Оглавление

[1.Задание 4](#_Toc533764786)

[2.Назначение программы 5](#_Toc533764787)

[3.Руководство пользователя 6](#_Toc533764788)

[4.Выбор среды разработки 13](#_Toc533764789)

[5.Форматы данных 14](#_Toc533764790)

[5.1 Входные данные 14](#_Toc533764791)

[5.1.1 Обоснование выбора формата файла для хранения исходных данных 14](#_Toc533764792)

[5.1.2 Описание внутренней структуры двоичного файла 15](#_Toc533764793)

[5.2 Выходные данные 16](#_Toc533764794)

[5.2.1 Обоснование выбора формата файла для хранения выходных данных 16](#_Toc533764795)

[6.Стурктура программы 17](#_Toc533764796)

[6.1. Модуль <Main> 17](#_Toc533764797)

[6.1.1 Вспомогательные типы данных 17](#_Toc533764798)

[6.1.2 Класс оконной формы MainForm 17](#_Toc533764799)

[6.1.2.1 Компоненты VCL 18](#_Toc533764800)

[6.1.2.2 Прочие поля класса формы 21](#_Toc533764801)

[6.1.2.3 Методы обработки событий формы и ее компонентов 22](#_Toc533764802)

[6.1.2.4 Прочие методы обработки 27](#_Toc533764803)

[6.1.3 Переменные 27](#_Toc533764804)

[6.2. Модуль <Function> 29](#_Toc533764805)

[6.2.1 Функции 30](#_Toc533764806)

[6.2.2. Переменные 30](#_Toc533764807)

[7.Описание алгоритмов 32](#_Toc533764808)

[7.1. Блок-схема укрупненного алгоритма работы пользовательского интерфейса приложения 32](#_Toc533764809)

[7.2 Блок-схема алгоритма обработки события, возникающего при нажатии пользователем на кнопку «Рассчитать» void \_\_fastcall TMainForm::bZadanie1Click(TObject \*Sender) 33](#_Toc533764810)

[7.3 Блок-схема алгоритма обработки события, возникающего при закрытии …. приложения 35](#_Toc533764811)

[void \_\_fastcall TMainForm::FormCloseQuery(TObject \*Sender, bool &CanClose) 35](#_Toc533764812)

[7.4 Блок-схема алгоритма обработки события, возникающего при заполнении eTime 36](#_Toc533764813)

[void \_\_fastcall TMainForm::eTimeChange(TObject \*Sender) 36](#_Toc533764814)

[7.5 Блок-схема алгоритма обработки события, возникающего при редактировании содержимого ячейки и выхода из нее 37](#_Toc533764815)

[void \_\_fastcall TMainForm::sgResultSetEditText(TObject \*Sender, int ACol, int ARow, const UnicodeString Value) 37](#_Toc533764816)

[7.6 Блок-схема алгоритма для функции 38](#_Toc533764817)

[double norma(double La, double T, double Lz, double Ln, int n) 38](#_Toc533764818)

[7.7 Блок-схема алгоритма обработки события, возникающего при нажатии на пункт главного меню «Файл»-«Сохранить как…» 39](#_Toc533764819)

[void \_\_fastcall TMainForm:: mSaveAsClick(TObject \*Sender) 39](#_Toc533764820)

[7.8 Блок-схема алгоритма обработки события, возникающего при нажатии на пункт главного меню «Файл»-«Загрузить…» 40](#_Toc533764821)

[void \_\_fastcall TMainForm::mOpenClick(TObject \*Sender) 40](#_Toc533764822)

[8.Код программы 41](#_Toc533764823)

# 1.Задание

Расчет годовой нормы расхода запчастей проводится по формуле:

, где

 - амортизационный пробег автомобиля (т. км);

 - ресурс новой детали (т. км);

*n* - число деталей этого типа на автомобиле;

 - амортизационный срок службы автомобиля;

- ресурс запасной части (т. км).

**Задания:**

1. Определить N для следующих исходных данных:

При:

 = 8, 9, 12, 14, 17 лет,

= 400000 т. км,

|  |  |
| --- | --- |
|  |  |
| 30 | 20 |
| 40 | 25 |
| 50 | 30 |
| 60 | 45 |

= 4.

Вывод результата оформить в виде таблицы.

1. Найти амортизационный срок службы автомобиля, при котором норма расхода запчастей принимает заданное значение.
2. Для каждого набора *Lн* и *Lз.ч.* найти срок службы, при котором годовая норма расхода запчастей равна четырем.

**2.Назначение программы**

Данное оконное приложение с графическим пользовательским интерфейсом предназначено для:

* Расчета годовой нормы расхода автомобильных запчастей при варьирующихся исходных данных.
* Расчет амортизационного срока службы автомобиля при заданных начальных условиях.

Преимущество данного приложения заключаются в следующем:

* Исходные данные могут вводиться пользователем вручную непосредственно в программе, либо считываться из внешнего файла.
* Вывод результата вычисления производится в виде таблицы.

# 3.Руководство пользователя

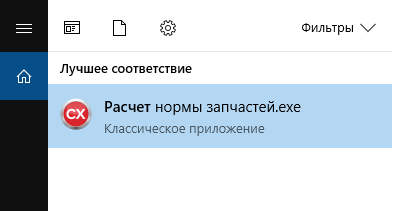
Пользователь должен иметь начальные навыки работы с ОС MS Windows.

Для запуска приложения «Расчет нормы запчастей» необходимо зайти в папку **«KurRab»-«Win32»-«Debug**» и выбрать приложение **«Расчет нормы запчастей»** (рис. 1)

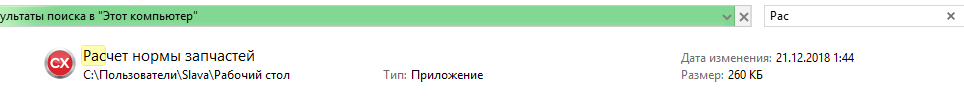
C:\Users\Slava\YandexDisk\Скриншоты\2018-12-23_16-49-58.png

*рис. 1 –приложение «Расчет нормы запчастей»*

Либо воспользоваться поиском, который находится на панели задач (рис. 2) (для OC Window 10, в более ранних версиях ОС строка поиска появится при нажатии на кнопку «Пуск») или поиском, который находится в верхнем левом углу в папке **«Мой компьютер»** (рис. 3)



*рис. 2 – поиск приложения через строку поиска*

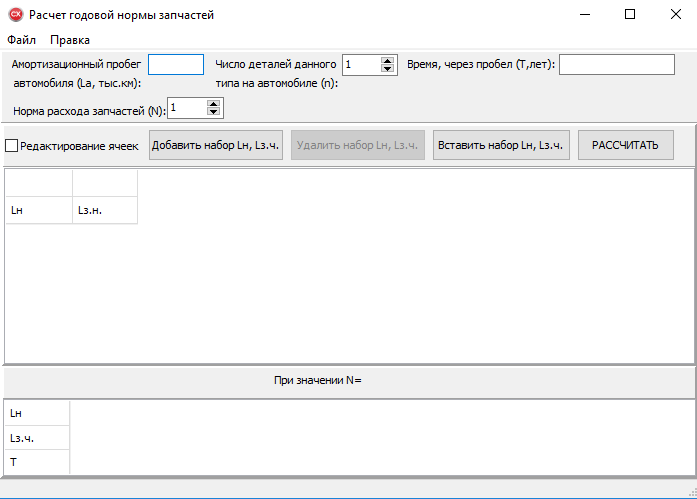


*рис. 3 – поиск приложения через строку поиска*

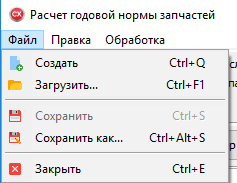
При открытии приложения появится главное окно программы (рис. 4). Перед началом работы с программой «Расчет нормы запчастей» **необходимо** ознакомиться с основными возможностями данной программы:

* **Интерфейс программы**
* **Раздел «Главное меню»**

Представляет из себя набор компонентов для создания проекта, управления загрузкой, сохранением данных, изменения шрифта для удобной работы пользователя. Обратите внимания, что для быстрого доступа к компонентам главного можно пользоваться «горячими клавишами». (рис. 5)



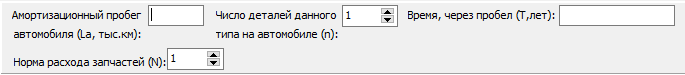
*рис. 4 – главное окно программы*

****

*рис. 5 – главное меню*

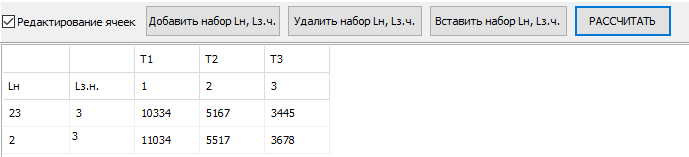
* **Панель ввода данных**

На данной панели представлены формы, предназначенные для ввода исходных данных, либо вывода в них информации из загружаемого файла. (рис. 6)



*рис. 6 – панель ввода данных*

* **Панель управления таблицей и таблица для вывода рассчитанных значений.** (рис. 7)

****

*рис. 7 – таблица и панель управления таблицей*

В первый и второй столбец таблицы вводятся, соответственно, значения Lн и Lз.ч.

С помощью данной панели пользователь может включать/отключать возможность редактирования таблицы, добавлять, удалять, вставлять дополнительные наборы Lн и Lз.ч.

