

Программные системы и вычислительные методы

Правильная ссылка на статью:

Савельева Ю.О., Ильин А.В., Лукьянов А.А., Милихина К.М. Разработка прикладного приложения для автоматизации проектирования в КОМПАС-3D: таблица контроля качества основного металла // Программные системы и вычислительные методы. 2025. № 1. DOI: 10.7256/2454-0714.2025.1.73073 EDN: VBHBRP URL: https://nbpublish.com/library_read_article.php?id=73073

Разработка прикладного приложения для автоматизации проектирования в КОМПАС-3D: таблица контроля качества основного металла

Савельева Юлия Олеговна

ORCID: 0009-0004-8971-4110

кандидат технических наук

доцент, Филиал ФГБОУ ВО "СамГТУ" в г. Сызрани; Самарский государственный технический университет

446001, Россия, Самарская область, г. Сызрань, ул. Советская, 45, каб. 404

✉ Savelieva_yu_ol@mail.ru



Ильин Артем Владимирович

студент, Филиал ФГБОУ ВО "СамГТУ" в г. Сызрани; Самарский государственный технический университет

446001, Россия, Самарская область, г. Сызрань, ул. Советская, 45

✉ Savelieva_yu_ol@mail.ru



Лукьянов Александр Александрович

студент, Филиал ФГБОУ ВО "СамГТУ" в г. Сызрани; Самарский государственный технический университет

446001, Россия, Самарская область, г. Сызрань, ул. Советская, 45

✉ Savelieva_yu_ol@mail.ru



Милихина Ксения Михайловна

студент, Филиал ФГБОУ ВО "СамГТУ" в г. Сызрани; Самарский государственный технический университет

446001, Россия, Самарская область, г. Сызрань, ул. Советская, 45

✉ Savelieva_yu_ol@mail.ru



[Статья из рубрики "Системный анализ, поиск, анализ и фильтрация информации"](#)

DOI:

10.7256/2454-0714.2025.1.73073

EDN:

VBHBRP

Дата направления статьи в редакцию:

18-01-2025

Аннотация: Сложность составления таблиц контроля качества основного металла ТБ1 для проектирования изделий атомной энергетики определяется необходимостью учета совокупности множества требований различных источников (которые могут пересекаться, дополнять друг друга): стандарты, например, отраслевой стандарт (ОСТ), строительные нормы и правила (СНиП), технические условия, требования заказчика, а также внутренние нормы на их заполнение и другие нормативные документы. Некорректно составленные документы, переданные в уполномоченную организацию контроля качества (организацию уполномоченную на проведение работ по оценке соответствия в форме приемки (испытаний) продукции, предназначенной для применения в элементах объектов использования атомной энергии Российской Федерации), означают для разработчика финансовые и временные потери, трудности при дальнейшей сдаче изделия. В данной работе приводится порядок разработки прикладного пользовательского приложения (библиотеки), позволяющего автоматически сформировать и построить таблицу контроля качества основного металла ТБ1 на документе чертежа в КОМПАС-3D. Разработан код прикладного приложения (библиотеки) для САПР КОМПАС-3D на языке Delphi, с помощью которого реализуется автоматизированное построение ТБ1, необходимой при проектировании оборудования для атомной энергетики. Одной из важнейших характеристик современной инженерной системы моделирования является возможность внедрения автоматизации, например, использование подсистем для расширения возможностей программного пакета, в качестве которых могут выступать прикладные приложения (пользовательские программные библиотеки). В данной работе сформулированы основные требования к функционалу прикладного пользовательского приложения (библиотеки) для КОМПАС-3D: детали осуществления запуска формирования ТБ1; возможности выбора контрольных операций для материалов сборки, операций с шифрами, также добавление новых операций, с последующим их сохранением в памяти программы; возможность внесения оперативных изменений. Код приложения позволяет считать данные с файла сборки, открытого в КОМПАС-3D, создать новый документ чертежа, построить таблицу по требованиям ТБ1, занести туда данные, взятые со сборочного документа.

Ключевые слова:

таблица контроля качества, основной металл, Т Б 1 , КОМПАС-3D, прикладное приложение, пользовательская библиотека, программная библиотека, автоматизация проектирования, Delphi, инструменты создания библиотек

Введение. Эффективность программного пакета КОМПАС-3D для трехмерного моделирования, создания чертежей, спецификаций определяется использованием не только базовых инструментальных средств для реализации целевой функции, но и возможностью расширения функционала программного пакета с помощью прикладных

пользовательских приложений, пользовательских программных библиотек [\[1-5\]](#). В САПР КОМПАС-3D уже предусмотрено такое средство автоматизации, как встроенные прикладные библиотеки – подключаемые программные модули, число которых в разных версиях отличается, позволяющие сократить время проектирования за счет простоты и высокой скорости создания и использования стандартных элементов на основе базовых функций [\[6-8\]](#).

Несмотря на значительное количество библиотек (конструкторской библиотеки, содержащей более двухсот параметрических изображений типовых элементов: различных крепежных элементов, подшипников, профилей, элементов соединений трубопроводов и др.; библиотеки редукторов, электродвигателей, крепежей, канавок, пружин, деталей штампов, пресс-форм, материалов и сортаментов и прочих [\[6\]](#)), не всегда их достаточно для реализации целевой функции. Например, какой-то конкретной библиотеки нет в свободном доступе или на предприятиях разрабатываются изделия по собственным нормативным документам. Готовые модули для разработки приложений в КОМПАС содержат в себе описание интерфейсов в соответствии с синтаксисом определенного языка программирования.

На предприятиях при проектировании и изготовлении оборудования атомной энергетики составляются таблицы контроля качества основного металла (ТБ1), на базе которых разрабатываются программы качества. Таблицам контроля качества, устанавливающим необходимость выполнения конкретных контрольных операций, присваиваются определенные шифры: «ТБ1» - таблицы контроля качества основного металла; «ТБ2» - таблицы контроля качества сварных соединений и наплавов; «ТБ3» - таблицы контроля качества производственных сварных соединений на производственных контрольных сварных соединениях [\[9\]](#).

Проекты с использованием ТБ1 проходят приёмку в уполномоченной организации по программам контроля качества [\[10-11\]](#). Программа контроля качества изделий атомной энергетики, состоящая из текстовой части и таблиц контроля качества, определяет принципиальные положения и требования, предъявляемые к качеству проектируемых сборочных единиц, изделий, оборудования и трубопроводов, установок в процессе изготовления [\[9\]](#).

Составление ТБ1 – сложный процесс, т.к. конструктору-исполнителю при проектировании необходимо опираться на различные стандарты, например, на отраслевые стандарты (ОСТ), государственные стандарты, строительные нормы и правила (СНиП), технические условия и другие нормативные документы, действуя в соответствии с требованиями главного проектировщика – заказчика. Некорректно составленные документы, переданные в уполномоченную организацию контроля качества, влекут за собой финансовые и временные потери для разработчиков, а также трудности при дальнейшей сдаче изделия.

Объект исследования: таблица контроля качества основного металла (ТБ1).

Предмет исследования: порядок разработки прикладного пользовательского приложения.

Постановка задачи. Необходимо разработать прикладное пользовательское приложение для Компас-3D позволяющее автоматически сформировать ТБ1.

Основные требования к функционалу пользовательского приложения:

1. Запуск происходит со сборки в КОМПАС-3D через панель команд или через кнопки «Приложения» - «ТБ1».
2. После запуска в окне программы отображается таблица с полученными со сборки деталями (также детали всех подсборок), их количеством, обозначением, материалом и соответствующим стандартом.
3. В программе должен быть функционал позволяющий конструктору менять все графы таблицы, выбирать те позиции, которые построятся в таблице на чертеже, добавлять новые позиции, удалять добавленные, зачеркивать текст и возможность разделить таблицу на несколько листов.
4. В приложении должен быть предусмотрен выбор контрольных операций для материалов сборки, операций с шифрами, также добавление новых операций, с последующим их сохранением в памяти программы.
5. Возможность выбора символов для операций по клику на соответствующую ячейку.
6. По нажатию на соответствующую кнопку автоматически должен создаваться документ чертежа, с построенной таблицей со всеми изменениями, внесенными конструктором в интерфейс программы.
7. Возможность добавить функционал считывания данных с чертежа для внесения изменений через интерфейс программы.

