

Частное учреждение  
профессиональная образовательная организация  
ТЕХНИКУМ «БИЗНЕС И ПРАВО»

ЗАЩИЩЕНО С ОЦЕНКОЙ

\_\_\_\_\_  
\_\_\_\_\_  
(подпись и ФИО руководителя)  
«\_\_\_\_\_» \_\_\_\_\_ 20\_\_ г.

**ОТЧЕТ О ВЫПОЛНЕНИИ ИНДИВИДУАЛЬНОГО ЗАДАНИЯ  
ПО УЧЕБНОЙ ПРАКТИКЕ**

по ПМ.01	Разработка программных модулей программного обеспечения для компьютерных систем
Выполнил	Фамилия И.О.
Курс, группа	3 курс, группа П-3
Специальность	09.02.03 Программирование в компьютерных системах
Руководитель практики	Кукса Н.П.

г. Белореченск  
2019

## СОДЕРЖАНИЕ

ПОСТАНОВКА ЗАДАЧИ	3
СТРУКТУРА ПРОЕКТА	4
ПРОЕКТИРОВАНИЕ ЭКРАННОЙ ФОРМЫ	5
РАЗРАБОТКА ПРОГРАММНОГО КОДА	6
РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ	13
ПРИЛОЖЕНИЕ	16

## ПОСТАНОВКА ЗАДАЧИ

Написать программу, при помощи которой можно проконтролировать умение устно складывать два числа, умножать и вычитать первое из второго. Пользователь выбирает арифметическое действие. Программа должна предложить испытуемому 10 примеров и по окончании процесса тестирования выставить оценку. На решение каждого примера отводится не более 15 секунд (ход времени отображается с помощью ProgressBar). Компонент Label2 используется для вывода примера, Label3 — для вывода сообщения об ошибке и выводится верный результат, Edit2 – для ввода результата пользователем, кнопка ОК – для перехода к следующему примеру. Оценка тестирования выводится в отдельной форме. Также предусмотрен вывод справочной информации и «О программе» (AboutBox) с указанием автора, группы, учебного заведения, года выполнения работы. Рекомендуемый вид формы показан на рисунке 1. Оформить интерфейс.

