

Лабораторная работа №4.  
**Разработка интерфейса пользователя.**

Время: 180 мин.

Что нужно освоить:

- 1) свойства и методы редактора RichEdit;
- 2) методы работы с главным и контекстным меню;
- 3) обработка «горячих клавиш»;
- 4) работа с ini-файлом приложения.

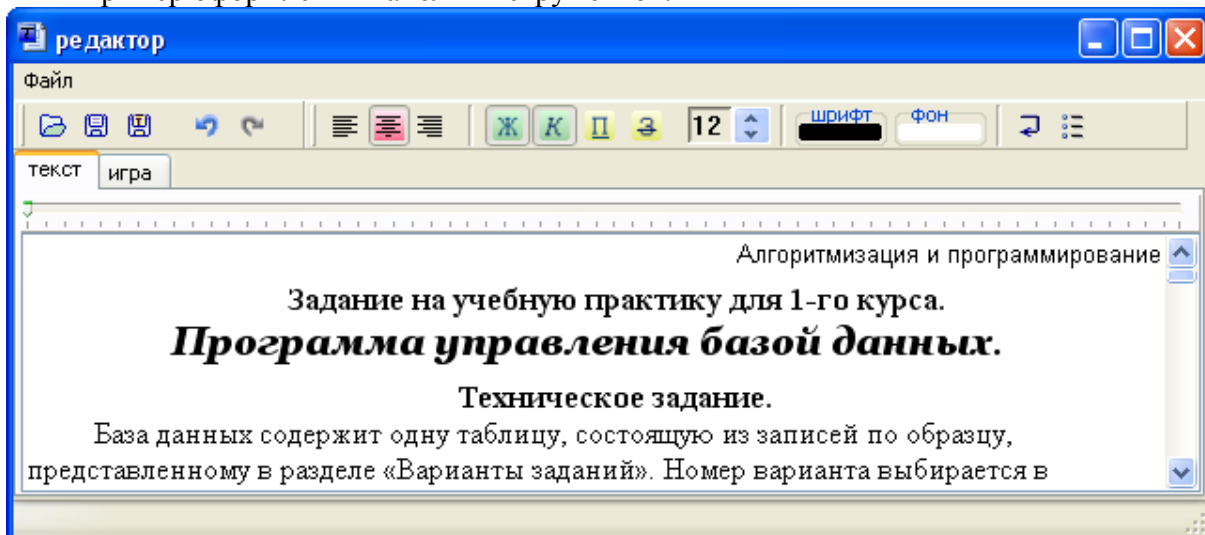
### 1. Техническое задание.

Требуется разработать приложение RTFEdit.exe (текстовый редактор) в среде Delphi – аналог редактора WordPad.

Требования к текстовому редактору RTFEdit.

1. Текстовый редактор должен быть построен на базе компонента RichEdit.
2. В верхней части располагается главное меню приложения, которое дублирует некоторые функции с панели инструментов редактора.
3. В нижней части главной формы должен быть размещен компонент StatusBar. Панели должны отображать текущее время, имя редактируемого файла, а также текущее положение курсора (номер строки и номер позиции в строке).
4. Ниже главного меню располагается панель инструментов, на которую вынесены основные функции редактора.
5. Панель инструментов **обязательно** должна содержать такие функциональные клавиши: открытие и сохранение файла, выравнивание абзаца, размер, цвет и стиль шрифта. Дополнительные требования по наполнению панели инструментов смотрите в разделе 2 «Варианты заданий».

Пример оформления панели инструментов:



6. Должна быть предусмотрена возможность сохранения файла, как в текстовом (txt), так и в расширенном (rtf) формате.

7. Экранные клавиши (управления стилем шрифта и выравнивания абзаца), расположенные на панели инструментов, должны быть построены на базе компонента SpeedButton. Клавиши настройки выравнивания абзаца не могут быть одновременно нажатыми (абзац не может быть сразу выровнен и по левому краю и по центру), а клавиши настройки стиля шрифта наоборот должны иметь такую возможность (шрифт может быть одновременно полужирным и подчеркнутым). При перемещении курсора

по тексту все экранные клавиши панели инструментов должны отображать текущие настройки абзаца и шрифта (пример см. в приложении «Редактор» и на рис. 1 и 2).

Требования к изображениям для клавиш SpeedButton:

- изображения должны быть уникальными, их следует выполнить самостоятельно, плагиат не допускается;
- изображение должны быть интуитивно понятными пользователю;
- изображения, размещенные на экранных клавишах панели инструментов, должны изменяться в зависимости от положения клавиш.

## 2. Варианты заданий.

Кроме обязательной части, описанной в техническом задании, каждый студент обязан выполнить индивидуальную часть. Номер варианта индивидуальной части выбирается в соответствии с последней цифрой порядкового номера студента в журнале группы.