В данной работе построение ТБ1 осуществляется согласно ОСТ 108.004.10-86^[9] и внутренним нормам на их заполнение у заказчика. Программный код приложения написан на языке Delphi.

Порядок разработки прикладного пользовательского приложения. Создается «заготовка» для приложения (библиотеки) КОМПАС:

- 1) В Delphi выполняется команда File - New - Other, в появившемся окне New Items со всеми возможными шаблонами Windows-приложений и файлами Delphi (исполняемый exe-файл, консольное приложение, форма или компонент Delphi и пр.) на вкладке «Delphi Projects» выбирается пункт Dynamic-Link Library.
- 2) Подключаются файлы, которые содержат в себе описание всех интерфейсов API, версии 7. Для этого выполняется команда Tools - Options – Library, в графе Library path указывается путь к каталогу include, который находится в папке КОМПАС.
- 3) Выполняется команда Projects - options - Application. В поле Target file extension вводится расширение.
- 4) В проект необходимо добавляется новый модуль и 64-битная платформа, выполняется сохранение проекта и компиляция.
- 5) Чтобы «заготовка» могла взаимодействовать с КОМПАС, в ней должны присутствовать функции: LibraryEntry (точка входа в библиотеку), LibraryId (возвращает идентификатор библиотеки), GetNewKompasAPI (получение доступа к новому API; не обязательна, но удобна тем, что не приходится каждый раз писать проверку на API). Эти функции обязательно должны быть экспортными, т.е. экспортируемыми, чтобы система КОМПАС могла их видеть и вызывать; по этой причине они выносятся в раздел exports.

Необходимо осуществить считывание данных со сборки. Все основные процедуры и

функции взаимодействия приложения с КОМПАС-3D находятся в модуле UstatementMaterials.

Как и во всех библиотеках под КОМПАС-3D используется процедура LIBRARYENTRY, код показан на Рис. 1.

```

procedure LIBRARYENTRY( command: WORD ); pascal;
var
  handleFAssembly, handleFDrawing, HandleFActiveList: HWND;
begin
  //доработанная процедура LIBRARYENTRY
  handleFAssembly:= FindWindow(PWideChar('TFmain'),'ТБ1 (со сборки)');
  //поиск окна с названием программы
  handleFDrawing:= FindWindow(PWideChar('TFmain'),'ТБ1 (с чертежа)');
  HandleFActiveList:= FindWindow(PWideChar('TFmain'),'ТБ1 (Активный лист)');
  //если найдено окно с таким же названием вывод сообщения
  if ((handleFAssembly <> 0) or (handleFDrawing <> 0) or (HandleFActiveList
<> 0)) then
  begin
    ShowKompasMessage('Приложение уже запущено.', );
    if handleFAssembly <> 0 then
    begin
      SetFocus(handleFAssembly);
    end
    else if handleFDrawing <> 0 then
    begin
      SetFocus(handleFDrawing);
    end
    else if HandleFActiveList <> 0 then
    begin
      SetFocus(HandleFActiveList);
    end;
    Exit;
  end;

  Application.Handle := GetHWND;
  GetNewKompasAPI;

  case command of
    1: CreateLineDimension;
  end;
end;

```

Рис. 1. Процедура LIBRARYENTRY

Процедура модуля CreateLineDimension, с помощью которого проверяется активный документ (если запущена сборка (Рис. 2), то данные считаются со сборки (в качестве примера тестовой сборки приводится термопистолет), если чертеж – с чертежа), приводится на Рис. 3. Вызываются функции, с помощью которых считываются данные со сборки и с чертежа, GetAssemblyComponents и GetDrawingComponents (Рис. 4), все полученные данные заносятся в ComponentsList (Рис. 5).

С помощью процедуры FormCreate создается форма и ее компоненты (Рис. 6 и Рис. 7), из ComponentsList данные заносятся в интерфейс приложения компонент TreeList (Рис. 8).