Кнопка «**РАССЧИТАТЬ**» производит расчет годовой нормы запчастей согласно заданию. Вывод значения N производится в таблицу, на пересечении соответствующих значений Lн, Lз.ч. и T. (рис. 8)

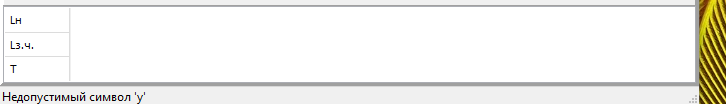
* **Таблица для вывода результатов поиска по заданному значению N.** (рис. 8)



*рис. 8 – таблица для вывода результатов поиска*

* **Строка состояния**

Находится внизу главного окна программы. Здесь будет выводится информация о неверном формате введенных пользователем данных, а также имя файла, в который были сохранены данные. (рис. 9)



*рис. 9 – строка состояния*

* **Начало работы**

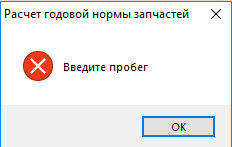
Чтобы произвести расчет годовой нормы расхода запчастей, пользователю необходимо заполнить поля всех форм, располагающихся на панели ввода данных, а также ввести хотя бы один набор значений Lн, Lз.ч.. Либо загрузить уже имеющийся файл с исходными данными при помощи компонента меню **«Файл»-«Загрузить…».**

Для удобства, программа заблокирует ввод недопустимых символов и выведет информацию в строку состояния (рис. 10):

C:\Users\Slava\YandexDisk\Скриншоты\2018-12-20_02-27-07.png

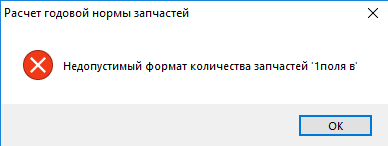
*рис. 10 – вывод ошибки в строку состояния*

Если по каким-либо причинам пользователь не ввел значение, необходимое для расчета, то при нажатии на кнопку «РАССЧИТАТЬ» программа выведет диалоговое окно с указанием незаполненного поля формы (рис. 11):



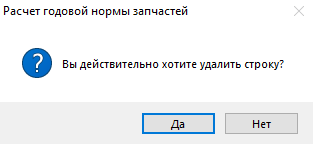
*рис. 11 – предупреждение о незаполненной форме*

Если пользователь ввел значение несоответствующего формата, то при нажатии на кнопку «РАССЧИТАТЬ» программа выведет диалоговое окно, с указанием формы, где находится неверное значение (рис. 12):



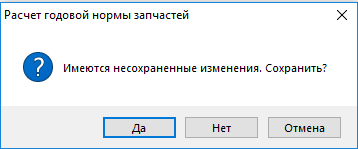
*рис. 12 – предупреждение о неверном формате значения*

Для того, чтобы пользователь случайно не удалил набор Lн, Lз.ч., предусмотрено следующее диалоговое окно (рис. 13):



*рис. 13 – диалоговое окно, появляющееся при удалении строки*

В случае, если пользователь создает новый проект, либо загружает новый файл при имеющихся несохраненных данных, программа выведет следующее диалоговое окно (рис 14.):



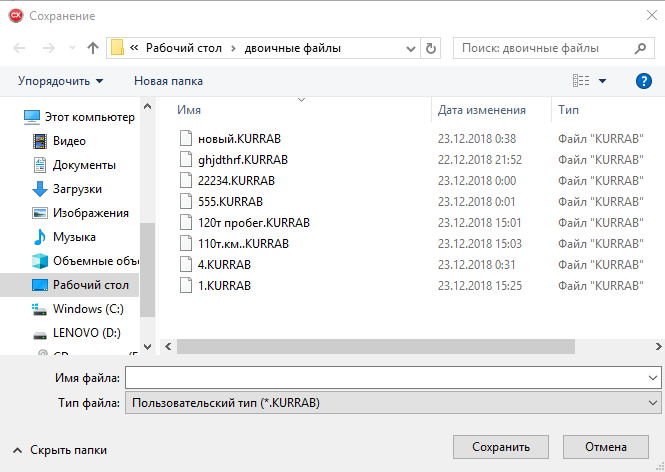
*рис. 14 – диалоговое окно, появляющееся при наличии несохраненных данных*

Если проект связан с файлом, то при нажатии на кнопку **«Да»**, произойдет сохранение несохраненных данных.

Если проект не связан с файлом, то вызовется стандартное диалоговое окно сохранения. (рис 15)

При нажатии на кнопку **«Нет»** будет загружен/создан новый проект.

При нажатии на кнопку **«Отмена»** будет закрыто диалоговое окно и пользователь сможет продолжить работу со старым проектом.



*рис. 15 – стандартное диалоговое окно сохранения*

Другие возможности, предусмотренные в программе:

* быстрое создание нового проекта при выборе пункта меню **«Файл»-«Создать»**
* сохранение исходных данных в указанный файл при выборе пункта меню **«Файл»-«Сохранить как…»**
* изменение шрифта при выборе пункта меню **«Правка»-«Шрифт…»**
* сохранение результатов расчета в текстовый файл при выборе пункта меню **«Обработка»-«Сохранить результаты…»**
* **Завершение работы**

Для завершения работы можно воспользоваться системным выходом, либо выбрать пункт меню «**Файл**»-«**Закрыть**».

В случае, если пользователь не сохранил измененные данные, то программа выведет диалоговое окно. (рис. 14)

При нажатии на кнопку **«Да»** произойдет сохранения несохраненных данных и приложение закроется. Если текущий проект не связан с файлом, то вызовется стандартное диалоговое окно сохранения. (рис 15)

При нажатии на кнопку **«Нет»** приложение закроется, и пользователь потеряет несохраненные изменения.

При нажатии на кнопку **«Отмена»** закроется диалоговое окно, и пользователь сможет продолжить работу.

# 4.Выбор среды разработки

Для решения поставленной задачи выбрана среда разработки Embarcadero C++ Builder. Данная среда включает в себя обширный набор графических компонентов, представленных в библиотеке VCL, такие как:

* TEdit
* TSpinEdit
* TStringGrid

При помощи данных компонентов пользователю удобно вводить данные, а также получать результаты в виде наглядных таблиц. Благодаря свойствам компонента TSpinEdit исключается возможность появления ошибок в случаях, когда перемененная может иметь только целочисленный тип.

# 5.Форматы данных

В данном приложении используются следующий набор данных:

* амортизационный пробег автомобиля (т. км)
* ресурс новой детали (т. км) – количество значений ресурса может быть любым, определяется пользователем, но должно совпадать с количеством значений ресурса запасной части
* число деталей этого типа на автомобиле
* амортизационный срок службы автомобиля – количество сроков службы может быть любым, определяется пользователем.
* ресурс запасной части (т. км)
* годовая норма расхода запчастей

## 5.1 Входные данные

|  |  |  |  |
| --- | --- | --- | --- |
| Обозначение | Название  параметра | Тип значения | Допустимые значения |
| La | амортизационный  пробег  автомобиля | вещественное | ( 0; 4\*…) |
| Lн | ресурс новой детали | вещественное | ( 0; 60 ) |
| n | число деталей этого типа на автомобиле | целочисленное | ( 0; 1000 ) |
| t | амортизационный  срок службы  автомобиля | целочисленное | ( 0; 70 ) |
| Lз.ч. | ресурс запасной части | вещественное | ( 0; 60 ) |

### 5.1.1 Обоснование выбора формата файла для хранения исходных данных

Для хранения данных был выбран двоичный формат по следующим причинам:

1. В данной программе пользователь может обрабатывать большой объем данных, с последующей записью полученных результатов в файл, либо считыванием из него. Двоичные файлы хранят информацию в том виде, в котором она представляется в памяти компьютера во время работы программы. Поэтому при чтении файла или записи данных в файл не выполняется никаких преобразований, что ускоряет процесс чтения и записи.

Таким образом полностью отсутствует трансляция (не нужно преобразовывать символы-разделители).

1. При работе с текстовым файлом необходимо проверять данные на корректность. Благодаря закрытости двоичных файлов, проверки делать не нужно.

### 5.1.2 Описание внутренней структуры двоичного файла

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **La** | **n** | **length\_masT** | **masT[0]** | **…** | **masT[length\_t-1]** | **N** |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **length\_Ln** | **Ln[0]** | **…** | **Ln[length\_Ln-1]** | **length\_Lz4** | **Lz4[0]** | **…** | **Lz4[length\_Lz4-1]** |

|  |  |  |
| --- | --- | --- |
| Имя поля | Количество байт | Описание |
| La | 8ми байтовое вещественно | Значение амортизационного пробега |
| n | 4х байтовое  целое, беззнаковое | Значение количества деталей данного типа на автомобиле |
| length\_masT | 4х байтовое  целое, беззнаковое | Длина массива, содержащего значения амортизационного срока службы |
| masT[0] …  mast[length\_t-1] | length\_masT \* 8 байт  вещественное | Значения амортизационного срока службы |
| N | 4х байтовое  целое, беззнаковое | Значение нормы годового расхода запчастей |
| length\_Ln | 4х байтовое  целое, беззнаковое | Длина массива, содержащего значения ресурса новой детали |
| Ln[0] … Ln[length\_Ln-1] | length\_Ln \* 8 байт  вещественное | Значения ресурса новой детали |
| length\_Lz4 | 4х байтовое  целое, беззнаковое | Длина массива, содержащего значения ресурса запасной части |
| Lz4[0] … Lz4[length\_Lz4-1] | length\_Lz4 \* 8 байт  вещественное | Значения ресурса запасной части |

## 5.2 Выходные данные

|  |  |  |  |
| --- | --- | --- | --- |
| N | Годовая норма расхода запчастей | вещественное | ( 0;) |
| La | амортизационный  пробег  автомобиля | вещественное | ( 0; 4\*…) |
| Lн | ресурс новой детали | вещественное | ( 0; 60 ) |
| n | число деталей этого типа на автомобиле | целочисленное | ( 0; 1000 ) |
| t | амортизационный  срок службы  автомобиля | целочисленное | ( 0; 70 ) |
| Lз.ч. | ресурс запасной части | вещественное | ( 0; 60 ) |

### 5.2.1 Обоснование выбора формата файла для хранения выходных данных

Для хранения выходных данных был выбран текстовый формат по следующим причинам:

1. Формат является открытым, пользователь видит расчеты в удобном форматированном табличном виде.
2. Пользователь может использовать данные для дальнейшей работы (печать, копирование в другой редактор).

**Пример содержимого файла с выходными данными:**

ТАБЛИЦА ЗНАЧЕНИЙ ГОДОВОЙ НОРМЫ РАСХОДА ЗАПЧАСТЕЙ (N)

T1 T2 T3 T4 T5

Lн Lз.ч. 1 2 3 4 5

1 23 323 24 12 8 6 5

**6.Стурктура программы**

## 6.1. Модуль <Main>

Данный модуль состоит из трех файлов:

1. Файл модуля (.cpp) - cодержит исходный код, связанный с формой.
2. Заголовочный файл (.h) - содержит описание класса формы, то есть описания содержащихся на форме компонентов и обработчиков событий.
3. Файл формы (.dfm) - двоичный файл, содержащий сведения об опубликованных (то есть доступных в инспекторе объектов) свойствах компонентов, содержащихся в форме. Двоичный файл формы содержит информацию, используемую для конструирования формы из компонентов, расположенных на ней. При добавлении компонента к форме и заголовочный файл, и двоичный файл формы модифицируются. При редактировании свойств компонента в инспекторе объектов эти изменения сохраняются в двоичном файле формы.