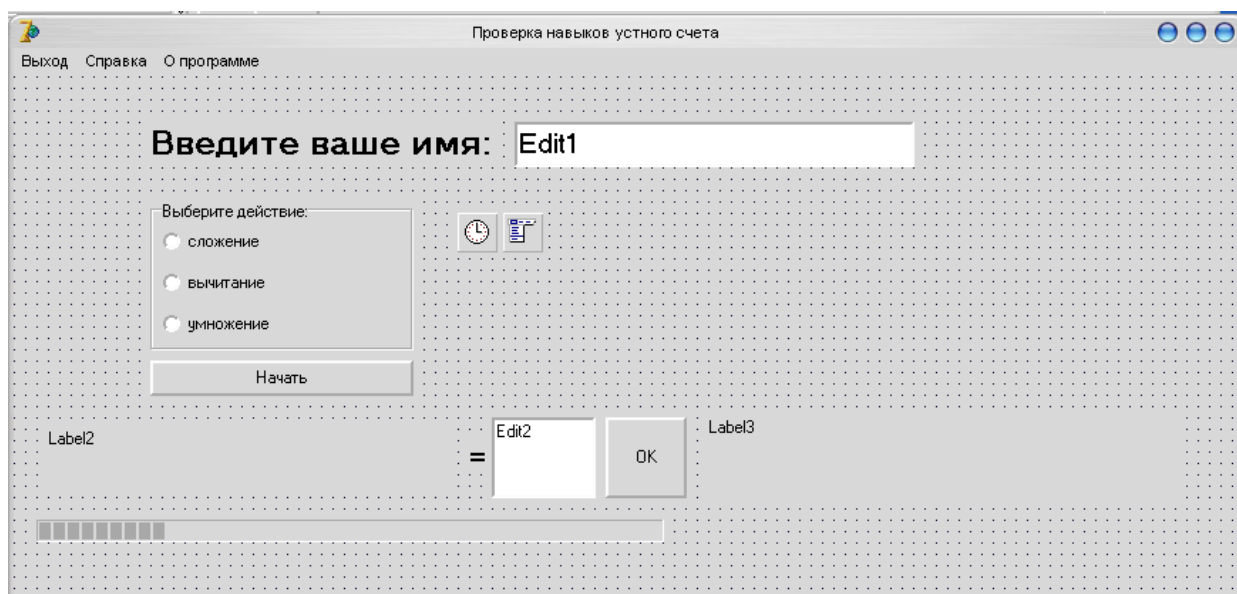


Рисунок 1 – Форма программы проверки умения устного счета

## СТРУКТУРА ПРОЕКТА

Структура проекта представлена на рисунке 2.

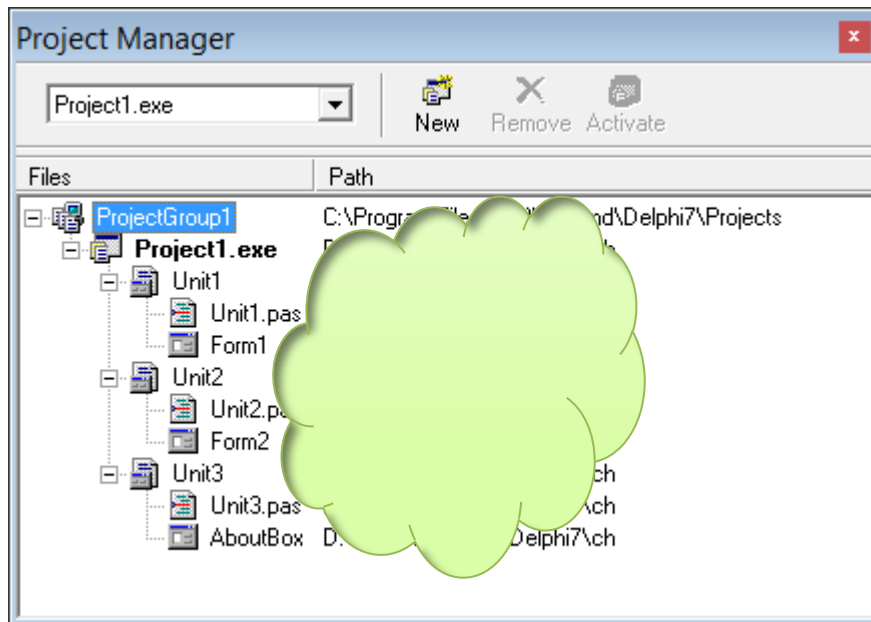


Рисунок 2 – Структура проекта

Проект состоит из трех форм:

- главной формы;
- формы результатов;
- формы «О программе».

Работа программы начинается с запуска главной формы.

## ПРОЕКТИРОВАНИЕ ЭКРАННОЙ ФОРМЫ

Экранная форма имеет следующий вид, представленный на рисунке 3.