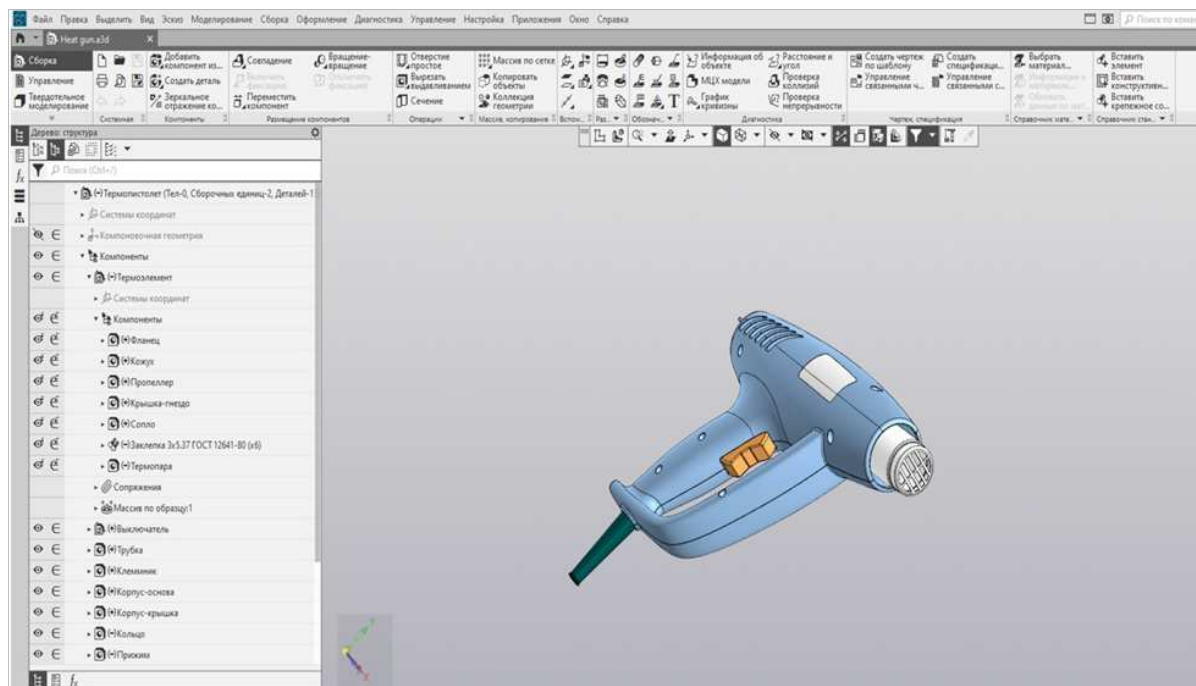


Рис. 2. Тестовая сборка

```

procedure CreateLineDimension;
var
  SR: TSearchRec;
begin
  GetProgramPath; //Получения пути до программы

  //Для загрузки операций и настроек
  if FindFirst(ProgramPath, faDirectory, SR) <> 0 then
  begin
    ShowKompasMessage('Не найден корневой каталог. Требуется переустановка
программы.', 2);
    exit;
  end;
  kDoc := newKompasAPI.ActiveDocument;
  if (kDoc <> nil) then if (kDoc.DocumentType = ksDocumentAssembly) or
(kDoc.DocumentType = ksDocumentDrawing) then
  begin
    GetSettingsVariables; //получение настроек с файла
    ReportMemoryLeaksOnShutdown := true;
    //листы для хранения данных
    OperList:= TList<TOper>.create; //операции
    ComponentsList:= TList<TComponents>.create; //запись с компаса
    OperationsList:= TList<TOperations>.Create; //загружаем с трилиста на
чертеж
    //получение типа документа
    if (kDoc.DocumentType = ksDocumentAssembly) then
    begin
      //если документ сборки, вызов функции считывания со сборки
      ComponentsList:= GetAssemblyComponents(ComponentsList);
      //создание основной формы
      FMainTable:= TFMain.Create(ComponentsList, 1);
      FMainTable.Caption:= 'ТБ1 (со сборки)';
    end
    //если документ чертеж
    else if (kDoc.DocumentType = ksDocumentDrawing) then
    begin
      //обработчик ошибки при считывании таблицы
      try
        GetDrawingComponents(ComponentsList);
      except on E: Exception do
        showkompasmessage('Ошибка при считывании с чертежа. Не найдена
таблица. ' + #10 + e.Message, 1);
      end;
      //создание основной формы
      FMainTable:= TFMain.Create(ComponentsList, 0);
      //название формы
      FMainTable.Caption:= 'ТБ1 (с чертежа)';
    end;
    //показываем форму
    FMainTable.Show;
  end
  else
  begin
    //если документ не является чертежом или сборкой вывод сообщения
    ShowKompasMessage('Откройте сборку или чертеж.', 2);
  end;
end;

```

Рис.3. Процедура модуля CreateLineDimension


```

function GetAssemblyComponents(List: TList<TComponents>): TList<TComponents>;
var
  c, i, a, level: integer;
  topPart: IPart7;
  Details : OleVariant;
  assembly : IKompasDocument3D;
  dispatchDetail : IDispatch;
  propMng : IPropertyMng;
  part7 : IPart7;
  Components: TComponents;
  subassembliesPartsList : TList<TPart>;
  propKeeper : IPropertyKeeper;
begin
  //получение интерфейсов для по
  if KDoc <> nil then assembly:= kDoc as IKompasDocument3D;
  //лист для промежуточных данных
  subassembliesPartsList := TList<TPart>.Create;
  //менеджер свойств документа
  propMng := newKompasAPI as IPropertyMng;
  if (assembly <> nil) then
  begin
    //поиск свойства TEl, для автомат. проставления галочек в программе
    FindProperty(propMng, assembly);
    //получение первой детали в дереве
    topPart := GetTopPart7(assembly);
    if (topPart <> nil) then
    begin
      //массив деталей
      Details := TopPart.PartsEx[0];
      if VarType(Details) = (varArray or varDispatch) then
      begin
        c := VarArrayHighBound(Details, 1) - VarArrayLowBound(Details, 1);
        Components.designation:= topPart.Name;
        Components.notation:= topPart.Marking;
        Components.Atype:= 'Сборка';
        Components.count:= c;
        Components.level:= 0;
        List.Add(Components);
        //считывание свойств деталей из массива
        for i := 0 to c do
        begin
          //уровень вложенности деталей (в данный момент функция не нужна)
          level:= 0;
          dispatchDetail := Details[i];
          if(dispatchDetail <> nil)then
          begin
            part7 := dispatchDetail as IPart7;
            if (part7 <> nil) then
            begin
              subassembliesPartsList.Clear;
              //рекурсивное считывание дерева, если деталь имеет item-ы, то
              //вызываем еще раз и записываем в лист
              Recurs(part7, subassembliesPartsList, level+1);
              for a := 0 to subassembliesPartsList.Count - 1 do
              begin
                propKeeper := subassembliesPartsList.items[a].Part as
                IPropertyKeeper;
                //получение свойств - наименование, обозначение и материал и
                запись
                Components:= GetPropertiesFound(propKeeper, propMng,
                assembly, a, subassembliesPartsList);
                if Components.designation <> '' then
                begin
                  //добавление в лист
                  List.Add(Components);
                end;
              end;
            end;
          end;
        end;
      end;

      SortList:= TList<TComponents>.Create;
      //объединение одинаковых позиций
      List:= CompleteTable(List);

      subassembliesPartsList.Free;
      ComponentsCount:= List.Count;
      result:= list;
    end;
  end;
end;
end;
end;

```

Рис.4. Реализация GetAssemblyComponents

⊞ [0]	(Термопистолет, '23.00.00.00.00', '1', 'Сборка', "", "", 0, 16 {\$10}, 0)
⊞ [1]	(Скелетная геометрия, "", '1', 'Деталь', "", "", 1, 0, 0)
⊞ [2]	(Фланец, '23.01.01.00.00', '1', 'Полиамид П-125-20 ТУ 6-05-898-73', 'Деталь', "", "", 2, 0, 0)
⊞ [3]	(Кожух, '23.02.00.00.00', '1', 'Полиамид П-125-20 ТУ 6-05-898-73', 'Деталь', "", "", 2, 0, 0)
⊞ [4]	(Пропеллер, '23.01.03.00.00', '1', 'Полиамид П-125-20 ТУ 6-05-898-73', 'Деталь', "", "", 2, 0, 0)
⊞ [5]	(Крышка-гнездо, '23.01.04.00.00', '1', 'Полиамид П-125-20 ТУ 6-05-898-73', 'Деталь', "", "", 2, 0, 0)
⊞ [6]	(Сопло, '23.01.05.00.00', '1', 'Сталь 10 ГОСТ 1050-2013', 'Деталь', "", "", 2, 0, 0)
⊞ [7]	(Термопара, '23.01.06.00.00', '1', 'Олово ГОСТ 860-75', 'Деталь', "", "", 2, 0, 0)
⊞ [8]	(Выключатель BNG-2000, '23.02.01.00.00', '1', 'Полиамид П-125-20 ТУ 6-05-898-73', 'Деталь', "", "", 2, 0, 0)
⊞ [9]	(Клавиша, '23.02.02.00.00', '1', 'Полиамид П-125-20 ТУ 6-05-898-73', 'Деталь', "", "", 2, 0, 0)
⊞ [10]	(Трубка, '23.03.00.00.00', '1', 'Смесь резиновая 3063 ТУ 38-1051082-86', 'Деталь', "", "", 1, 0, 0)
⊞ [11]	(Клемник, '23.04.00.00.00', '1', 'Полиэтилен НД ГОСТ 16338-85', 'Деталь', "", "", 1, 0, 0)
⊞ [12]	(Корпус-основа, '23.05.00.00.00', '1', 'Полиамид П-125-20 ТУ 6-05-898-73', 'Деталь', "", "", 1, 0, 0)
⊞ [13]	(Корпус-крышка, '23.06.00.00.00', '1', 'Полиамид П-125-20 ТУ 6-05-898-73', 'Деталь', "", "", 1, 0, 0)
⊞ [14]	(Кольцо, '23.07.00.00.00', '1', 'Сталь 10 ГОСТ 1050-2013', 'Деталь', "", "", 1, 0, 0)
⊞ [15]	(Прижим, '23.08.00.00.00', '1', 'Сталь 10 ГОСТ 1050-2013', 'Деталь', "", "", 1, 0, 0)