### 6.1.1 Вспомогательные типы данных

|  |  |  |  |
| --- | --- | --- | --- |
| Имя пользовательского  типа | Полное объявление типа данных | Описание структуры | Назначение типа в целом |
| TDA | typedef DynamicArray<double> TDA | DynamicArray<T> - динамический массив  <double> - тип данных элементов массива | Для хранения нескольких значений:  амортизационного срока службы, ресурса новой детали, ресурса запасной части |

### 6.1.2 Класс оконной формы MainForm

На данной оконной форме размещены все компоненты, используемые в данном приложении. Создается форма автоматически при запуске программы.

#### 6.1.2.1 Компоненты VCL

**Обозначения**:

* 1-й уровень вложенности
* 2-й уровень вложенности
* 3-й уровень вложенности
* 4-й уровень вложенности
* 5-й уровень вложенности

|  |  |  |
| --- | --- | --- |
| Имя компонента | Класс | Назначение |
| * OpenDialog | TOpenDialog | позволяет пользователю выбирать файл  для открытия. В данном файле будут хранится следующие данные:  - амортизационный пробег автомобиля  -ресурс новой детали  *-*число деталей этого данного на автомобиле;  - амортизационный срок службы автомобиля  - ресурс запасной части |
| * Panel1 | TPanel | используется для отображения надписей и группирования кнопок. |
| * Panel4 | TPanel | Используется для группирования надписей и компонентов ввода данных |
| * eProbeg | TEdit | Ввод значения амортизационного пробег автомобиля |
| * eTime | TEdit | Ввод значений амортизационного срока службы автомобиля |
| * seDetal | TSpinEdit | Ввод числа деталей данного типа на автомобиле |
| * seN | TSpinEdit | Ввод нормы расхода запчастей  (для задания) |
| * Label1   Label2 | TLabel | Пояснительная надпись для амортизационного пробега автомобиля |
| * Label3 | TLabel | Пояснительная надпись для  ресурса новой детали |
| * Label7   Label8 | TLabel | Пояснительная надпись для числа деталей данного типа на автомобиле |
| * Label9 | TLabel | Пояснительная надпись для нормы расхода запчастей. |
| * Panel2 | TPanel | Используется для группирования надписей и компонентов ввода, вывода данных |
| * Panel3 |  |
| * Panel6 |  |
| * bZada-   nie1 | TButton | При нажатии на данную данный компонент производится расчет и запись в таблицу значений годовой нормы расхода запчастей. |
| * cbEdit | TComboBox | Разрешить/запретить редактирование содержимого ячеек таблицы |
| * eRowM | TButton | При нажатии на данный компонент происходит удаление наборов:  ресурс новой детали/ресурс запасной части |
| * eRowP | TButton | При нажатии на данный компонент происходит добавление строки в таблицу, с целью ввода пользователем очередного набора:  ресурс новой детали/ресурс запасной части |
| * bPaste | TButton | При нажатии на данный компонент происходит вставка строки в таблицу, с целью ввода пользователем очередного набора |
| * Panel7 | TPanel | Используется для размещения таблицы sgResult |
| * sgResult | TStringGrid | Таблица, предназначенная для ввода набора:  ресурс новой детали/ресурс запасной части.  Вывода введенного пользователем  амортизационного срока службы автомобиля  Вывода, найденной в ходе выполнения программы, нормы расхода запчастей |
| * Panel5 | TPanel | Используется для объединения панелей, содержащих компоненты вывода, которые необходимы при выполнении поиска амортизационного срока службы автомобиля по заданной норме расхода запчастей |
| * Panel8 | TPanel | Используется для группирования надписей и компонента вывода данных |
| * eNResult | TEdit | При расчетах дублирует значение компонента seN, для наглядности пользователя |
| * sgSearch | TStringGrid | Служит для вывода найденных значений амортизационного срока службы автомобиля, ресурса новой детали, ресурса запасной части, соответствующих заданному значению нормы расхода запчастей |
| * SaveDialog | TSaveDialog | Служит для сохранения в двоичный файл:  - амортизационного пробега автомобиля  -ресурса новой детали  *-*числа деталей данного типа на автомобиле;  - амортизационного срок службы автомобиля  - ресурса запасной части |
| * sbInfo | TStatusBar | Вывод подсказок на панель, не прерывая работу программы. |
| * MainMenu1 | TMainMenu | Создание удобного интерфейса для работы с меню приложения |
| * mOpen |  | Загрузка нужного файла |
| * mSave |  | Сохранение изменений |
| * mSaveAs |  | Сохранение изменений в выбранный файл |
| * mClose |  | Закрытие приложения |
| * mFont |  | Изменение шрифта |
| * mSaveSG |  | Сохранение результатов расчета |
| * mCreate |  | Создание нового проекта |
| * mZadanie |  | Расчет годовой нормы расхода |
| * mRowP |  | Добавление набора ресурс новой детали/ресурс запасной части |
| * mRowM |  | Удаление набора ресурс новой детали/ресурс запасной части |
| * mPaste |  | Вставка набора ресурс новой детали/ресурс запасной части |

#### 6.1.2.2 Прочие поля класса формы

|  |  |  |  |
| --- | --- | --- | --- |
| Спецификатор видимости | Тип данных | Имя | Назначение |
| private  private | TDA | masT | Массив, предназначенный для хранения значений амортизационного срока службы автомобиля. |
| Lz4 | Массив, предназначенный для хранения значений ресурса запасной части автомобиля. |
| Ln | Массив, предназначенный для хранения значений ресурса новой детали. |
| int | n | Переменная содержит значение  числа деталей данного типа на автомобиле. |
| N | Переменная содержит значение годовой нормы расхода запчастей данного типа. |
| double | La | Переменная содержит значение амортизационного пробега автомобиля. |
| TFont\* | UFont | Переменная хранит информацию о шрифте, который выбрал пользователь |
| UnicodeString | sOld | Хранение имени открытого файла |
| double | LaOld | Хранение значения амортизационного пробега |
| int | nOld | Хранение числа запчастей |
| NOld | Хранение нормы расхода запчастей |
| bool  bool | bLa | Хранение информации о наличии несохраненных значений амортизационного пробега |
| bmasT | Хранение информации о наличии несохраненных значений времени |
| bn | Хранение информации о наличии несохраненных значений числа запчастей |
| bN | Хранение информации о наличии несохраненных значений нормы расхода запчастей |
| sg | Хранение информации о наличии несохраненных значений в ячейках таблицы |
| bSave | Хранение информации о том, произошел ли выбор файла в стандартном диалоге сохранения |
| UnicodeString | CurText | Хранение содержимого ячейки, при входе в режим редактирования |

#### 6.1.2.3 Методы обработки событий формы и ее компонентов

|  |  |  |
| --- | --- | --- |
| Имя метода | Условия вызова | Назначение |
| eProbegChange | Вызывается при вводе пользователем амортизационного пробега  (событие **onChange** компонента **eProbeg**) | Информирование пользователя о недопустимом формате введенного текста и вывода предупреждения в sbInfo |
| eTimeChange | Вызывается при вводе пользователем амортизационного срока службы  (событие **onChange** компонента **eTime**) | Информирование пользователя о недопустимом формате введенного текста и вывода предупреждения в sbInfo.  Определение количества введённых значений амортизационного срока службы автомобиля. |
| FormCreate | Вызывается в **MainForm**  во время начала работы программы.  (событие **onCreate**) | Заполнение фиксированных ячеек в sgResult и sgSearch, установка в sgResult высоты нефиксированной строки в 0.  Закрашивание компонента eNResult. |
| eRowPClick | Вызывается при нажатии на кнопку **«Добавить набор Lн, Lз.ч.»** и выборе пункта главного меню **«Правка»-«Добавить набор»**  (событие **onClick** компонента **eRowP**) | Добавление ячейки в sgResult с целью ввода пользователем очередного значения ресурс новой детали /ресурс запасной части. |
| bPasteClick | Вызывается при нажатии на кнопку **«Вставить набор Lн, Lз.ч.»** и выборе пункта главного меню **«Правка»-«Вставить набор»**  (событие **onClick** компонента **bPaste**) | Вставка набора ресурс новой детали /ресурс запасной части в  выбранную часть таблицы sgResult. |
| eRowMClick | Вызывается при нажатии на кнопку **«Удалить набор Lн, Lз.ч.»** и выборе пункта главного меню **«Правка»-«Удалить набор**  (событие **onClick** компонента **eRowM**) | Удаление ячейки sgResult и одного набора значений  ресурс новой детали /ресурс запасной части. |
| cbEditClick | Вызывается в компоненте **cbEdit** при нажатии на него.  (событие **onClick**) | Запрет/разрешение редактирования ячеек в sgResult. |
| bZadanie1Click | Вызывается при нажатии на кнопку **«РАССЧИТАТЬ»** и выборе пункта главного меню **«Обработка»-«Рассчитать»**  (событие **onClick** компонента **bZadanie1**) | Обработка всех ошибок, допущенных пользователем, с выводом информации о них.  Расчет значения нормы расхода запчастей, вывод данных значений в sgResult.  Поиск амортизационного срока службы автомобиля  По заданной норме расхода запчастей. Вывод найденных значений в sgSearch. |
| mOpenClick | Вызывается при выборе  пункта главного меню **«Файл»-«Загрузить…»**  (событие **onClick**) | Загрузка файла с исходными данными. |
| mSaveClick | Вызывается при выборе  пункта главного меню **«Файл»-«Сохранить»**  (событие **onClick**) | Сохранение изменений в файл. |
| mSaveAsClick | Вызывается при выборе  пункта главного меню **«Файл»-«Сохранить как…»**  (событие **onClick**) | Сохранение изменений в выбранный пользователем файл. |
| mFontClick | Вызывается при выборе  пункта главного меню **«Правка»-«Шрифт»**  (событие **onClick**) | Изменение шрифта. |
| mSaveSGClick | Вызывается при выборе  пункта главного меню **«Обработка»-«Сохранить результаты»**  (событие **onClick**) | Сохранение результатов расчета годовой нормы расхода запчастей в файл. |
| mCreateClick | Вызывается при выборе  пункта главного меню **«Файл»-«Создать»**  (событие **onClick**) | Создание нового проекта. |
| eTimeKeyPress | Вызывается при вводе пользователем амортизационного срока службы с клавиатуры (событие **onKeyPress** компонента **eTime)** | Разрешение вводить только цифры, разделители, с целью предотвращения ввода пользователем некорректных данных.  Разрешение копирования и вставки в eTime->Text. |
| sgResultKeyPress | Вызывается при вводе пользователем данных в таблицу sgResult с клавиатуры  (событие **onKeyPress)** | Разрешение вводить только цифры, разделители, с целью предотвращения ввода пользователем некорректных данных.  Разрешение копирования и вставки в sgResult->Cells[i][j]. |
| sgResultSetEditText | Вызывается при выходе из режима редактирования таблицы sgResult  (событие **onSetEditText)** | Проверка ячеек таблицы sgResult на корректность. В случае некорректного значения – вывод данного значения в sbInfo. |
| eProbegKeyPress | Вызывается при вводе амортизационного пробега пользователем с клавиатуры  (событие **onKeyPress** компонента **eProbeg**) | Разрешение вводить только цифры, разделители, с целью предотвращения ввода пользователем некорректных данных.  Разрешение копирования и вставки в eProbeg->Text. |
| FontDialog1Apply | Вызывается при нажатии на кнопку **«Применить»** пункта главного меню **«Правка»-«Шрифт»-«Применить»**  (событие **onApply**) | Позволяет пользователю подобрать шрифт, с возможностью вернуть прежний. |
| FormCloseQuery | Вызывается при выборе  пункта главного меню **«Файл»-«Закрыть»** и при системном выходе  (событие **onCloseQuery)** | Позволяет отменить выход, а так же сохранить перед выходом данные, измененные в ходе работы программы. |
| sgResultDrawCell | Вызывается автоматически при необходимости перерисовать таблицу **sgResult**  (событие **onDrawCell**) | Изменение цвета фона ячеек. |
| sgResultRowMoved | Вызывается, когда пользователь меняет местами строки таблицы **sgResult**  (событие **onRowMoved**) | Изменение нумерации наборов Lн, Lз.ч. |
| sgResultGetEditText | Вызывается при входе в режим редактирования таблицы **sgResult**  (событие **onGetEditText**) | Отслеживание изменений, которые произошли после входа в режим редактирования. |