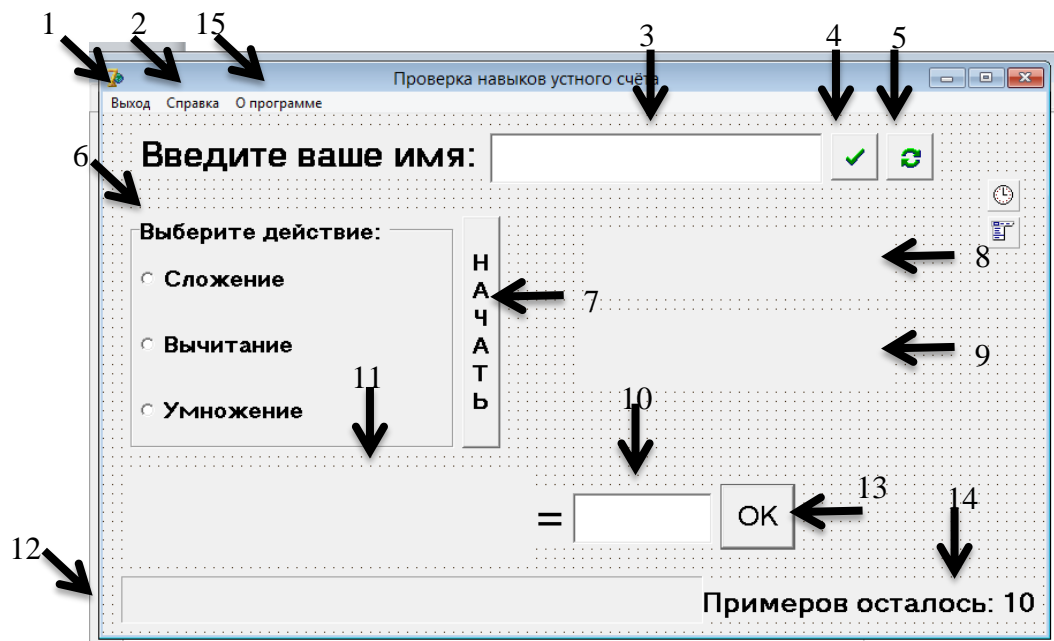


Рисунок 3 – Экранная форма с визуальными элементами

Здесь:

1. Кнопка выхода, элемент компонента MainMenu
2. Кнопка справки, открывает HTML-справку, элемент MainMenu
3. Edit1 - элемент ввода текста, в данном случае – имени пользователя
4. BitBtn1 – кнопка для завершения ввода имени и перехода к основной программе
5. BitBtn2 – кнопка для повторения программы с самого начала
6. RadioGroup1 – группа RadioButton для выбора типа задач (сложение, вычитание, умножение)
7. Кнопка старта, после появления задач преобразуется в кнопку «Пропустить»
8. Элемент Label4 – в нём выводится результат проверки ответа (Правильно/Неправильно)
9. Элемент Label5 – при неверном ответе в нём выводится верный ответ
10. Элемент Edit2 – используется для ввода ответа на пример
11. Элемент Label2 – выводит пример

12. ProgressBar1 – визуально отображается время, которое осталось пользователю на решение примера (15 секунд)
13. Кнопка «ОК» - подтверждение ответа и отправка его на проверку
14. Элемент Label6 – отображает количество примеров, которые ещё нужно решить пользователю
15. Кнопка «О программе», открывает новое окно с информацией о разработчике программы, элемент компонента MainMenu

## РАЗРАБОТКА ПРОГРАММНОГО КОДА

Глобальные переменные представлены на рисунке 4.

```
var  
  Form1: TForm1;  
  uName:string;  
  a,b,res,k,t,f: integer;  
  s:char;
```

Рисунок 4 –Глобальные переменные

Здесь a, b – переменные для записи значений выражения, res – результат операции над ними, k – количество пройденных примеров, t – количество верных ответов, f – количество неверных, s – символ операции над переменными a и b (+,-,\*).

Процедура создания формы представлена на рисунке 5.

```
procedure TForm1.FormCreate(Sender: TObject)  
begin  
  Random  
  
  //496  
  //815  
  //false;  
  //false;  
end;
```

Рисунок 5 – Код обработчика создания формы

Здесь реализовано подключение модуля Randomize (используется для генерации случайных чисел), а также задание начальных значений элементов формы.

Процедура генерации примеров и проверки на завершение генерации показана на рисунке 6.