Рис.5. Данные, считанные со сборки

```

procedure TFmain.FormCreate(Sender: TObject);
var
  p, i, count: integer;
  nd: TcxTreeNode;
  s: string;
  col: TcxTreeListColumn;
begin
  SetColumnFormat; //включаем колонку с номерами
  TreeList:= TList<TNodeRec>.create;
  tlMaterials.BeginUpdate;
  tlMaterials.Clear;
  tlMaterials.DefaultRowHeight:= 30;

  if _TypeDoc = 1 then
  begin
    _ComponentsList:= SortingNotation(_ComponentsList);

    if stAssembly then
    begin
      nd := tlMaterials.Add;
      nd.Values[tlMaterialsType.ItemIndex]:= 'kompas';
      nd.Values[tlMaterialsDES.ItemIndex]:= _ComponentsList.Items[0].notation
+ ' ' + _ComponentsList.Items[0].designation;
      nd.Values[tlMaterialsType.ItemIndex]:= 'Сборка';

      count:= 0;
      for p := 1 to _ComponentsList.Count - 1 do
        if _ComponentsList.Items[p].level = 1 then
          count:= count + 1;

      i:=1;
      while i <> count+1 do
      begin
        Recurs(nd, i, _ComponentsList);
        i:= i + 1;
      end;
    end
  else
  begin
    for i := 0 to _ComponentsList.Count - 1 do
    begin
      if _ComponentsList.Items[i].atype <> 'Сборка' then
      begin
        nd := tlMaterials.Add;
        if (_ComponentsList.Items[i].bool = 'Her') or
        (_ComponentsList.Items[i].bool = '') then
        begin
          nd.Values[tlMaterialsCHECK.ItemIndex]:= false;
        end
        else
        begin
          nd.Values[tlMaterialsCHECK.ItemIndex]:= true;
        end;
      end;
    end;
  end;
end;

```

Рис.6. Процедура FormCreate (начало кода)


```

nd.Values[tlMaterialsType.ItemIndex]:= 'kompas';
nd.Values[tlMaterialsDes.ItemIndex]:=
_ComponentsList.Items[i].designation;
nd.Values[tlMaterialsNOT.ItemIndex]:=
_ComponentsList.Items[i].notation;
nd.Values[tlMaterialsGOST.ItemIndex]:=
MaterialCut(_ComponentsList.Items[i].material, 'GOST');
nd.Values[tlMaterialsMAT.ItemIndex]:=
MaterialCut(_ComponentsList.Items[i].material, 'MAT');
nd.Values[tlMaterialsAMO.ItemIndex]:=
_ComponentsList.Items[i].amount;
end;
end;
end;
SetNumber;
end
else if _TypeDoc = 0 then
begin
AddOperations(OperList);
for i := 0 to _ComponentsList.Count - 1 do
begin
nd:= tlMaterials.Add;
if (_ComponentsList.Items[i].designation = '-----') and
(_ComponentsList.Items[i].notation = '-----') and
(_ComponentsList.Items[i].material = '-----') then
begin
nd.Values[tlMaterialsType.ItemIndex]:= '';
end
else
nd.Values[tlMaterialsType.ItemIndex]:= 'kompas'; //Чтобы можно
было удалить строки обозначающие перенос на некст лист
CheckStrikeOutCell(i, _ComponentsList);
nd.Values[tlMaterialsCHECK.ItemIndex]:= true;
nd.Values[tlMaterialsNUM.ItemIndex]:= _ComponentsList.Items[i].Number;
//nd.Values[tlMaterialsNUM.ItemIndex]:= inttostr(i+1);
nd.Values[tlMaterialsDes.ItemIndex]:=
_ComponentsList.Items[i].designation;
nd.Values[tlMaterialsNOT.ItemIndex]:=
_ComponentsList.Items[i].notation;
nd.Values[tlMaterialsAMO.ItemIndex]:= _ComponentsList.Items[i].amount;
nd.Values[tlMaterialsMAT.ItemIndex]:=
_ComponentsList.Items[i].material;
nd.Values[tlMaterialsGOST.ItemIndex]:= _ComponentsList.Items[i].GOST;
if stAddColumnNote then
nd.Values[tlMaterials.ColumnByName('tlMaterialsNOTE').ItemIndex]:=
_ComponentsList.Items[i].Note;
for p := 0 to
length(_ComponentsList.Items[i].Operations){Length(ArrColOperation)} - 1 do
begin
nd.Values[ArrColOperation[p].ItemIndex]:=
_ComponentsList.Items[i].Operations[p];
end;
end;
end;
tlMaterials.EndUpdate;
end;
end;

```

Рис.7. Процедура FormCreate (конец кода)

ТБ1 (со сборки)

№п	Наименование детали или сборочной единицы	Количество деталей	Обозначение чертежа	Материал	
				Обозначение стандарта	Материал группа испытаний
<input type="checkbox"/>	Фланец	1	23.01.01.00.00	Полиамид П-125-20 ТУ 6-05-898-73	
<input type="checkbox"/>	Пропеллер	1	23.01.03.00.00	Полиамид П-125-20 ТУ 6-05-898-73	
<input type="checkbox"/>	Крышка-гнездо	1	23.01.04.00.00	Полиамид П-125-20 ТУ 6-05-898-73	
<input type="checkbox"/>	Сопло	1	23.01.05.00.00	ГОСТ 1050-2013	Сталь 10
<input type="checkbox"/>	Термопара	1	23.01.06.00.00	ГОСТ 860-75	Олово
<input type="checkbox"/>	Кожух	1	23.02.00.00.00	Полиамид П-125-20 ТУ 6-05-898-73	
<input type="checkbox"/>	Выключатель ВНК-2000	1	23.02.01.00.00	Полиамид П-125-20 ТУ 6-05-898-73	
<input type="checkbox"/>	Клавиша	1	23.02.02.00.00	Полиамид П-125-20 ТУ 6-05-898-73	
<input type="checkbox"/>	Трубка	1	23.03.00.00.00	Смесь резиновая 3063 ТУ 38-1051082-86	
<input type="checkbox"/>	Клениник	1	23.04.00.00.00	ГОСТ 16338-85	Полиэтилен НД
<input type="checkbox"/>	Корпус-основа	1	23.05.00.00.00	Полиамид П-125-20 ТУ 6-05-898-73	
<input type="checkbox"/>	Корпус-крышка	1	23.06.00.00.00	Полиамид П-125-20 ТУ 6-05-898-73	
<input type="checkbox"/>	Кольцо	1	23.07.00.00.00	ГОСТ 1050-2013	Сталь 10

Рис. 8. Данные, считанные со сборки в интерфейсе приложения

Интерфейс окна приложения (Рис. 8) содержит панель со следующими кнопками:

1) «Сформировать таблицу» - вызов процедуры, отвечающей за формирование таблицы на чертеже. После нажатия на эту кнопку запускается процедура CreateDrawingDocument, которая создает документ чертежа и выполняет несколько проверок (Рис.9, Рис.10). Процедура создания нового листа, если данные не помещаются на одном листе, показана на Рис. 11

```

procedure CreateDrawingDocument;
var
  i, rowCount, SheetNumb: Integer;
  KDoc: IKompasDocument;
  kViews: IViews;
  kView: IView;
  LSheet: ILayoutSheet;
  LSheets: ILayoutSheets;
  DManager: IViewsAndLayersManager;
  kStamp: IStamp;
  nd: tckTreeListNode;
  sformat: ShortInt;
begin
  if newKompasAPI <> nil then
  begin
    KDoc:= newKompasAPI.ActiveDocument;
    if KDoc.DocumentType = ksDocumentDrawing then
    begin
      if KDoc <> nil then KompasDocument2d:= KDoc as IKompasDocument2D;
      if KompasDocument2d <> nil then DManager:=
KompasDocument2d.ViewsAndLayersManager;
      if DManager <> nil then kViews:= DManager.Views;
      LSheets:= KompasDocument2d.LayoutSheets;

      if CheckTableOnDrawing(kViews) then
      begin
        if newKompasAPI.MessageBoxEx('Ведомость уже сформирована, вы хотите
перезаписать данные?',
'Sообщение',MB_YESNO + MB_ICONINFORMATION ) = idYes then
        begin
          ViewsDelete(kViews, LSheets);
          setSheetFormat(LSheets.Item[0]);
        end
        else
        begin
          exit;
        end;
      end
      else
      begin
        ViewsDelete(kViews, LSheets);
      end;

      if LSheets.Count = 1 then
      begin
        setSheetFormat(LSheets.Item[0]);
      end;
    end
    else
    begin
      kDoc:= newKompasAPI.Documents.Add(ksDocumentDrawing, true);
      if KDoc <> nil then KompasDocument2d:= KDoc as IKompasDocument2D;
      LSheet:= KompasDocument2d.LayoutSheets.Item[0];
      if LSheet <> nil then
      begin
        setSheetFormat(LSheet);
        LSheet.Format.VerticalOrientation:= false;
        kStamp:= LSheet.Stamp;
        kStamp.Text[2].Str:= ComponentsList.Items[0].notation + ' ' +
ComponentsList.Items[0].designation;
        kStamp.Text[1].Str:= 'Таблица контроля качества основного металла';
        kStamp.update;
        LSheet.Update;
      end;
    end;
  end;
end;