#### 6.1.2.4 Прочие методы обработки

* **void \_fastcall TMainForm::CLProject()**

Метод, предназначенный для создания нового проекта во время работы программы. При вызове данного метода происходит очистка таблиц sgResult и sgSearch, а также всех полей для ввода данных.

* **void \_\_fastcall TMainForm::Write(UnicodeString FileN)**

Метод, предназначенный для записи данных в двоичный файл.

|  |  |  |
| --- | --- | --- |
| Имя параметра  (направление передачи данных) | Тип данных параметра | Назначение |
| FileN (вх.) | Unicode  String | Имя файла, в который будет происходить запись несохраненных изменений. |

* **void \_\_fastcall TMainForm::AutoFix()**

Метод, предназначенный для автоматического заполнения фиксированного столбца.

* **void \_\_fastcall TMainForm::AutoSizeSG()**

Метод, предназначенный для автоматического подбора ширины и высоты ячеек при изменении шрифта.

### 6.1.3 Переменные

**Глобальные переменные**

|  |  |  |
| --- | --- | --- |
| Имя переменной | Тип данных | Назначение |
| MainForm | TMainForm\* | Для обращения к форме TMainForm |

**Локальные переменные**

* **TMainForm::eTimeChange(TObject \*Sender)**

|  |  |  |
| --- | --- | --- |
| Имя переменной | Тип данных | Назначение |
| count | int | Данная переменная хранит количество значений амортизационного срока службы автомобиля, находящихся в eTime.  Служит для задания числа столбцов sgResult. |
| s | string | Строка, хранящая все данные, введенные в eTime. |
| newS | string | В данную переменную считываются все значения, которые хранятся в s, с целью выявления некорректных значений |

* **TMainForm::bZadanie1Click(TObject \*Sender)**

|  |  |  |
| --- | --- | --- |
| Имя переменной | Тип данных | Назначение |
| newS | string | В данную переменную считываются все значения, которые хранятся в s, с целью выявления некорректных значений. |
| s | string | Строка, хранящая все данные, введенные в eTime. |
| i | int | Используется в цикле for для:  - проверки наличия пустых ячеек в sgResult.  - корректности данных в ячейке.  -заполнения массивов Ln, Lz4  -расчета нормы расхода запчастей и вывода их в sgResult  -вывода, по заданию, при заданных условия, значений срока службы, ресурса новой детали и реурса запасной части в sgSearch. |
| k | int | Счетчик столбцов в sgSearch.  Используется в цикле for для:  -вывода, по заданию, при заданных условия, значений срока службы, ресурса новой детали и ресурса запасной части в sgSearch. |
| j | int | Используется в цикле for для:  -расчета нормы расхода запчастей и вывода их в sgResult. |

* **TMainForm::mSaveClick(TObject \*Sender)**

|  |  |  |
| --- | --- | --- |
| Имя переменной | Тип данных | Назначение |
| Q | int | Хранит длину массива masT, Ln, Lz4.  Используется в цикле for для:  -запись размера массивов masT, Ln, Lz4, значений masT, Ln, Lz4 в бинарный файл. |

* **TMainForm::mOpenClick(TObject \*Sender)**

|  |  |  |
| --- | --- | --- |
| Имя переменной | Тип данных | Назначение |
| Q | int | В данную переменную из бинарного файла считывается длина массива masT, Ln, Lz4.  Используется в цикле for для:  -считывания из бинарного файла значений срока службы, ресурса новой детали и реcурса запасной части |
| s | string | Хранит все значения массива masT, с целью последующего вывода значений в eTime. |
| i | int | Используется в цикле for для:  -считывания из бинарного файла значений срока службы, ресурса новой детали и реcурса запасной части. |

* **TMainForm::bPasteClick(TObject \*Sender)**

|  |  |  |
| --- | --- | --- |
| Имя переменной | Тип данных | Назначение |
| i | int | Используется в цикле for для:  - Задания числа строк  - Изменения содержимого строк. |

* **TMainForm::eRowMClick(TObject \*Sender)**

|  |  |  |
| --- | --- | --- |
| Имя переменной | Тип данных | Назначение |
| i | int | Используется в цикле for для:  - Задания числа строк  - Изменения содержимого строк. |

## 6.2. Модуль <Function>

Данный модуль состоит из двух файлов:

1. Файл модуля (.cpp) – cодержит описание пользовательских функций.
2. Заголовочный файл (.h) – содержит прототипы функций.

### 6.2.1 Функции

* **double norma(double La, double T, double Lz, double Ln, int n)**

Данная функция служит для расчета значения годовой нормы расхода запчастей N.

Тип возвращаемого значения – double. Функция возвращает округленное до единиц в большую сторону значение годовой нормы расхода запчастей N.

|  |  |  |
| --- | --- | --- |
| Имя параметра  (направление передачи данных) | Тип данных параметра | Назначение |
| La (вх.) | double | Передает в функцию значение амортизационного пробега автомобиля. |
| T (вх.) | double | Передает в функцию амортизационный срок службы автомобиля. |
| Lz (вх | double | Передает в функцию ресурс запасной части. |
| Ln (вх.) | double | Передает в функцию ресурс новой детали. |
| n (вх.) | int | Передает в функцию число деталей данного типа на автомобиле. |

### 6.2.2. Переменные

**Локальные переменные**

* **double norma(double La, double T, double Lz, double Ln, int n)**

|  |  |  |
| --- | --- | --- |
| Имя переменной | Тип данных | Назначение |
| La | double | Хранит значение амортизационного пробега автомобиля. |
| T | double | Хранит амортизационный срок службы автомобиля. |
| Lz | double | Хранит ресурс запасной части. |
| Ln | double | Хранит ресурс новой детали. |
| n | int | Хранит число деталей данного типа на автомобиле. |
| N | double | Хранит значение годовой нормы расхода запчастей (значение, которое возвращает функция). |

# 7.Описание алгоритмов

## 7.1. Блок-схема укрупненного алгоритма работы пользовательского интерфейса приложения



## 7.2 Блок-схема алгоритма обработки события, возникающего при нажатии пользователем на кнопку «Рассчитать» void \_\_fastcall TMainForm::bZadanie1Click(TObject \*Sender)



Создание объекта ss потокового

класса stringstream.

Считывание строки s в объект ss.

конец



## 7.3 Блок-схема алгоритма обработки события, возникающего при закрытии приложения c несохраненными данными

## void \_\_fastcall TMainForm::FormCloseQuery(TObject \*Sender, bool &CanClose)



## 7.4 Блок-схема алгоритма обработки события, возникающего при заполнении eTime

## void \_\_fastcall TMainForm::eTimeChange(TObject \*Sender)



Создание объекта ss потокового класса stringstream.

Считывание строки s в объект ss.

## 

## 7.5 Блок-схема алгоритма обработки события, возникающего при редактировании содержимого ячейки и выхода из нее

## void \_\_fastcall TMainForm::sgResultSetEditText(TObject \*Sender, int ACol, int ARow, const UnicodeString Value)



## 7.6 Блок-схема алгоритма для функции

## double norma(double La, double T, double Lz, double Ln, int n)



## 7.7 Блок-схема алгоритма обработки события, возникающего при нажатии на пункт главного меню «Файл»-«Сохранить как…»

## void \_\_fastcall TMainForm:: mSaveAsClick(TObject \*Sender)



## 7.8 Блок-схема алгоритма обработки события, возникающего при нажатии на пункт главного меню «Файл»-«Загрузить…»

## void \_\_fastcall TMainForm::mOpenClick(TObject \*Sender)



Создание объекта in потокового класса ifstream

# 8.Код программы

**Файл <Function.cpp>**

#pragma hdrstop

#include "Function.h"

#include "math.h"

double norma(double La, double T, double Lz, double Ln, int n)

{

double N=(La-Ln)\*n\*100/Lz/T;

return ceil(N);

}

//---------------------------------------------------------------------------

#pragma package(smart\_init)

**Файл <KurRab.cpp>**

#include <vcl.h>

#pragma hdrstop

#include <tchar.h>

//---------------------------------------------------------------------------

USEFORM("Main.cpp", MainForm);

//---------------------------------------------------------------------------

int WINAPI \_tWinMain(HINSTANCE, HINSTANCE, LPTSTR, int)

{

try

{

Application->Initialize();

Application->MainFormOnTaskBar = true;

Application->Title = "Расчет годовой нормы запчастей";

Application->CreateForm(\_\_classid(TMainForm), &MainForm);

Application->Run();

}

catch (Exception &exception)

{

Application->ShowException(&exception);

}

catch (...)