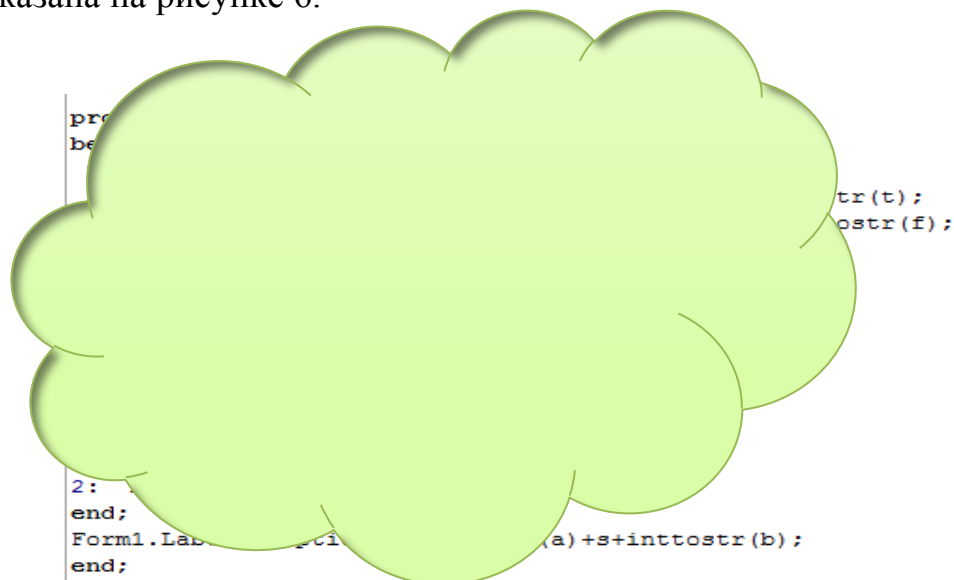


Рисунок 6 – Процедура генерации примеров

Процедура нажатия на кнопку завершения ввода имени (а также проверка корректности введённых данных) и перехода к основной части программы показана на рисунке 7.



Рисунок 7 – Процедура нажатия на кнопку завершения ввода имени

Процедура, которая разблокирует кнопку начала тестирования, при выборе одного элемента из группы RadioGroup показана на рисунке 8.

```
procedure TForm1.Button1Click(Sender: TObject);  
begin  
  ...  
end;
```

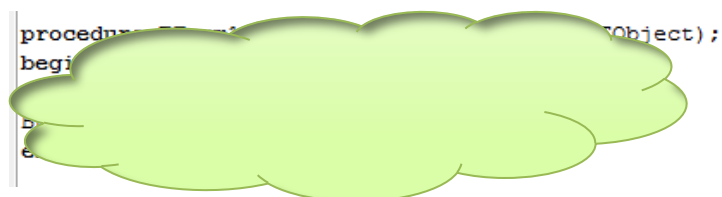


Рисунок 8 – Процедура обработки выбора элемента из RadioGroup

Процедура нажатия на кнопку «НАЧАТЬ» и «ПРОПУСК» (которые соответственно запускают процесс тестирования или пропускают текущий пример, в зависимости от стадии работы программы) показана на рисунке 9.

```
procedure TForm1.Button1Click(Sender: TObject);  
begin  
  ...  
end;
```