```

Рис. 9. Процедура CreateDrawingDocument (начало кода)

```

if stAssembly then
begin
nd:= FMaintable.tlMaterials.Items[0];
TreeListrecurs(nd, OperationsList);
end
else
begin
TreeListGetValues(OperationsList);
DeleteDelimiter(OperationsList);
end;

SheetNumb:= 0;
while OperationsList.count <> 0 do //Создание и заполнение таблицы
begin
if SheetNumb = 0 then
begin
CreateTable(SheetNumb, stCountStrFirst);
rowCount:= DeleteRowTable(table.RowsCount, 64, SheetNumb+1);
end
else
begin
CreateNewList(KompasDocument2d); //Создание нового листа
CreateTable(SheetNumb, stCountStrNext); //создание вида и таблицы на
нем
rowCount:= DeleteRowTable(table.RowsCount, 22, SheetNumb+1);
end;
for i := 0 to rowCount - 5 do
begin
OperationsList.Delete(0);
end;
SheetNumb:= sheetnumb + 1;
end;
end;
OperationsList.clear;
end;

```

Рис.10. Процедура CreateDrawingDocument (конец кода)

```

Procedure CreateNewList(KDoc2D: IKompasDocument2D); //Создание нового листа
var
LSheet: ILayoutSheet;
begin
//ChDir('..\Sys\');
LSheet:= KDoc2D.LayoutSheets.Add;
if LSheet <> nil then
begin
if (FileExists(TPath.Combine(GetCurrentDir, 'GRAPHIC.LYT'), true)) then
LSheet.LayoutLibraryFileName:= 'C:\Program Files\ASCON\KOMPAS-3D
v22\Sys\graphic.lyt';
LSheet.LayoutStyleNumber:= 2;
if stSheetFormat = 'A3' then
LSheet.Format.Format:= ksFormatA3
else if stSheetFormat = 'A4x3' then
begin
LSheet.Format.Format:= ksFormatA4;
LSheet.Format.FormatMultiplicity:= 3;
end
else if stSheetFormat = 'A4x4' then
begin
LSheet.Format.Format:= ksFormatA4;
LSheet.Format.FormatMultiplicity:= 4;
end;
LSheet.Format.VerticalOrientation:= false;
LSheet.Update;
end;
end;

```

Рис.11. Процедура создания нового листа

Процедура создания самой таблицы приведена на Рис. 12 - Рис. 14.

```

procedure CreateTable(SheetNumb, RowCount: integer); //Создание и заполнение
таблицы
var
del: integer;
begin
if RowCount > OperationsList.count then
RowCount:= OperationsList.Count;

del:= FindDelimiter;
if del > 0 then
begin
if del <= RowCount then
begin
RowCount:= del;
OperationsList.Delete(del);
end;
end;

DrawTable(CreateView(KompasDocument2d, SheetNumb), RowCount);
end;

```

Рис.12. Процедура создания таблицы (начало кода)

```

function DrawTable(kView: IView; RowCount: integer): integer; //Рисование
таблицы и заполнение шапки

procedure CreateNumberColumn;
var
  i: integer;
begin
  table.AddColumn(0, false);
  AddCell(0,0,11,69,table,2);
  AddCell(0,1,29,69,table,2);
  AddText('Мн',0,0,3.5,table, false, false);
  for i := 0 to RowCount - 1 do
    AddText(inttostr(i+1),i+4,0,3.5,table, False, false);
  DrawingTable.Update;
end;

var
  i, n, ColCount: integer;
  kf: double;
  DrawSymbCont: ISymbols2DContainer;
  iCellBound: ICellBoundaries;
begin
  if kView <> nil then DrawSymbCont:= kView as ISymbols2DContainer;
  DrawingTable:= DrawSymbCont.DrawingTables.Add(4, 1, 8, 5, ksTTLFirstRow);
  DrawingTable.x:= 23;
  DrawingTable.y:= 276;

  table:= DrawingTable as ITable;
  for i := 0 to 4 do
    table.AddColumn(table.ColumnsCount - 1,true);

  table.range[0,3,0,4].CombineCells;
  table.range[0,0,3,0].CombineCells;
  table.range[0,1,3,1].CombineCells;
  table.range[0,2,3,2].CombineCells;
  table.range[1,3,3,3].CombineCells;
  table.range[1,4,3,4].CombineCells;

  OperCount:= Trunc((FMainTable.tlMaterials.Bands.Count - 8)/2);
  if stAddColumnNote then
  begin
    if stSheetFormat = 'A4x3' then
      kf:= Simpleroundto(386/Opercount, -2)
    else if stSheetFormat = 'A3' then
      kf:= Simpleroundto(176/Opercount, -2)
    else if stSheetFormat = 'A4x4' then
      kf:= Simpleroundto(596/Opercount, -2);
    ColCount:= FMainTable.tlMaterials.Bands.Count - 2;
  end
  else
  begin
    if stSheetFormat = 'A4x3' then
      kf:= Simpleroundto(416/Opercount, -2)
    else if stSheetFormat = 'A3' then
      kf:= Simpleroundto(206/Opercount, -2)
    else if stSheetFormat = 'A4x4' then
      kf:= Simpleroundto(626/Opercount, -2);
    ColCount:= FMainTable.tlMaterials.Bands.Count - 1;
  end;
  if stReadColumnWidth and (FMainTable.Caption = 'ТБ1 (с чертёжа)') then
  begin
    AddCell(0,0,WDesignation,69,table,2); // Наименование
    AddCell(0,1,WAmount,69,table,2); // Количество
    AddCell(0,2,WNotation,69,table,1); // Обозначение
    AddCell(1,3,WMaterial,69,table,2); // Материал
    AddCell(1,4,WGost,69,table,2); // ГОСТ
  end
  else
  begin
    //if stAddColumnNumber then AddCell(0,0,40,69,table,2); // Номер
    AddCell(0,0,40,69,table,2); // Наименование
    AddCell(0,1,11,69,table,2); // Количество
    AddCell(0,2,55,69,table,1); // Обозначение
    AddCell(1,3,44,69,table,2); // Материал
    AddCell(1,4,32,69,table,2); // ГОСТ
  end;

  AddCell(1,5,kf,47,table,0);
  AddCell(3,5,kf,7,table,2);
  AddCell(2,0,0,0,Table,2);