{

try

{

throw Exception("");

}

catch (Exception &exception)

{

Application->ShowException(&exception);

}

}

return 0;

}

**Файл <Main.cpp>**

#include <vcl.h>

#pragma hdrstop

#include "Main.h"

#include "Function.h"

#include <iostream>

#include <sstream>

#include <string>

#include <fstream>

#include <iomanip>

using namespace std;

//---------------------------------------------------------------------------

#pragma package(smart\_init)

#pragma resource "\*.dfm"

TMainForm \*MainForm;

//---------------------------------------------------------------------------

\_\_fastcall TMainForm::TMainForm(TComponent\* Owner)

: TForm(Owner)

{

}

//---------------------------------------------------------------------------

void \_\_fastcall TMainForm::AutoSizeSG()

{

for (int j = 0; j < sgResult->ColCount; j++)

{

int TempColSize = 0;

//int TempRowSize=0;

//int MaxRowSize=sgResult->DefaultRowHeight;

int MaxColSize =sgResult->DefaultColWidth;

for (int i = 0; i < sgResult->RowCount; i++)

{

sgResult->Canvas->Font->Assign(sgResult->Font);

TempColSize = sgResult->Canvas->TextWidth (sgResult->Cells[j][i]);

if (TempColSize > MaxColSize)

{

MaxColSize = TempColSize;

}

}

if (MaxColSize == sgResult->DefaultColWidth)

{

sgResult->ColWidths[j] = sgResult->DefaultColWidth;

}

else

{

sgResult->ColWidths[j] = MaxColSize + 15;

}

}

for (int j = 0; j < sgResult->RowCount; j++)

{

int TempRowSize=0;

int MaxRowSize=sgResult->DefaultRowHeight;

for (int i = 0; i < sgResult->ColCount; i++)

{

sgResult->Canvas->Font->Assign(sgResult->Font);

TempRowSize = sgResult->Canvas->TextHeight (sgResult->Cells[i][j]);

if (TempRowSize > MaxRowSize)

{

MaxRowSize = TempRowSize;

}

}

if (MaxRowSize == sgResult->DefaultRowHeight)

{

sgResult->RowHeights[j] = sgResult->DefaultRowHeight;

}

else

{

sgResult->RowHeights[j] = MaxRowSize + 10;

}

}

}

void \_fastcall TMainForm::AutoFix()

{

for(int i=2; i<sgResult->RowCount;i++)

{

sgResult->Cells[0][i]=IntToStr(i-1);

}

}

void \_fastcall TMainForm::CLProject()

{

eProbeg->Text=L"";

seDetal->Text=L"1";

eTime->Text=L"";

seN->Text=L"1";

eNResult->Text="";

for(int i=2;i<sgResult->RowCount;i++)

sgResult->Rows[i]->Clear();

sgResult->RowCount=3;

sgResult->RowHeights[2]=0;

for(int i=1;i<sgSearch->ColCount;i++)

sgSearch->Cols[i]->Clear();

sgSearch->ColCount=2;

sgSearch->ColWidths[1]=0;

Ln.Length=0;

Lz4.Length=0;

StringFile=L"";

bLa=false;

bmasT=false;

bn=false;

bN=false;

sg=false;

mSave->Enabled=false;

}

void \_\_fastcall TMainForm::Write(UnicodeString FileN)

{

ofstream out;

out.open(AnsiString(FileN).c\_str(), ios::binary);

n=StrToInt(seDetal->Text);

N=StrToInt(seN->Text);

TDA LnOld;

TDA Lz4Old;

out.write((char\*)&La,sizeof(La));

out.write((char\*)&n, sizeof(n));

int Q;

Q=masT.Length;

out.write((char\*)&Q,sizeof(Q));

for(int i=0;i<Q;i++)

{

out.write((char\*)&masT[i],sizeof(masT[i]));

}

out.write((char\*)&N, sizeof(N));

Q=0;

for(int i=2; i<sgResult->RowCount; i++)

{

if(sgResult->Cells[1][i]!="")

Q++;

}

LnOld.Length=Q;

out.write((char\*)&Q,sizeof(Q));

Q=0;

for(int i=2; i<sgResult->RowCount; i++)

{

if(sgResult->Cells[1][i]!="")

{

LnOld[Q]=StrToFloat(sgResult->Cells[1][i]);

Q++;

}

}

for(int i=0;i<Q;i++)

{

out.write((char\*)&LnOld[i],sizeof(LnOld[i]));

}

Q=0;

for(int i=2; i<sgResult->RowCount; i++)

{

if(sgResult->Cells[2][i]!="")

Q++;

}

Lz4Old.Length=Q;

out.write((char\*)&Q,sizeof(Q));

Q=0;

for(int i=2; i<sgResult->RowCount; i++)

{

if(sgResult->Cells[2][i]!="")

{

Lz4Old[Q]=StrToFloat(sgResult->Cells[2][i]);

Q++;

}

}

for(int i=0;i<Q;i++)

{

out.write((char\*)&Lz4Old[i],sizeof(Lz4Old[i]));

}

out.close();

}

//------------------------------------------------------------

void \_\_fastcall TMainForm::eProbegChange(TObject \*Sender)

{

try

{

La=StrToFloat(eProbeg->Text);

if(La==LaOld)

{

bLa=false;

}

else

{

bLa=true;

}

}

catch(...)

{

if(eProbeg->Text!="")

sbInfo->SimpleText=L"Недопустимый формат пробега '"+UnicodeString(eProbeg->Text)+"'";

La=false;

}

if(eProbeg->Text==L""&&StringFile=="")

bLa=false;

if(bLa==true||bN==true||bn==true||bmasT==true||sg==true)

{

mSave->Enabled=true;

}

else

{

mSave->Enabled=false;

}

}

//---------------------------------------------------------------------------

void \_\_fastcall TMainForm::eTimeChange(TObject \*Sender)

{

int count=0;

string newS;

string s=AnsiString(eTime->Text).c\_str();

stringstream ss;

ss<<s;

while(ss>>newS)

{

count++;

masT.Length=count;

try

{

masT[count-1]=StrToFloat(newS.c\_str());

sgResult->ColCount=3+count;

sgResult->Cells[2+count][0]="T"+IntToStr(count);

sgResult->Cells[2+count][1]=FloatToStr(masT[count-1]);

}

catch(...)

{

sbInfo->SimpleText=L"Неверный формат времени '" + UnicodeString(newS.c\_str()) + "'";

}

}

if(eTime->Text==""&&StringFile=="")

bmasT=false;

if(s==AnsiString(sOld).c\_str())

bmasT=false;

else

bmasT=true;

if(bLa==true||bN==true||bn==true||bmasT==true||sg==true)

mSave->Enabled=true;

else

mSave->Enabled=false;

if(eTime->Text=="")

sgResult->ColCount=3;

}

//---------------------------------------------------------------------------

void \_\_fastcall TMainForm::FormCreate(TObject \*Sender)

{

sgResult->Cells[1][1]=L"Lн";

sgResult->Cells[2][1]=L"Lз.ч.";

sgSearch->Cells[0][0]=L"Lн";

sgSearch->Cells[0][1]=L"Lз.ч.";

sgSearch->Cells[0][2]=L"T";

sgResult->Cells[0][1]="№";

sgResult->ColWidths[0]=22;

eNResult->Brush->Color=clBtnFace;

sgResult->RowHeights[2]=0;

sgSearch->ColWidths[1]=0;

StringFile="";

bLa=false;

bmasT=false;

bn=false;

bN=false;

sg=false;

bSave=false;

}

//---------------------------------------------------------------------------

void \_\_fastcall TMainForm::eRowPClick(TObject \*Sender)

{

eRowM->Enabled=true;

mRowM->Enabled=true;

if(sgResult->RowHeights[2]==0)

{

sgResult->RowHeights[2]=sgResult->DefaultRowHeight;

sgResult->Row=2;

AutoFix();

}

else

sgResult->RowCount=sgResult->RowCount+1;

sgResult->Row=sgResult->RowCount-1;

sgResult->Rows[sgResult->Row]->Clear();

sgResult->SetFocus();

Ln.Length=sgResult->RowCount-2;

Lz4.Length=sgResult->RowCount-2;

AutoFix();

}

//---------------------------------------------------------------------------

void \_\_fastcall TMainForm::eRowMClick(TObject \*Sender)

{

if(Application->MessageBoxW((UnicodeString(L"Вы действительно хотите удалить набор №")+sgResult->Cells[0][sgResult->Row]+L" ?").w\_str(),Application->Title.w\_str(),MB\_YESNO|MB\_ICONQUESTION)!=ID\_YES)

return;

if(sgResult->RowCount==3)

{

Ln.Length=0;

Lz4.Length=0;

sgResult->RowHeights[2]=0;

eRowM->Enabled=false;

mRowM->Enabled=false;

sg=true;

if(bLa==true||bN==true||bn==true||bmasT==true||sg==true)

{

mSave->Enabled=true;

}

else

{

mSave->Enabled=false;

}

return;

}

int n=sgResult->Selection.Bottom-sgResult->Selection.Top+1;

if(sgResult->Selection.Bottom<sgResult->RowCount-1)

for(int i=sgResult->Selection.Bottom+1;i<sgResult->RowCount;i++)

sgResult->Rows[i-n]=sgResult->Rows[i];

if(sgResult->RowCount-n>=2)

sgResult->RowCount=sgResult->RowCount-n;

if(sgResult->RowCount==2)

{

eRowM->Enabled=false;

mRowM->Enabled=false;

}

sg=true;

if(bLa==true||bN==true||bn==true||bmasT==true||sg==true)

{

mSave->Enabled=true;

}

else

{

mSave->Enabled=false;

}

AutoFix();

}

//---------------------------------------------------------------------------

//---------------------------------------------------------------------------

void \_\_fastcall TMainForm::cbEditClick(TObject \*Sender)

{

if(cbEdit->Checked==true)

{

sgResult->Options=sgResult->Options<<goEditing;

}

else

{

sgResult->Options=sgResult->Options>>goEditing;

}

}

//---------------------------------------------------------------------------

void \_\_fastcall TMainForm::sgResultKeyPress(TObject \*Sender, System::WideChar &Key)

{

if(Key==VK\_ESCAPE&&sgResult->EditorMode)

{

sgResult->Cells[sgResult->Col][sgResult->Row]=CurText;

return;

}

if(!(Key>=L'0'&&Key<='9'||Key==FormatSettings.DecimalSeparator||Key==VK\_BACK||Key==24||Key==3||Key==22||Key==26||Key==13||Key==VK\_TAB))

{

sbInfo->SimpleText=L"Недопустимый символ '"+UnicodeString(Key)+"'";

Key=0;

}

else

sbInfo->SimpleText=L"";

if(sgResult->Row>1&&sgResult->Col>2)

if(!(Key==3||Key==13))

{

sbInfo->SimpleText=L"Недопустимый символ '"+UnicodeString(Key)+"'";

Key=0;

}

}

//---------------------------------------------------------------------------

void \_\_fastcall TMainForm::sgResultSetEditText(TObject \*Sender, int ACol, int ARow, const UnicodeString Value)

{

Ln.Length=sgResult->RowCount-2;

Lz4.Length=sgResult->RowCount-2;

if(sgResult->Cells[ACol][ARow]==L"")

{

return;

}

if(ACol==1)

{

try

{

StrToFloat((sgResult->Cells[ACol][ARow]));

}

catch(...)