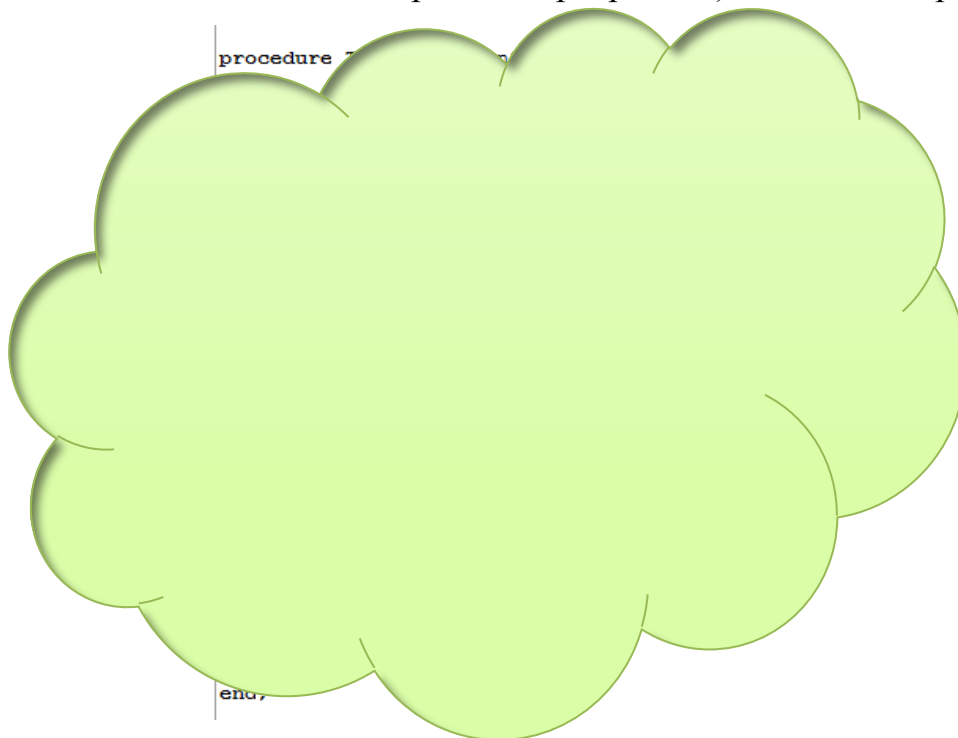


Рисунок 9 – Процедура нажатия на кнопку «НАЧАТЬ»/ «ПРОПУСК»

В процессе тестирования кнопка имеет значение «ПРОПУСК», тогда при её нажатии текущий пример будет засчитан как неправильный и программа перейдёт к следующему. В начале работы программы, когда тестирование ещё не началось, эта кнопка запускает процесс генерации заданий.



Процедура обработки ответа, при нажатии на кнопку «ОК» (проверка ответа, вывод сообщения о корректности/некорректности ответа на экран, а также при неверном результате, показывается правильный ответ) показана на рисунке 10.



Рисунок 10 – Процедура нажатия на кнопку «ОК»

При равенстве введённого значения переменной *res* (заранее рассчитанному ответу на выражение), программа выводит сообщение о том, что ответ на пример верен и засчитывает его как верный (в переменную *t*), иначе сообщает о неправильном ответе, выводит корректный ответ и относит его к неверным (переменная *f*).

Процедура завершения работы программы, при нажатии пункта меню «Выход» (закрытие основной формы) изображено на рисунке 11.

```
procedure TForm1.N1Click(Sender: TObject);
begin
  close;
end;
```

Рисунок 11 – Процедура нажатия на кнопку «Выход»

Процедура вывода справки показана на рисунке 12.

```
procedure TForm1.N2Click(Sender: TObject);  
begin  
WinExec('hh Ooo.chm', SW_RESTORE);  
end;
```

Рисунок 12 – Процедура нажатия на кнопку «Справка»

Процедура вывода информации о программе показана на рисунке 13.

```
procedure TForm1.N3Click(Sender: TObject);  
begin  
AboutBox.Show;  
end;
```

Рисунок 13 – Процедура нажатия на кнопку «О программе»

Процедура перезапуска программы, возвращение к исходному состоянию (сброс всех пользовательских действий, обнуление переменных, откат к состоянию первоначального запуска) показана на рисунке 14.

```
procedure TForm1.BitBtn1Click(Sender: TObject);  
begin  
ProgramBar1.  
Le  
P  
Re  
For  
Form1.  
Timer1.Enabled:=  
end;
```



Рисунок 14 – Процедура нажатия на кнопку повтора

Процедура заполнения ProgressBar и проверки на заполнение (истечение времени (отсчитывается с помощью компонента Timer), положенного на решение примера – 15 секунд, при этом ответ засчитывается как неверный и тестирование продолжается) показана на рисунке 15.