В редакторе должно быть предусмотрено наличие возможности ... :

для номера варианта	задание
1, 6	выбрать цвет фона редактора
2, 7	включать/выключать режим «перенос по словам»
3, 8	установить для выделенного участка режим «список»
4, 9	Undo/Redo
5, 0	установить отступ абзаца слева

Должна быть предусмотрена возможность выбора пользователем «горячих клавиш»:

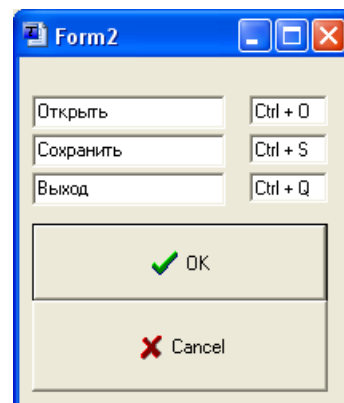
для номера варианта	задание
1, 2	открытие файла выход из программы
3, 4	сохранения файла как txt сохранения файла как rtf
5, 6	выбора цвета шрифта выбора цвета фона
7, 8	выравнивания абзаца по левому краю выравнивания абзаца по правому краю выравнивания абзаца по центру
9, 0	установки полужирного шрифта установки курсивного шрифта установки шрифта с подчеркиванием установки зачеркнутого шрифта

Пример оформления процедуры выбора пользователем «горячих клавиш»:

**Обязательное условие:** назначенные пользователем «горячие клавиши» должны сохраняться в ini-файл приложения.

Таким образом, студенту с вариантом №4 предстоит выполнить индивидуальную часть:

- 1) добавить на панель инструментов клавишу выбора цвета фона редактора,
- 2) добавить возможность вызывать диалог выбора цвета шрифта и фона «горячими клавишами», которые хранятся в ini-файле.



### 3. Защита лабораторной работы.

Студент:

- 1) демонстрирует работоспособность заявленных функций текстового редактора.
- 2) отвечает на вопросы преподавателя по программному коду;
- 3) выполняет дополнительное задание по модификации программного кода.

*Критерии оценивания:*

- «**отлично**» – программа работоспособна, студент свободно ориентируется в тексте программы и в состоянии его модернизировать по требованию преподавателя;
- «**хорошо**» – большая часть функций программы работоспособна, студент ориентируется в тексте программы и в состоянии объяснить суть реализованных в программе алгоритмов;
- «**удовлетворительно**» – некоторые функции программы работоспособны, студент в состоянии объяснить назначение отдельных блоков программы (процедур, функций, обработчиков событий), а также используемых структурных операторов;
- «**неудовлетворительно**» – программа неработоспособна или студент не в состоянии объяснить назначение используемых обработчиков событий или структурных операторов.

*Основания для снижения оценки:*

1. Текст программного кода не имеет структуры, например:

```
procedure TForm1.sbbClick(Sender: TObject);
var n: byte;
begin
n:=(Sender as TSpeedButton).Tag;
with re.SelAttributes do
if msb[n].Down
then Style:=Style+fs[n]
else Style:=Style-fs[n];
end;
```

Следует использовать фиксированные отступы, одинаковые для всего текста программы:

```
procedure TForm1.sbbClick(Sender: TObject);
var n: byte;
begin
    n:=(Sender as TSpeedButton).Tag;
    with re.SelAttributes do
        if msb[n].Down
            then Style:=Style+fs[n]
            else Style:=Style-fs[n];
end;
```

2. Программный код имеет значительное число повторов аналогичных участков алгоритма:

```
procedure TForm1.sbbClick(Sender: TObject);
begin
  with re.SelAttributes do
    if sbb.Down
      then Style:=Style+[fsBold]
      else Style:=Style-[fsBold];
end;

procedure TForm1.sbiClick(Sender: TObject);
begin
  with re.SelAttributes do
    if sbi.Down
      then Style:=Style+[fsItalic]
      else Style:=Style-[fsItalic];
end;

procedure TForm1.sbuClick(Sender: TObject);
begin
  with re.SelAttributes do
    if sbu.Down
      then Style:=Style+[fsUnderline]
      else Style:=Style-[fsUnderline];
end;

procedure TForm1.sbpClick(Sender: TObject);
begin
  with re.SelAttributes do
    if sbp.Down
      then Style:=Style+[fsStrikeOut]
      else Style:=Style-[fsStrikeOut];
end;
```

3. Не используется оператор объединения там, где это позволит сделать код компактнее:

```
procedure TForm1.sbbClick(Sender: TObject);
var n: byte;
begin
  n:=(Sender as TSpeedButton).Tag;
  if msb[n].Down
    then re.SelAttributes.Style:=re.SelAttributes.Style+fs[n]
    else re.SelAttributes.Style:=re.SelAttributes.Style-fs[n];
end;
```

4. При обнаружении плагиата – оценка может быть снижена до «*неудовлетворительно*».