{

sbInfo->SimpleText=L"Неверный формат Ln в наборе №"+IntToStr(ARow-1);

}

}

if(ACol==2)

{

try

{

StrToFloat(sgResult->Cells[ACol][ARow]); //.c\_str()

}

catch(...)

{

sbInfo->SimpleText=L"Неверный формат Lз.ч. в наборе №"+IntToStr(ARow-1);

}

}

for( int i=0;i<sgResult->RowCount-2;i++)

{

if(FloatToStr(Ln[i])!=sgResult->Cells[1][i+2]||FloatToStr(Lz4[i])!=sgResult->Cells[2][i+2])

{

sg=true;

break;

}

else

sg=false;

}

if(bLa==true||bN==true||bn==true||bmasT==true||sg==true)

mSave->Enabled=true;

else

mSave->Enabled=false;

}

//---------------------------------------------------------------------------

void \_\_fastcall TMainForm::bZadanie1Click(TObject \*Sender)

{

for(int i=1; i<sgSearch->ColCount;i++)

{

sgSearch->Cols[i]->Clear();

}

sgSearch->ColCount=2;

sgSearch->ColWidths[1]=0;

if(eProbeg->Text=="")

{

eProbeg->SetFocus();

Application->MessageBoxW(L"Введите пробег",Application->Title.w\_str(),MB\_OK|MB\_ICONERROR);

return;

}

try

{

La=StrToFloat(eProbeg->Text);

}

catch(...)

{

eProbeg->SetFocus();

sbInfo->SimpleText=L"Недопустимый формат пробега '"+UnicodeString(eProbeg->Text)+"'";

Application->MessageBoxW(sbInfo->SimpleText.w\_str(),Application->Title.w\_str(),MB\_OK|MB\_ICONERROR);

return;

}

//-----------------------------

if(seDetal->Text=="")

{

seDetal->SetFocus();

sbInfo->SimpleText=L"Заполните форму для количества запчастей";

Application->MessageBoxW(sbInfo->SimpleText.w\_str(),Application->Title.w\_str(),MB\_OK|MB\_ICONERROR);

return;

}

try

{

n=StrToInt(seDetal->Text);

}

catch(...)

{

seDetal->SetFocus();

sbInfo->SimpleText=L"Недопустимый формат количества запчастей '"+UnicodeString(seDetal->Text)+"'";

Application->MessageBoxW(sbInfo->SimpleText.w\_str(),Application->Title.w\_str(),MB\_OK|MB\_ICONERROR);

return;

}

//-----------------------------

if(eTime->Text=="")

{

eTime->SetFocus();

sbInfo->SimpleText=L"Заполните форму для времени";

Application->MessageBoxW(sbInfo->SimpleText.w\_str(),Application->Title.w\_str(),MB\_OK|MB\_ICONERROR);

return;

}

string newS;

string s=AnsiString(eTime->Text).c\_str();

stringstream ss;

ss<<s;

while(ss>>newS)

{

try

{

StrToFloat(newS.c\_str());

}

catch(...)

{

sbInfo->SimpleText=L"Неверный формат времени '" + UnicodeString(newS.c\_str()) + "'";

eTime->SetFocus();

Application->MessageBoxW(sbInfo->SimpleText.w\_str(),Application->Title.w\_str(),MB\_OK|MB\_ICONERROR);

return;

}

}

//-----------------------------

if(seN->Text=="")

{

seN->SetFocus();

sbInfo->SimpleText=L"Введите норму запчастей";

Application->MessageBoxW(sbInfo->SimpleText.w\_str(),Application->Title.w\_str(),MB\_OK|MB\_ICONERROR);

return;

}

try

{

N=StrToInt(seN->Text);

}

catch(...)

{

seN->SetFocus();

sbInfo->SimpleText=L"Недопустимый формат нормы запчастей '"+UnicodeString(seN->Text)+"'";

Application->MessageBoxW(sbInfo->SimpleText.w\_str(),Application->Title.w\_str(),MB\_OK|MB\_ICONERROR);

return;

}

//--------------------------------------

if(sgResult->RowHeights[2]==0)

{

Application->MessageBoxW(UnicodeString(L"Добавьте набор Lн, Lз.ч.").w\_str(),Application->Title.w\_str(),MB\_OK|MB\_ICONERROR);

eRowP->SetFocus();

return;

}

for(int i=2; i<sgResult->RowCount;i++)

{

if(sgResult->Cells[1][i]=="")

{

sgResult->SetFocus();

sbInfo->SimpleText=L"Заполните Lн в наборе №"+IntToStr(i-1);

Application->MessageBoxW(sbInfo->SimpleText.w\_str(),Application->Title.w\_str(),MB\_OK|MB\_ICONERROR);

return;

}

if(sgResult->Cells[2][i]=="")

{

sgResult->SetFocus();

sbInfo->SimpleText=L"Заполните Lз.ч. в наборе №"+IntToStr(i-1);

Application->MessageBoxW(sbInfo->SimpleText.w\_str(),Application->Title.w\_str(),MB\_OK|MB\_ICONERROR);

return;

}

}

//----------------------------

for(int i=2;i<sgResult->RowCount;i++)

{

if(sgResult->Cells[1][i]!="")

{

try

{

StrToFloat(sgResult->Cells[1][i]);

}

catch(...)

{

sbInfo->SimpleText=L"Неверный формат Lн в наборе №"+IntToStr(i-1);

Application->MessageBoxW(sbInfo->SimpleText.w\_str(),Application->Title.w\_str(),MB\_OK|MB\_ICONERROR);

return;

}

}

}

for(int i=2;i<sgResult->RowCount;i++)

{

if(sgResult->Cells[2][i]!="")

{

try

{

StrToFloat(sgResult->Cells[2][i]);

}

catch(...)

{

sbInfo->SimpleText=L"Неверный формат Lз.ч. в наборе №"+IntToStr(i-1);

Application->MessageBoxW(sbInfo->SimpleText.w\_str(),Application->Title.w\_str(),MB\_OK|MB\_ICONERROR);

return;

}

}

}

Ln.Length=sgResult->RowCount-2;

Lz4.Length=sgResult->RowCount-2;

for(int i=2;i<sgResult->RowCount;i++)

{

Ln[i-2]=StrToFloat(sgResult->Cells[1][i]);

Lz4[i-2]=StrToFloat(sgResult->Cells[2][i]);

}

//----------------------------

int k=1; // счетчик столбцов для новой таблицы

eNResult->Text=seN->Text;

for(int i=0;i<sgResult->RowCount-2;i++)

for(int j=0;j<masT.Length;j++)

{

sgResult->Cells[j+3][i+2]=FloatToStr(norma(La,masT[j],Lz4[i],Ln[i],StrToInt(seDetal->Text)));

if(StrToFloat(sgResult->Cells[j+3][i+2])==StrToFloat(seN->Text))

{

if(sgSearch->ColWidths[1]==0)

{

sgSearch->ColWidths[1]=sgSearch->DefaultColWidth;

sgSearch->Cells[k][2]=masT[j];

sgSearch->Cells[k][1]=Lz4[i];

sgSearch->Cells[k][0]=Ln[i];

k++;

}

else

{

sgSearch->ColCount=sgSearch->ColCount+1;

sgSearch->Cells[k][2]=masT[j];

sgSearch->Cells[k][1]=Lz4[i];

sgSearch->Cells[k][0]=Ln[i];

k++;

}

}

}

}

//---------------------------------------------------------------------------

void \_\_fastcall TMainForm::mSaveAsClick(TObject \*Sender)

{

try

{

StrToFloat(eProbeg->Text);

}

catch(...)

{

if(eProbeg->Text=="")

{

eProbeg->SetFocus();

sbInfo->SimpleText=L"Введите пробег";

Application->MessageBoxW(sbInfo->SimpleText.w\_str(),Application->Title.w\_str(),MB\_OK|MB\_ICONERROR);

return;

}

eProbeg->SetFocus();

sbInfo->SimpleText=L"Недопустимый формат пробега '"+UnicodeString(eProbeg->Text)+"'";

Application->MessageBoxW(sbInfo->SimpleText.w\_str(),Application->Title.w\_str(),MB\_OK|MB\_ICONERROR);

return;

}

try

{

StrToInt(seDetal->Text);

}

catch(...)

{

seDetal->SetFocus();

sbInfo->SimpleText=L"Недопустимый формат количества запчастей '"+UnicodeString(seDetal->Text)+"'";

Application->MessageBoxW(sbInfo->SimpleText.w\_str(),Application->Title.w\_str(),MB\_OK|MB\_ICONERROR);

return;

}

try

{

StrToInt(seN->Text);

}

catch(...)

{

seN->SetFocus();

sbInfo->SimpleText=L"Недопустимый формат нормы запчастей '"+UnicodeString(seN->Text)+"'";

Application->MessageBoxW(sbInfo->SimpleText.w\_str(),Application->Title.w\_str(),MB\_OK|MB\_ICONERROR);

return;

}

if(sgResult->RowHeights[2]==0)

{

sbInfo->SimpleText=L"Необходимо ввести набор Lн, Lз.ч.";

eRowP->SetFocus();

Application->MessageBoxW(sbInfo->SimpleText.w\_str(),Application->Title.w\_str(),MB\_OK|MB\_ICONERROR);

return;

}

int z=0;

for(int i=2;i<sgResult->RowCount;i++)

{

if(sgResult->Cells[1][i]==""&&sgResult->Cells[2][i]=="")

z++;

}

if(z==sgResult->RowCount-2)

{

sbInfo->SimpleText=L"Необходимо ввести набор Lн, Lз.ч.";

Application->MessageBoxW(sbInfo->SimpleText.w\_str(),Application->Title.w\_str(),MB\_OK|MB\_ICONERROR);

return;

}

for(int i=2;i<sgResult->RowCount;i++)

{

if(sgResult->Cells[1][i]!="")

{

try

{

StrToFloat(sgResult->Cells[1][i]);

}

catch(...)

{

sbInfo->SimpleText=L"Неверный формат Lн в наборе №"+IntToStr(i-1);

Application->MessageBoxW(sbInfo->SimpleText.w\_str(),Application->Title.w\_str(),MB\_OK|MB\_ICONERROR);

return;

}

}

}

for(int i=2;i<sgResult->RowCount;i++)

{

if(sgResult->Cells[2][i]!="")

{

try

{

StrToFloat(sgResult->Cells[2][i]);

}

catch(...)