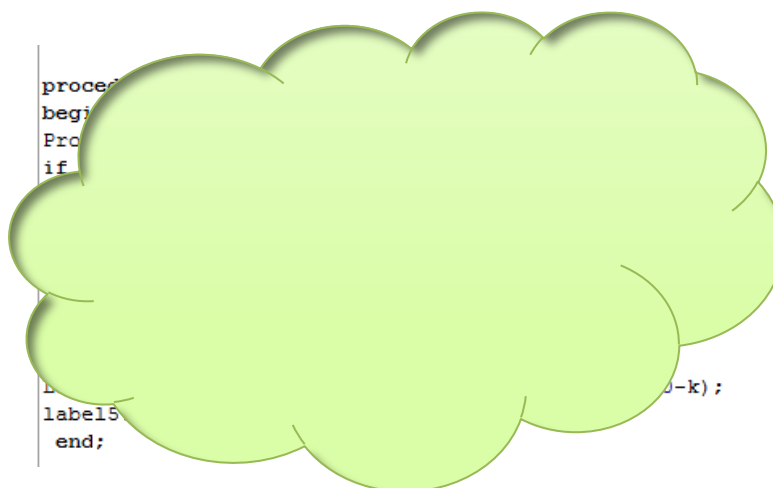


Рисунок 15 – Процедура заполнения ProgressBar

Процедура проверки ввода в поле для ответа (запрет всех символов, кроме цифр, кнопки BackSpace и символа минуса) показана на рисунке 16.

```
procedure TForm1.Edit2KeyPress(Sender: TObject; var Key: Char);  
begin  
if not(key in ['0'..'9', '-', #8]) then key:=#0;  
end;
```

Рисунок 16 – Процедура проверки вводимых символов

Для создания файла HTML-справки необходимо воспользоваться программой HTML Help Workshop. В ней создаются отдельные страницы справки на языке HTML, которые изображены на рисунке 17.

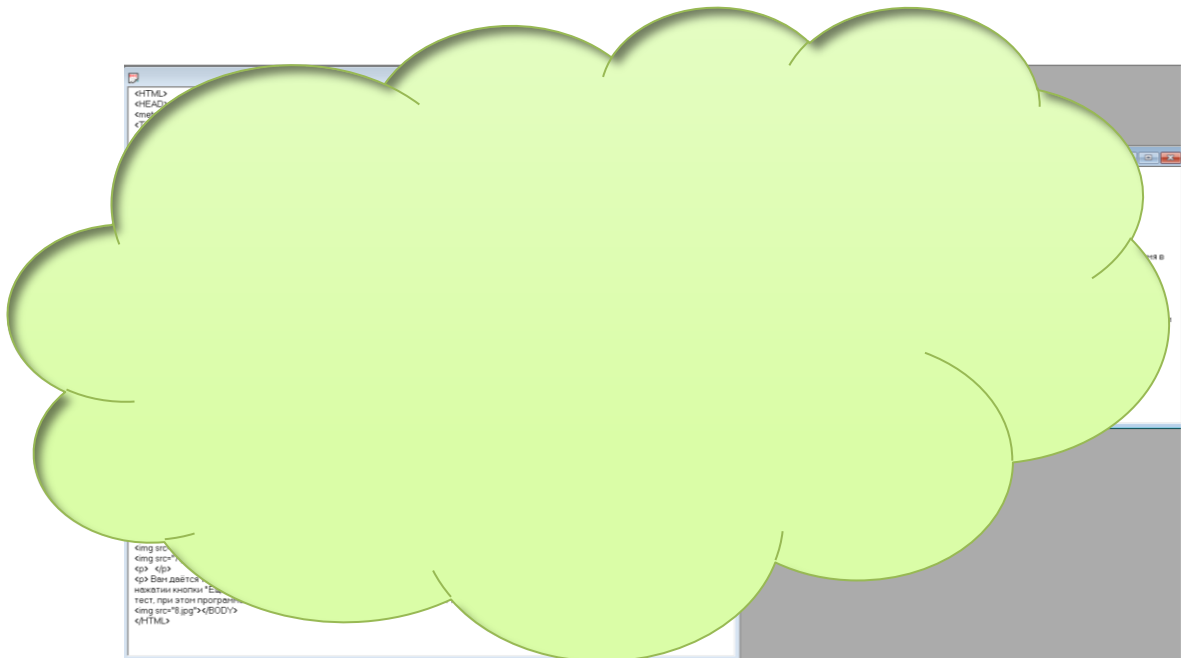


Рисунок 17 – Страницы справки в исходном виде

После компиляции и связки этих страниц окно справки будет выглядеть, как показано на рисунке 18.

