;

1. Создать процедуру, реагирующую на щелчок по кнопке **Button1**, и заполнить ее следующим кодом:

|  |  |  |  |
| --- | --- | --- | --- |
| Выделенный объект | Вкладка окна **Object Inspector** | Имя свойства/ Имя события | Значение/Действие |
| Button1 | Events | OnClick | factW := StrToInt(Edit1.text); Rost := StrToInt(Edit2.Text); OptW :=Rost - 100; Delta := abs(factW - OptW); if OptW = factW then  Label3.caption := 'Ваш вес идеален!' else  if OptW > factW then  Label3.caption := 'Вам надо поправиться на '+IntToStr(Delta)+' кг.'  else Label3.caption := 'Вам надо похудеть на '+IntToStr(Delta)+' кг.' |

Пояснение  
Функция **StrToInt** преобразует строку в целое число, функция **IntToStr** выполняет обратное действие – целое число преобразует в строку.

1. Усовершенствовать программу так, чтобы можно было бы вводить любые десятичные величины. Для этого необходимо использовать вещественный тип переменных **Real**:

VAR  
factW, optW, Rost, Delta : real;

Преобразование действительных чисел в строковый тип и строковый тип в действительное число выполняется с помощью функций: **FloatToStr** и **StrToFloat**. Внесите соответствующие изменения в обработку события **OnClick** компонента **Button1**.

1. Сохранить проект окончательно, запустить и протестировать его.

## Задание для самостоятельного выполнения, №6

|  |  |  |
| --- | --- | --- |
|  | Задание | Подсказка |
| **1** | Для форматирования и преобразования действительных чисел вместо функции **FloatToStr**() применить функцию **Format(‘ %f ’,[пeременная]).**  Например, Format(‘%f ‘, [Delta]) |  |
| **2** | Для форматирования и преобразования применить эту же функцию в таком виде **Format(‘Строка %f’,[пeременная])**.  Например, Format(‘Вам надо похудеть %f',[Delta]) . |  |
| **3** | Усовершенствовать проект:   * сделать к программе заголовок; * сделать шрифт выводимой реплики отличным от стандартного по виду, цвету и размеру; * вставить кнопку выхода из программы; * предусмотреть возможность повторного запуска программы (см. проект Диалог). |  |
| **4** | Сделайте так, чтобы в начале программы или после повторного запуска объекты **Label2** и **Edit2** были не видны и появлялись бы на экране только после того, как будет введен вес. | Свойству **Visible** нужно присвоить **False**. |

**5.** В случае преобразования строкового типа в числовой тип может возникнуть ситуация появления ошибки, если введены недопустимые символы. Если функции **StrToInt** или **StrToFloat** обнаружат ошибку в записи числа, они инициируют так называемую исключительную ситуацию (исключений), которая обычно приводит к аварийному завершению работы программы.

Обработчик исключений строится в виде таких конструкций:

Try   
<защищенный блок операторов>  
except  
<обработка исключений>  
end;

Если при выполнении операторов из защищенного блока возникнет исключение, управление будет передано в блок операторов, располагающийся между **except** и **end**, но если обработка пройдет без ошибок, блок исключений игнорируется и управление передается оператору, следующему за **end**.

Пример использования обработки исключений для процедуры **Edit1KeyPressed** может выглядеть так:

try  
FactW:=StrToInt(Edit1.Text);  
except  
ShowMessage('Ошибочная запись числа: ' + Edit1.Text);  
Edit1.SetFocus;  
Exit;  
end;

В результате выполнения

FactW:=StrToInt(Edit1.Text);

если возникнет исключительная ситуация, то на экране появится окно с текстовым сообщением и кнопкой **ОК**. После появления окна работа программы приостановится в ожидании реакции пользователя. При вызове стандартной процедуры **Exit** снова активизируется редактор, в котором обнаружен ошибочный текст.

Внесите необходимые изменения для обработки исключительных ситуаций, возникающих при вводе чисел.

Примечание  
При работе в среде Delphi эксперименты с исключениями плохо прослеживаются, так как при каждом исключении среда перехватывает управление программой. Для отмены этого необходимо в команде **Tools/Debugger Options/Language Exceptions** убрать флаг у опции **Stop on Delphi Exceptions.**

**6.** Измените алгоритм расчета с учетом Индекса массы тела.

Вес – X,  
Рост – Y.  
Индекс массы тела – A, где A = X / Y^2\*10000 (кг/м2)

Результат определяется по таблице:

|  |  |  |
| --- | --- | --- |
|  | Значение индекса | Результат  (сообщение, которое надо вывести) |
| 1 | A < 18 | Большой недовес |
| 2 | 18 <= A < 20 | Маловато и небезопасно, можно получить истощение |
| 3 | 20 <= A <= 25 | Идеально |
| 4 | 26 <=A <=30 | Легкий недобор |
| 5 | 30 < A | Срочно нужно худеть |

## Листинг программы, № 6

Для самоконтроля ниже приводится базовый текст программы.

|  |
| --- |
| unit Unit5;  interface  uses Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs, StdCtrls, Buttons;  type TForm1 = class(TForm) Label1: TLabel; Label2: TLabel; Edit1: TEdit; Edit2: TEdit; Button1: TButton; Label3: TLabel; BitBtn1: TBitBtn; BitBtn2: TBitBtn; procedure Button1Click(Sender: TObject); procedure BitBtn2Click(Sender: TObject); procedure Edit1Change(Sender: TObject); private { Private declarations } public { Public declarations } end;  var  Form1: TForm1;  implementation  {$R \*.DFM}  procedure TForm1.Button1Click(Sender: TObject); var A,factW,optW,Rost,Delta:Real; begin try factW:=StrToFloat(Edit1.Text); except ShowMessage('Ошибочная запись числа: '+Edit1.Text); Edit1.SetFocus; Exit; end; try Rost:=StrToFloat(Edit2.Text); except ShowMessage('Ошибочная запись числа: '+Edit2.Text); Edit2.SetFocus; Exit; end; OptW:=Rost-100; Delta:=abs(factW-OptW); if OptW=factW then Label3.Caption:='Ваш вес идеален' else if optW>FactW then Label3.Caption:=Format('Вам надо поправиться на %f кг.',[Delta]) else Label3.Caption:=Format('Вам надо похудеть на %f кг.',[Delta]); A:=(factW\*10000)/(Rost\*Rost); if A<18 then ShowMessage('Жуткий недовес') else if (A>=18) and (A<20) then ShowMessage('Маловато и небезопасно, можно получить истощение') else if (A>=20) and (A<=25) then ShowMessage('Идеально') else if (A>=25) and (A<=30) then ShowMessage('Легкий перебор') else if (A>30) then ShowMessage('Срочно нужно худеть'); end;  procedure TForm1.BitBtn2Click(Sender: TObject); begin Edit1.Text:=''; Edit2.Text:=''; Edit1.SetFocus; Label3.Caption:=''; Label2.Visible:=false; Edit2.Visible:=false; end;  procedure TForm1.Edit1Change(Sender: TObject); begin Edit2.Visible:=true; Label2.Visible:=true; end;  end. |