{

sbInfo->SimpleText=L"Неверный формат Lз.ч. в наборе №"+IntToStr(i-1);

Application->MessageBoxW(sbInfo->SimpleText.w\_str(),Application->Title.w\_str(),MB\_OK|MB\_ICONERROR);

return;

}

}

}

SaveDialog->FileName="";

if(!SaveDialog->Execute())

{

bSave=false;

return;

}

Write(SaveDialog->FileName);

sOld=AnsiString(eTime->Text).c\_str();

LaOld=La;

nOld=n;

NOld=N;

sbInfo->SimpleText=StringFile;

mSave->Enabled=false;

bLa=false;

bmasT=false;

bn=false;

bN=false;

sg=false;

bSave=true;

AutoFix();

}

//---------------------------------------------------------------------------

void \_\_fastcall TMainForm::mOpenClick(TObject \*Sender)

{

if(mSave->Enabled==true)

switch(Application->MessageBoxW((UnicodeString("Вы хотите сохранить измененения в файл ")+L"'"+OpenDialog->FileName+L"' ?").w\_str(),Application->Title.w\_str(),MB\_YESNOCANCEL|MB\_ICONQUESTION))

{

case ID\_YES:

mSaveClick(Sender);

break;

case ID\_NO:

OpenDialog->FileName="";

break;

case ID\_CANCEL:

return;

}

OpenDialog->FileName="";

if(!OpenDialog->Execute())

return;

eRowM->Enabled=true;

mRowM->Enabled=true;

for(int i=1; i<sgSearch->ColCount;i++)

{

sgSearch->Cols[i]->Clear();

}

sgSearch->ColCount=2;

sgSearch->ColWidths[1]=0;

ifstream in;

in.open(AnsiString(OpenDialog->FileName).c\_str(), ios::binary);

StringFile=UnicodeString(OpenDialog->FileName);

for(int i=2; i<sgResult->RowCount;i++)

sgResult->Rows[i]->Clear();

in.read((char\*)&La,sizeof(La));

eProbeg->Text=FloatToStr(La);

//--------------------------------------------------

in.read((char\*)&n, sizeof(n));

seDetal->Text=IntToStr(n);

//--------------------------------------------------

masT.Length=0;

int Q;

in.read((char\*)&Q,sizeof(Q));

masT.Length=Q;

for(int i=0;i<Q;i++)

{

in.read((char\*)&masT[i],sizeof(masT[i]));

}

stringstream ss;

for(int i=0; i<Q;i++)

{

ss<<AnsiString(FloatToStr(masT[i]));

ss<<AnsiString(" ");

}

string s;

s=ss.str();

eTime->Text=s.c\_str();

//---------------------------------------------------

in.read((char\*)&N, sizeof(N));

seN->Text=IntToStr(N);

in.read((char\*)&Q, sizeof(Q));

Ln.Length=Q;

for(int i=0;i<Q;i++)

{

in.read((char\*)&Ln[i],sizeof(Ln[i]));

}

in.read((char\*)&Q, sizeof(Q));

Lz4.Length=Q;

for(int i=0;i<Q;i++)

{

in.read((char\*)&Lz4[i],sizeof(Lz4[i]));

}

if(Ln.Length==0||Lz4.Length==0)

{

sgResult->RowCount=3;

sgResult->RowHeights[2]=0;

}

else

{

if(Ln.Length>=Lz4.Length)

sgResult->RowCount=Ln.Length+2;

else

sgResult->RowCount=Lz4.Length+2;

sgResult->RowHeights[2]=sgResult->DefaultRowHeight;

}

for(int i=2;i<sgResult->RowCount;i++)

{

sgResult->Cells[1][i]=FloatToStr(Ln[i-2]);

sgResult->Cells[2][i]=FloatToStr(Lz4[i-2]);

}

in.close();

masTOld.Length=masT.Length;

for(int i=0; i<masT.Length; i++)

masTOld[i]=masT[i];

sOld=AnsiString(eTime->Text).c\_str();

LaOld=La;

nOld=n;

NOld=N;

mSave->Enabled=false;

bLa=false;

bmasT=false;

bn=false;

bN=false;

sg=false;

AutoFix();

MainForm->Caption =Application->Title+" - "+ExtractFileName(OpenDialog->FileName); //Возвращает название файла

//без пути

}

//---------------------------------------------------------------------------

void \_\_fastcall TMainForm::eTimeKeyPress(TObject \*Sender, System::WideChar &Key)

{

if(!(Key>=L'0'&&Key<='9'||Key==FormatSettings.DecimalSeparator|| Key==VK\_SPACE||Key==VK\_BACK||Key==24||Key==3||Key==22||Key==26))

{

sbInfo->SimpleText=L"Недопустимый символ "+UnicodeString(Key);

Key=0;

}

else

sbInfo->SimpleText=L"";

}

//---------------------------------------------------------------------------

//---------------------------------------------------------------------------

void \_\_fastcall TMainForm::eProbegKeyPress(TObject \*Sender, System::WideChar &Key)

{

if(!(Key>=L'0'&&Key<='9'||Key==FormatSettings.DecimalSeparator||Key==VK\_BACK||Key==24||Key==3||Key==22||Key==26))

{

sbInfo->SimpleText=L"Недопустимый символ "+UnicodeString(Key);

Key=0;

}

else

sbInfo->SimpleText=L"";

}

//---------------------------------------------------------------------------

void \_\_fastcall TMainForm::bPasteClick(TObject \*Sender)

{

eRowM->Enabled=true;

mRowM->Enabled=true;

if(sgResult->RowHeights[2]==0)

{

sgResult->RowHeights[2]=sgResult->DefaultRowHeight;

}

else

{

for(int i=sgResult->RowCount-1;i>=sgResult->Row;i--)

sgResult->Rows[i+1]=sgResult->Rows[i];

sgResult->RowCount=sgResult->RowCount+1;

}

sgResult->Rows[sgResult->Row]->Clear();

AutoFix();

}

//---------------------------------------------------------------------------

void \_\_fastcall TMainForm::FormCloseQuery(TObject \*Sender, bool &CanClose)

{

if(mSave->Enabled==true)

switch(Application->MessageBoxW((UnicodeString("Вы хотите сохранить измененения в файл ")+L"'"+OpenDialog->FileName+L"' ?").w\_str(),Application->Title.w\_str(),MB\_YESNOCANCEL|MB\_ICONQUESTION))

{

case ID\_YES:

mSaveClick(Sender);

if(!bSave) //Сохранили ли мы данные при выборе диалогового окна.

{

CanClose=false;

return;

}

else

{

return;

}

case ID\_NO:

return;

case ID\_CANCEL:

CanClose=false;

}

}

//---------------------------------------------------------------------------

void \_\_fastcall TMainForm::mFontClick(TObject \*Sender)

{

UFont=new TFont();

try

{

UFont->Assign(sgResult->Font);

if (FontDialog1->Execute())

{

sgResult->Font=FontDialog1->Font;

AutoSizeSG();

}

else

{

sgResult->Font=UFont;

AutoSizeSG();

}

}

\_\_finally

{

delete UFont;

}

}

//---------------------------------------------------------------------------

void \_\_fastcall TMainForm::FontDialog1Apply(TObject \*Sender, HWND Wnd)

{

sgResult->Font=FontDialog1->Font;

AutoSizeSG();

}

//---------------------------------------------------------------------------

void \_\_fastcall TMainForm::seDetalChange(TObject \*Sender)

{

try

{

n=StrToInt(seDetal->Text);

if(n==nOld)

{

bn=false;

}

else

{

bn=true;

}

}

catch (...)

{

sbInfo->SimpleText=L"Недопустимый формат количества деталей '"+UnicodeString(seDetal->Text)+"'";

bn=false;

}

if(seDetal->Text=="1"&&StringFile=="")

bn=false;

if(bLa==true||bN==true||bn==true||bmasT==true||sg==true)

{

mSave->Enabled=true;

}

else

{

mSave->Enabled=false;

}

}

//---------------------------------------------------------------------------

void \_\_fastcall TMainForm::seNChange(TObject \*Sender)

{

try

{

N=StrToInt(seN->Text);

if(N==NOld)

{

bN=false;

}

else

{

bN=true;

}

}

catch (...)

{

sbInfo->SimpleText=L"Недопустимый формат нормы расхода запчастей '"+UnicodeString(seN->Text)+"'";

bN=false;

}

if(seN->Text=="1"&&StringFile=="")

bN=false;

if(bLa==true||bN==true||bn==true||bmasT==true||sg==true)

mSave->Enabled=true;

else

mSave->Enabled=false;

}

//---------------------------------------------------------------------------

void \_\_fastcall TMainForm::mSaveClick(TObject \*Sender)

{

try

{

StrToFloat(eProbeg->Text);

}

catch(...)

{

eProbeg->SetFocus();

sbInfo->SimpleText=L"Недопустимый формат пробега '"+UnicodeString(eProbeg->Text)+"'";

Application->MessageBoxW(sbInfo->SimpleText.w\_str(),Application->Title.w\_str(),MB\_OK|MB\_ICONERROR);

return;

}

try

{

StrToInt(seDetal->Text);

}

catch(...)

{

seDetal->SetFocus();

sbInfo->SimpleText=L"Недопустимый формат количества запчастей '"+UnicodeString(seDetal->Text)+"'";

Application->MessageBoxW(sbInfo->SimpleText.w\_str(),Application->Title.w\_str(),MB\_OK|MB\_ICONERROR);

return;

}

try

{

StrToInt(seN->Text);

}

catch(...)

{

seN->SetFocus();

sbInfo->SimpleText=L"Недопустимый формат нормы запчастей '"+UnicodeString(seN->Text)+"'";

Application->MessageBoxW(sbInfo->SimpleText.w\_str(),Application->Title.w\_str(),MB\_OK|MB\_ICONERROR);

return;

}

if(sgResult->RowHeights[2]==0)

{

sbInfo->SimpleText=L"Необходимо ввести набор Lн, Lз.ч.";

eRowP->SetFocus();

Application->MessageBoxW(sbInfo->SimpleText.w\_str(),Application->Title.w\_str(),MB\_OK|MB\_ICONERROR);

return;

}

int z=0;

for(int i=2;i<sgResult->RowCount;i++)

{

if(sgResult->Cells[1][i]==""&&sgResult->Cells[2][i]=="")

z++;

}

if(z==sgResult->RowCount-2)

{

sbInfo->SimpleText=L"Необходимо ввести набор Lн, Lз.ч.";

Application->MessageBoxW(sbInfo->SimpleText.w\_str(),Application->Title.w\_str(),MB\_OK|MB\_ICONERROR);

return;

}

for(int i=2;i<sgResult->RowCount;i++)

{

if(sgResult->Cells[1][i]!="")

{

try

{

StrToFloat(sgResult->Cells[1][i]);

}

catch(...)