```

Рис.13. Процедура создания таблицы (продолжение кода)

```

AddText('Наименование детали или сборочной единицы',0,0,3.5,table, false,
false);
AddText('Количество деталей',0,1,3.5,table, true, false);
AddText('Обозначение чертежа',0,2,5,table, false, false);
AddText('Материал',0,3,5,table, false, false);
AddText('Материал, группа испытаний',1,3,3.5,table, false, false);
AddText('Обозначение стандарта',1,4,3.5,table, false, false);
AddText('Наименование операции',0,5,5,table, false, false);
AddText('Шифр операции',2,5,3.5,table, false, false);

for i := 0 to OperCount - 2 do //колонки для операций
    table.AddColumn(table.ColumnsCount - 1,true);
    table.range[0,5,0,table.ColumnsCount - 1].CombineCells;
    table.range[2,5,2,table.ColumnsCount - 1].CombineCells;

for i := 8 to Round(OperCount) + 7 do //название операции
begin
    AddText(FMainTable.tlMaterials.Bands[i].Caption.Text,1,i-3,3.5,table,
true, false);
end;

for i := Round(OperCount) + 8 to ColCount do //номер операции
begin
    AddText(FMainTable.tlMaterials.Bands[i].Caption.Text,3,i-3-
OperCount,3.5,table, false, false);
end;

if stAddColumnNumber then //если включена колонка номеров, то добавляем
слева
begin
    ColNum:= 1;
    table.AddColumn(0, false);
    if stReadColumnWidth and (FMainTable.Caption = 'ТБ1 (с чертежа)') then
    begin
        AddCell(0,0,WNumber,69,table,2);
        AddCell(0,1,WDesignation,69,table,2)
    end
    else
    begin
        AddCell(0,0,11,69,table,2);
        AddCell(0,1,29,69,table,2);
    end;
    AddText('Мн',0,0,3.5,table, false, false);
end
else
begin
    ColNum:= 0;
end;

iCellBound:= Table.Cell[2,5+ColNum] as ICellBoundaries;
iCellBound.LineStyle[kcCBAllBorders]:= kcCSNormal;

if stAddColumnNote then
begin
    table.AddColumn(table.ColumnsCount - 1, true);
    table.range[0,table.ColumnsCount - 1,3,table.ColumnsCount -
1].CombineCells;
    if stReadColumnWidth then
        AddCell(0,table.ColumnsCount - 1,WNote,69,table,2)
    else
        AddCell(0,table.ColumnsCount - 1,30,69,table,2);
    AddText('Примечание',0,table.ColumnsCount - 1,3.5,table, false, false);
end;

if stReadColumnWidth and (Length(WArray) <> 0) and (FMainTable.Caption =
'TБ1 (с чертежа)') then //считывание длины колонок
    SetOperationWidth;

for i := 3 to RowCount - 1 + 3 do
begin
    table.AddRow(i,true);
end;
DrawingTable.Update;

AddToTable(OperationsList, RowCount); //Добавление текста в таблицу
DrawingTable.Update;

end;

```

Рис.14. Процедура создания таблицы (конец кода)

2) «Добавить операции» - добавление операций и их шифров. При нажатии на данную кнопку появляется новая форма «Выбор операций» (Рис. 15), где пользователь выбирает необходимые операции или добавляет новые. При нажатии на «Добавить операцию» открывается окно с формой для заполнения, в которую пользователь вводит наименование операции и ее шифр. Отмена добавления операции осуществляется с помощью кнопки «Отмена» или закрытием активной формы. Нажатие кнопки «Ок» запускает процедуру добавления операции в форму, операция добавляется в конец списка и выводится в отсортированном виде по шифру.

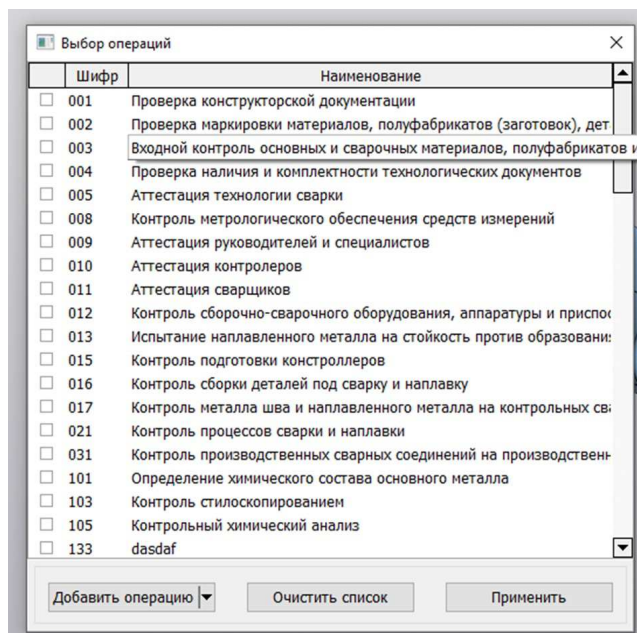


Рис.15. Форма «Выбор операций»

Нажатие на кнопки «Применить» приводит к тому, что форма «Выбор операций» закрывается и операции добавляются в основную форму (Рис. 16).

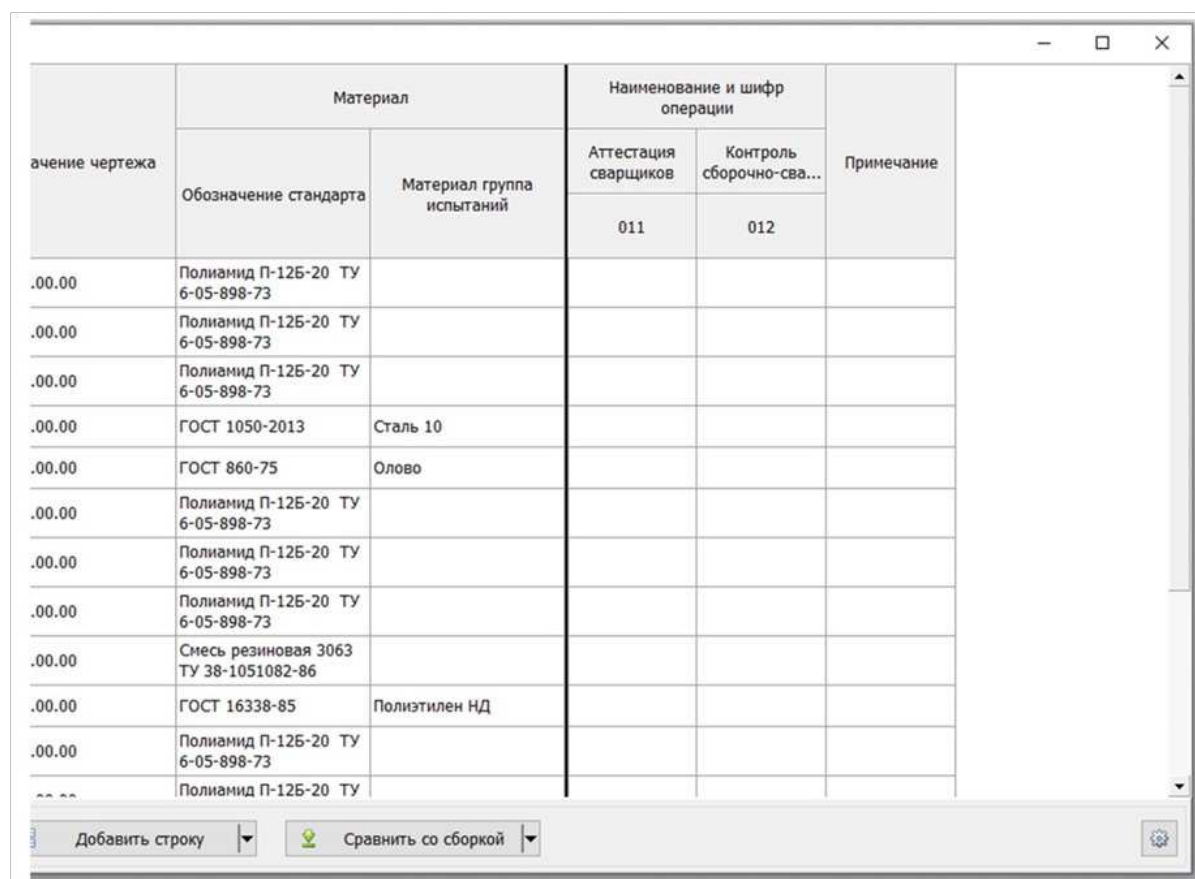


Рис.16. Добавленные операции в интерфейсе приложения

Нажатие на ячейку под операцией, приводит к появлению выпадающего списка с символами для вставки в эту ячейку.