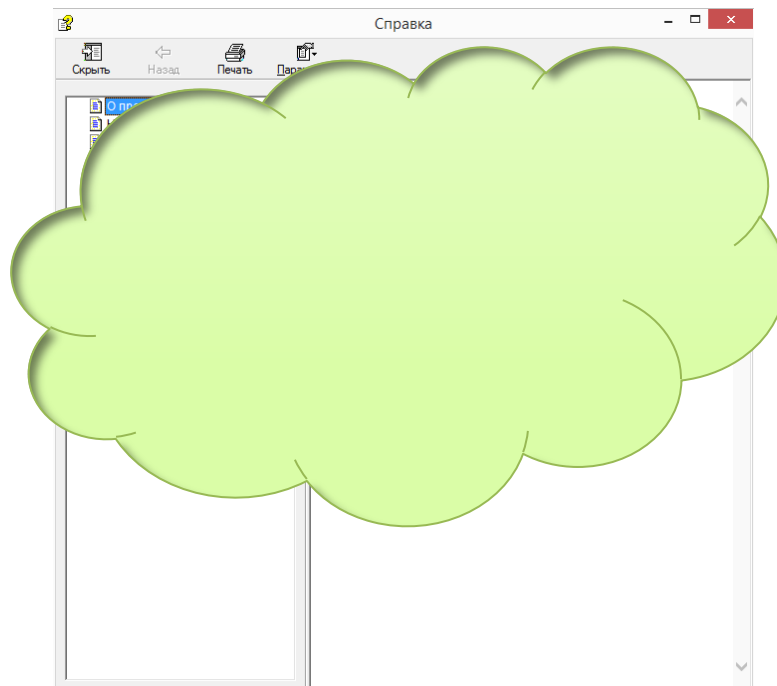


Рисунок 18 – Компилированное окно справки

После того, как была составлена форма с элементами, написан код программы и создана справка, можно запускать готовое к работе приложение.

## РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

Для запуска программы "Проверка навыков устного счёта" необходимо ввести своё имя в поле ввода и нажать кнопку с галочкой (рисунок 19), при пустом поле программа выдаст сообщение о том, что введено некорректное имя пользователя (рисунок 20).



Рисунок 19 – Поле ввода имени

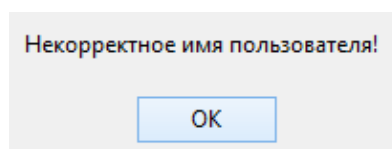


Рисунок 20 – Окно ошибки при некорректном вводе имени

После этого появится окно выбора действия (рисунок 21). Необходимо кликнуть на желаемый тип проверки. Затем станет доступной кнопка "Начать", кликнуть на неё, тогда тестирование начнётся (рисунок 22).

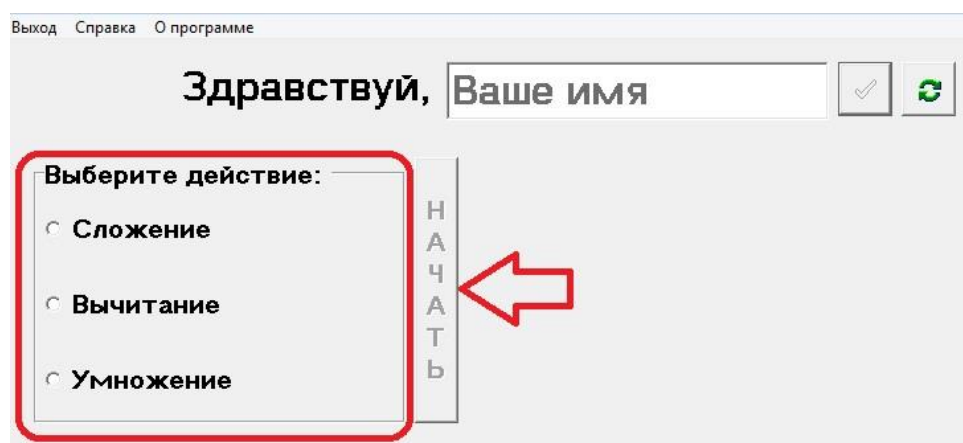


Рисунок 21 – Процесс выбора типа тестирования и начало тестирования



Рисунок 22 – Работающая программа

Здесь:

- 1- Кнопка выхода, закрывает программу
- 2- Кнопка вывода информации о программе
- 3- Кнопка повтора, сбрасывает все пользовательские действия
- 4- Кнопка ввода ответа, нажимайте на неё только после ввода ответа
- 5- Поле ввода, сюда вводится ваш ответ
- 6- Количество оставшихся примеров для решения
- 7- Выражение, которое необходимо вычислить
- 8- Панель времени, когда она заполнится (15 секунд), задание засчитается как неправильное

9- Пропуск примера, засчитывается как неправильный

В поле ввода (5) необходимо ввести ответ на выражение, находящийся слева от него (7), после чего выше. Если ответ окажется неправильным, то программа выдаст сообщение об этом и покажет правильный ответ (рисунок 23), иначе просто выдаст то, что ответ верный (рисунок 24).

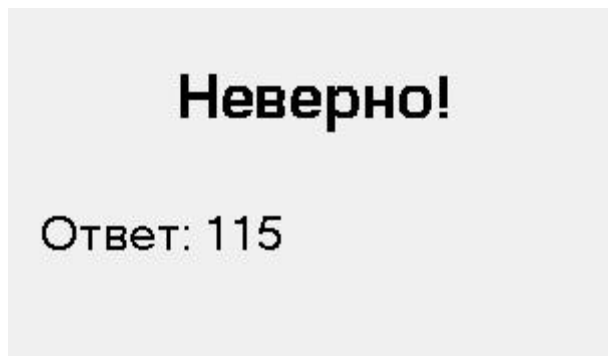


Рисунок 23 – Действия программы при некорректном ответе

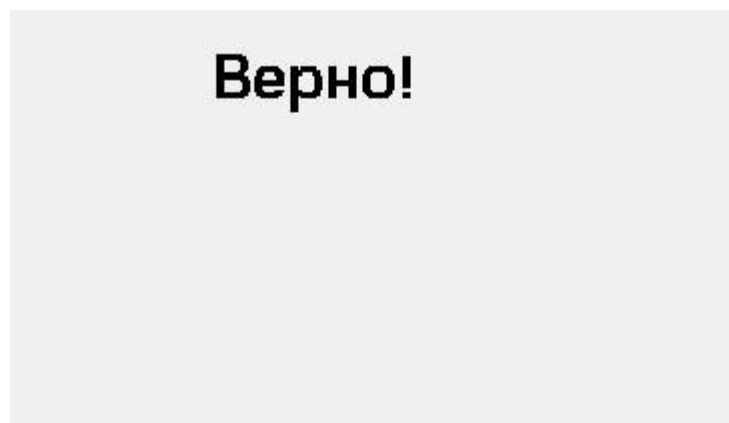


Рисунок 24 – Действия программы при корректном ответе

Даётся 10 заданий, после завершения проверки программа выдаст результат (количество правильных и неправильных ответов), при нажатии кнопки "Ещё раз" можно повторить тест, при этом необходимо заново ввести имя. Кнопка "Выход" закрывает программу, это окно показано на рисунке 25.

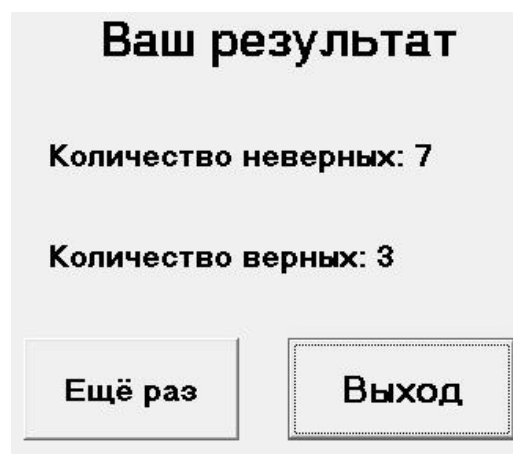


Рисунок 25 – Окно результата

## **ПРИЛОЖЕНИЕ**

Листинг программы Unit1.pas













Листинг программы Unit2.pas