{

sbInfo->SimpleText=L"Неверный формат Lн в наборе №"+IntToStr(i-1);

Application->MessageBoxW(sbInfo->SimpleText.w\_str(),Application->Title.w\_str(),MB\_OK|MB\_ICONERROR);

return;

}

}

}

for(int i=2;i<sgResult->RowCount;i++)

{

if(sgResult->Cells[2][i]!="")

{

try

{

StrToFloat(sgResult->Cells[2][i]);

}

catch(...)

{

sbInfo->SimpleText=L"Неверный формат Lз.ч. в наборе №"+IntToStr(i-1);

Application->MessageBoxW(sbInfo->SimpleText.w\_str(),Application->Title.w\_str(),MB\_OK|MB\_ICONERROR);

return;

}

}

}

ofstream out;

if(StringFile=="")

{

mSaveAsClick(Sender);

}

else

{

Write(StringFile);

sOld=AnsiString(eTime->Text).c\_str();

LaOld=La;

nOld=n;

NOld=N;

sbInfo->SimpleText=StringFile;

bLa=false;

bmasT=false;

bn=false;

bN=false;

sg=false;

mSave->Enabled=false;

AutoFix();

}

}

//---------------------------------------------------------------------------

void \_\_fastcall TMainForm::mExitClick(TObject \*Sender)

{

Close();

}

//---------------------------------------------------------------------------

void \_\_fastcall TMainForm::mSaveSGClick(TObject \*Sender)

{

SaveDialog->FileName="";

SaveDialog->Filter = "Текстовый файл (\*.txt)|\*.txt|Все файлы (\*.\*)|\*.\*";

if(!SaveDialog->Execute())

{

SaveDialog->Filter="Пользовательский тип (\*.KURRAB)|\*.KURRAB| Текстовый файл (\*.txt)|\*.txt|Все файлы (\*.\*)|\*.\*";

return;

}

ofstream out;

out.open(AnsiString(SaveDialog->FileName).c\_str());

if(!out.is\_open())

{

sbInfo->SimpleText="Файл не открыт";

return;

}

out<<" "<<"ТАБЛИЦА ЗНАЧЕНИЙ ГОДОВОЙ НОРМЫ РАСХОДА ЗАПЧАСТЕЙ (N)"<<endl<<endl;

for(int i=0; i<sgResult->RowCount;i++)

{

for(int j=0; j<sgResult->ColCount; j++)

out<<setw(7)<<AnsiString(sgResult->Cells[j][i]).c\_str();

out<<'\n';

}

out<<endl<<endl<<endl<<endl;

out<<" "<<"ПРИ НОРМЕ РАСХОДА ЗАПЧАСТЕЙ N="<<AnsiString(seN->Text).c\_str()<<endl<<endl;

for(int i=0; i<sgSearch->RowCount;i++)

{

for(int j=0;j<sgSearch->ColCount;j++)

out<<setw(7)<<AnsiString(sgSearch->Cells[j][i]).c\_str();

out<<'\n';

}

SaveDialog->Filter="Пользовательский тип (\*.KURRAB)|\*.KURRAB| Текстовый файл (\*.txt)|\*.txt|Все файлы (\*.\*)|\*.\*";

out.close();

return;

}

//---------------------------------------------------------------------------

void \_\_fastcall TMainForm::mCreateClick(TObject \*Sender)

{

if(mSave->Enabled==true)

switch(Application->MessageBoxW((UnicodeString("Вы хотите сохранить измененения в файл ")+OpenDialog->FileName+L"?").w\_str(),Application->Title.w\_str(),MB\_YESNOCANCEL|MB\_ICONQUESTION))

{

case ID\_YES:

mSaveClick(Sender);

CLProject();

return;

case ID\_NO:

CLProject();

return;

case ID\_CANCEL:

return;

}

else

{

CLProject();

MainForm->Caption =Application->Title;

return;

}

}

//---------------------------------------------------------------------------

void \_\_fastcall TMainForm::sgResultDrawCell(TObject \*Sender, int ACol, int ARow, TRect &Rect,

TGridDrawState State)

{

if(ARow>1&&ACol>2)

{

TRect r=Rect;

r.Left-=4;

sgResult->Canvas->Brush->Color=clSilver;

sgResult->Canvas->FillRect(r);

sgResult->Canvas->TextRect(r,r.Left+4,r.Top+2,sgResult->Cells[ACol][ARow]);

}

}

//---------------------------------------------------------------------------

void \_\_fastcall TMainForm::sgResultRowMoved(TObject \*Sender, int FromIndex, int ToIndex)

{

AutoFix();

}

//---------------------------------------------------------------------------

void \_\_fastcall TMainForm::sgResultGetEditText(TObject \*Sender, int ACol, int ARow,

UnicodeString &Value)

{

CurText=Value;

for( int i=0;i<sgResult->RowCount-2;i++)

{

if(FloatToStr(Ln[i])!=sgResult->Cells[1][i+2]||FloatToStr(Lz4[i])!=sgResult->Cells[2][i+2])

{

sg=true;

break;

}

else

sg=false;

}

if(bLa==true||bN==true||bn==true||bmasT==true||sg==true)

mSave->Enabled=true;

else

mSave->Enabled=false;

}

//---------------------------------------------------------------------------

**Файл <Main.h>**

//---------------------------------------------------------------------------

#ifndef MainH

#define MainH

//---------------------------------------------------------------------------

#include <System.Classes.hpp>

#include <Vcl.Controls.hpp>

#include <Vcl.StdCtrls.hpp>

#include <Vcl.Forms.hpp>

#include <Vcl.Menus.hpp>

#include <Vcl.Grids.hpp>

#include <Vcl.Mask.hpp>

#include <Vcl.ExtCtrls.hpp>

#include <Vcl.Samples.Spin.hpp>

#include <Vcl.ComCtrls.hpp>

#include <Vcl.Dialogs.hpp>

#include <iostream>

#include <sstream>

#include <string>

#include <fstream>

typedef DynamicArray<double> TDA;

//---------------------------------------------------------------------------

class TMainForm : public TForm

{

\_\_published: // IDE-managed Components

TMainMenu \*MainMenu1;

TMenuItem \*N1;

TMenuItem \*mOpen;

TMenuItem \*mSave;

TMenuItem \*mExit;

TMenuItem \*mFont;

TPanel \*Panel1;

TLabel \*Label1;

TLabel \*Label7;

TLabel \*Label8;

TEdit \*eProbeg;

TSpinEdit \*seDetal;

TPanel \*Panel4;

TEdit \*eTime;

TLabel \*Label3;

TStatusBar \*sbInfo;

TLabel \*Label2;

TCheckBox \*cbEdit;

TStringGrid \*sgResult;

TButton \*eRowP;

TButton \*eRowM;

TButton \*bZadanie1;

TPanel \*Panel2;

TPanel \*Panel3;

TPanel \*Panel5;

TPanel \*Panel6;

TPanel \*Panel7;

TEdit \*eNResult;

TStringGrid \*sgSearch;

TLabel \*Label9;

TSpinEdit \*seN;

TPanel \*Panel8;

TSaveDialog \*SaveDialog;

TMenuItem \*mSaveAs;

TOpenDialog \*OpenDialog;

TButton \*bPaste;

TMenuItem \*N7;

TMenuItem \*N8;

TFontDialog \*FontDialog1;

TPanel \*Panel9;

TMenuItem \*mCreate;

TMenuItem \*N2;

TMenuItem \*mZadanie;

TMenuItem \*mSaveSG;

TMenuItem \*N3;

TMenuItem \*mRowP;

TMenuItem \*mRowM;

TMenuItem \*mPaste;

void \_\_fastcall eProbegChange(TObject \*Sender);

void \_\_fastcall eTimeChange(TObject \*Sender);

void \_\_fastcall FormCreate(TObject \*Sender);

void \_\_fastcall eRowPClick(TObject \*Sender);

void \_\_fastcall eRowMClick(TObject \*Sender);

void \_\_fastcall cbEditClick(TObject \*Sender);

void \_\_fastcall sgResultKeyPress(TObject \*Sender, System::WideChar &Key);

void \_\_fastcall sgResultSetEditText(TObject \*Sender, int ACol, int ARow, const UnicodeString Value);

void \_\_fastcall bZadanie1Click(TObject \*Sender);

void \_\_fastcall mSaveAsClick(TObject \*Sender);

void \_\_fastcall mOpenClick(TObject \*Sender);

void \_\_fastcall eTimeKeyPress(TObject \*Sender, System::WideChar &Key);

void \_\_fastcall eProbegKeyPress(TObject \*Sender, System::WideChar &Key);

void \_\_fastcall bPasteClick(TObject \*Sender);

void \_\_fastcall FormCloseQuery(TObject \*Sender, bool &CanClose);

void \_\_fastcall mFontClick(TObject \*Sender);

void \_\_fastcall FontDialog1Apply(TObject \*Sender, HWND Wnd);

void \_\_fastcall seDetalChange(TObject \*Sender);

void \_\_fastcall seNChange(TObject \*Sender);

void \_\_fastcall mSaveClick(TObject \*Sender);

void \_\_fastcall mExitClick(TObject \*Sender);

void \_\_fastcall mSaveSGClick(TObject \*Sender);

void \_\_fastcall mCreateClick(TObject \*Sender);

void \_\_fastcall sgResultDrawCell(TObject \*Sender, int ACol, int ARow, TRect &Rect,

TGridDrawState State);

void \_\_fastcall sgResultRowMoved(TObject \*Sender, int FromIndex, int ToIndex);

void \_\_fastcall sgResultGetEditText(TObject \*Sender, int ACol, int ARow, UnicodeString &Value);

private:// User declarations

TDA masT;

double La; // амортизационный пробег

TDA Lz4;

TDA Ln;

int n;

int N;

TFont \*UFont;

UnicodeString StringFile;

TDA masTOld;

UnicodeString sOld;

double LaOld;

int nOld;

int NOld;

bool bLa;

bool bmasT;

bool bn;

bool bN;

bool sg;

bool bSave;

UnicodeString CurText;

public: // User declarations

\_\_fastcall TMainForm(TComponent\* Owner);

void \_\_fastcall CLProject();

void \_\_fastcall Write(UnicodeString FileN);

void \_\_fastcall AutoFix();

void \_\_fastcall AutoSizeSG();

};

//---------------------------------------------------------------------------

extern PACKAGE TMainForm \*MainForm;

//---------------------------------------------------------------------------

#endif