Все операции хранятся в текстовом файле, который находится в каталоге с приложением (Рис. 17). При запуске, приложение считывает все операции и заносит в форму, после закрытия операции сохраняются в файле OperationList.txt.

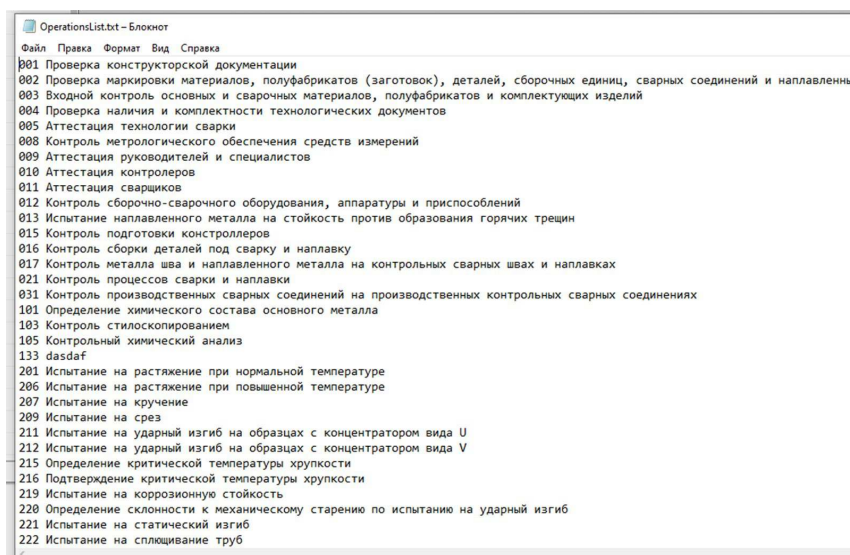


Рис.17. Файл с операциями

Процедуры, которые срабатывают после нажатия на кнопку «Применить», показаны на Рис. 18, Рис. 19.

```
procedure TFOperations.bApplyClick(Sender: TObject);
var
  i: Integer;
  operRec: TOper;
begin
  for i := 0 to tloper.Count - 1 do
  begin
    //проверка на выбранный компонент
    if not varIsNull(tloper.Items[i].Values[tlColChecks.ItemIndex]) then
      if tloper.Items[i].Values[tlColChecks.ItemIndex] then
        begin
          //запись в лист
          operRec.name:=
            varToStr(tloper.Items[i].Values[tlColOperation.ItemIndex]);
          operRec.number:=
            varToStr(tloper.Items[i].Values[tlColNumber.ItemIndex]);
          operRec.bool:= true;
          operlist.Add(operrec);
        end;
      end;
    //процедура добавления операций на главной форме
    FmainTable.AddOperatios(OperList);
    //закрытие формы операций
    self.Close;
  end
end
```

Рис.18. Процедуры, срабатывающие при нажатии на кнопку «Применить» (начало кода)

```

procedure TFmain.AddOperations(list: TList<TOper>);
var
  p, i, ln: integer;
  AColumn, LastCol: TcxTreeListColumn;
  AComboProperties: TcxComboBoxProperties;
  CF: TCharFormat;
  MS: TMemoryStream;
begin
  if tlMaterials.Bands.Count > 7 then //удаление бендов операций
  begin
    for i := 7 to tlMaterials.Bands.Count - 1 do
    begin
      tlMaterials.Bands.Items[7].Destroy;
    end;

    for i := 0 to length(ArrColOperation) - 1 do
    begin
      ArrColOperation[i].Destroy;
    end;

    Setlength(ArrColOperation, 0);
    try
      tlMaterials.ColumnByName('tlMaterialsNOTE').Destroy;
    except
      end;
  end;

  //Если лист пустой завершаем процедуру
  if list.Count = 0 then
    exit;

  BandCreate('Наименование и шифр операции', -1, 0, false);
  for i := 0 to list.Count - 1 do //бенды названий операций
  begin
    BandCreate(list.Items[i].name, 7, i+1, true);
  end;

  SetLength(ArrColOperation, list.Count);
  for i := 0 to list.Count - 1 do //бенды с шифрами операций
  begin
    BandCreate(list.Items[i].number, 8+i, 0, false);

    AColumn:= tlMaterials.CreateColumn();
    ArrColOperation[i]:= AColumn;

    //настройки колонки: позиция, шрифт, класс и тд
    AColumn.Position.BandIndex:= 8+list.count+i;
    AColumn.DataBinding.ValueTypeClass := TcxStringValue;
    AColumn.PropertiesClass := TcxComboBoxProperties;
    AComboProperties := AColumn.Properties as TcxComboBoxProperties;
    AComboProperties.Items.Add('+C');
    AComboProperties.Items.Add('-');
    AComboProperties.Items.Add('+$cn./it.7$');
    AColumn.Options.VertSizing:= false;
    AColumn.Properties.Alignment.Horz:= taCenter;
  end;
  //если включена настройка "колонка примечание"
  if stAddColumnNote then //бенд "примечание"
  begin
    BandCreate('Примечание', -1, 6, false);
    AColumn:= tlMaterials.CreateColumn();
    AColumn.Position.BandIndex:= tlMaterials.Bands.Count - 1;
    AColumn.name:= 'tlMaterialsNOTE';
  end;
end;

```

Рис.19. Процедуры, срабатывающие при нажатии на кнопку «Применить» (конец кода)

3) «Сравнить со сборкой» - сравнение считанных данных с теми, которые на текущий момент находятся в сборке, отличающиеся данные выделяются голубым цветом.

4) «Добавить строку» - добавление новой строки выше выбранной. Эта кнопка имеет выпадающий список, где находятся дополнительные кнопки для редактирования данных («Добавить строку снизу», «Начать с нового листа», «Удалить строку», «Выбрать все элементы», «Зачеркнуть текст в строке», «Зачеркнуть текст в ячейке»).

Результаты разработки. В результате последовательного выполнения шагов описываемого порядка разработки приложения, пользователь получит новый документ чертежа, со сформированной таблицей контроля качества основного металла ТБ1 (Рис. 20). Для примера, представленного на Рис.2, в таблице приводятся следующие сведения и данные: наименование детали или сборочной единицы, количество деталей, обозначение чертежа, материал (материал, группа испытаний; обозначение стандарта), наименование операции (входной контроль основных и сборочных материалов, полуфабрикатов и комплектующих изделий; проверка наличия и комплектности технологических документов; контроль метрологического обеспечения средств изменений; аттестация руководителей и специалистов; аттестация контролеров; контроль сборочно-сварочного оборудования, аппаратуры и приспособлений; испытание

наплавленного металла на стойкость против образования горячих трещин).

[illegible]

Рис. 20. Пример таблицы контроля качества основного металла

Разработанное прикладное пользовательское приложение автоматически считывает данные с файла сборки открытого в КОМПАС-3D, создает новый документ чертежа, осуществляет построение ТБ1, с занесением данных, взятых со сборочного документа.

Научная новизна полученных результатов. Разработанная процедура построения таблицы ТБ1 в КОМПАС-3D (в условиях ограниченной информации об алгоритмах составления таблиц контроля качества в свободном доступе и затруднениях возникающих у специалистов проектов, составляющих конструкторскую документацию таблицы ТБ1 и, на их основании, программ качества), в отличие от известных, позволяет осуществлять построение автоматически.

Заключение. В работе приведен порядок разработки прикладного пользовательского приложения для КОМПАС-3D, позволяющего в автоматизированном режиме сформировать таблицу контроля качества основного металла (ТБ1). Для реализации используется программный код, написанный в программной среде Delphi. Данное приложение может быть использовано предприятиями (организациями), специализирующимися на проектировании и изготовлении оборудования для атомной энергетики, АЭС.

Разработанное пользовательское приложение отвечает критерию гибкости, а также быстроты, в условиях возможности внесения оперативных изменений. Процесс разработки осуществлялся итеративно, что позволяло расширять функционал. Например, в качестве расширения возможностей кода, может быть предусмотрено использование процедуры считывания данных с одного листа (активным листом считается лист, выбранный в дереве) – приложение переходит в режим «ТБ1 Активный лист», в котором кнопка «Построить таблицу» заменяется на кнопку «Обновить данные на активном

листе». С помощью данного режима пользователь может легко изменить данные всего лишь на одном листе.

За счет универсальности программного кода, возможно его применение не только для КОМПАС-3D, но и для других CAD-систем.

Также в качестве перспектив дальнейшего использования для проектов по созданию прикладных библиотек для КОМПАС-3D могут быть взяты за основу принципы написания программного кода процедур разработки, что входит в область интересов как предприятий (например, для действующих конструкторов, специалистов, повышающих квалификацию в соответствующей области) и конструкторских бюро, так и учебных заведений.

Библиография

1. Норсеев С. Разработка приложений под КОМПАС в Delphi, 2013. – 346 с.[Электронный ресурс] – URL: <https://norseev.ru/books/kompas-delphi-2013/>
2. Сошкина Д.Е. Разработка пользовательской библиотеки в Компас-3D / Д.Е. Сошкина; науч. рук. А.А. Дронов // Прогрессивные технологии и экономика в машиностроении : сборник трудов XIII Всероссийской научно-практической конференции для студентов и учащейся молодежи, 7-9 апреля 2022 г., Юрга. – Томск: Изд-во ТПУ, 2022. – С. 34-36.
3. Гуца А.О. К задаче о разработке на С# пользовательских библиотек стандартных деталей для КОМПАС-3D / А.О. Гуца, Т.Д. Стасенко ; науч. рук. В.С. Мурашко // Исследования и разработки в области машиностроения, энергетики и управления : материалы XXIII Междунар. науч. – техн. конф. студентов, аспирантов и молодых ученых, Гомель, 27–28 апр. 2023 г. : в 2 ч. Ч. 2 / М-во образования Респ. Беларусь, Гомел. гос. техн. ун-т им. П.О. Сухого ; под общ. ред. А.А. Бойко. – Гомель: ГГТУ им. П. О. Сухого, 2023. – С. 268-270.
4. Терехин Р.Д., Фоломкин А.И. Разработка программных библиотек для демонстрации и построения линий пересечения различных поверхностей / Технические науки – от теории к практике // Сб. ст. по материалам LXXI междунар. науч. – практ. конф. № 6 (66). Новосибирск: Изд. АНС «СиБАК», 2017. – С. 13-18.
5. Решетникова Е.С., Савельева И.А., Свистунова Е.А. Геометрическое моделирование и разработка пользовательских библиотек при проектировании объектов машиностроения // Программные системы и вычислительные методы. 2020. № 1. С. 1-7. DOI: 10.7256/2454-0714.2020.1.32292 URL: https://nbpublish.com/library_read_article.php?id=32292
6. Кидрук М.И. Конструкторские библиотеки и инструменты для их создания в системе КОМПАС-3D. Часть 1. В помощь конструктору – библиотеки, расчетные модули и справочники КОМПАС-3D // статья в журнале «САПР и графика». Машиностроение, № 1, 2006 [Электронный ресурс] – URL: <https://sapr.ru/article/14850>
7. Кидрук М.И. Конструкторские библиотеки и инструменты для их создания в системе КОМПАС-3D. Часть 2. Сделай сам, или Как создать свою библиотеку для КОМПАС-3D // статья в журнале «САПР и графика». Машиностроение, № 2, 2006 [Электронный ресурс] – URL: <https://sapr.ru/article/15598>
8. Форум пользователей ПО Аскон. [Электронный ресурс] – URL: <http://forum.ascon.ru/index.php>
9. ОСТ 108.004.10-86. Отраслевой стандарт. Программа контроля качества изделий атомной энергетики. – Действует с 01.01.2007 – М.: Изменение № 12, 2019. 143 с. 1. Норсеев С. Разработка приложений под КОМПАС в Delphi, 2013. – 346 с.[Электронный ресурс] – URL: <https://norseev.ru/books/kompas-delphi-2013>
10. РД ЭО 1.1.2.01.0713-2019. Оценка соответствия в формах приемки, испытаний

продукции для атомных станций. Положение. – Взамен РД ЭО 1.1.2.01.0713-2013 «Положение об оценке соответствия в форме приемки и испытаний продукции для атомных станций». – Введ. с 20.02.2020 – М.: Актуализация 01.01.2021. – 60 с.

11. Форум DWG.RU для обсуждения проблем проектирования, конструирования в области строительства, архитектуры, машиностроения, смежных отраслей. [Электронный ресурс] – URL: <https://forum.dwg.ru/showthread.php?t=167100>

Результаты процедуры рецензирования статьи

В связи с политикой двойного слепого рецензирования личность рецензента не раскрывается.

Со списком рецензентов издательства можно ознакомиться [здесь](#).

Представленная статья на тему «Разработка прикладного приложения для автоматизации проектирования в КОМПАС-3D: таблица контроля качества основного металла» соответствует тематике журнала «Программные системы и вычислительные методы» и посвящена вопросу разработки прикладного пользовательского приложения для Компас-3D позволяющее автоматически сформировать таблицы контроля качества основного металла (ТБ1). Авторами в статье указаны основные требования к функционалу пользовательского приложения, среди которых: запуск происходит со сборки в КОМПАС-3D через панель команд или через кнопки «Приложения» - «ТБ1»; после чего в окне программы отображается таблица с полученными со сборки деталями (также детали всех подборок), их количеством, обозначением, материалом и соответствующим стандартом; также программа должна быть функциональной, позволяющей конструктору менять все графы таблицы, выбирать те позиции, которые построятся в таблице на чертеже, добавлять новые позиции, удалять добавленные, зачеркивать текст и возможность разделить таблицу на несколько листов; в приложении должен быть предусмотрен выбор контрольных операций для материалов сборки, операций с шифрами, также добавление новых операций, с последующим их сохранением в памяти программы; иметься возможность выбора символов для операций по клику на соответствующую ячейку; по нажатию на соответствующую кнопку автоматически должен создаваться документ чертежа, с построенной таблицей со всеми изменениями, внесенными конструктором в интерфейс программы; в наличии должна быть возможность добавлять функционал считывания данных с чертежа для внесения изменений через интерфейс программы.

В статье авторами приведен порядок разработки прикладного пользовательского приложения для КОМПАС-3D, позволяющего в автоматизированном режиме сформировать таблицу контроля качества основного металла (ТБ1). Для реализации этого использовался программный код, написанный в программной среде Delphi, которая используется предприятиями (организациями), специализирующимися на проектировании и изготовлении оборудования для атомной энергетики, АЭС.

Стиль и язык изложения материала является достаточно доступным для широкого круга читателей. Практическая значимость статьи четко обоснована. Статья достаточно структурирована - в наличии введение, заключение, внутреннее членение основной части.

К недостаткам можно отнести следующие моменты: из содержания статьи не прослеживается научная новизна. Отсутствует четкое выделение предмета, объекта исследования. Статья не соответствует по объему минимальным требованиям (объем должен быть не менее 12 000 знаков).

Рекомендуется четко обозначить научную новизну исследования, сформулировать предмет, объект. Увеличить объем статьи до 12 000 знаков или более. Также будет

целесообразным добавить о перспективах дальнейшего исследования.

Статья «Разработка прикладного приложения для автоматизации проектирования в КОМПАС-3D: таблица контроля качества основного металла» требует доработки по указанным выше замечаниям. После внесения поправок рекомендуется к повторному рассмотрению редакцией рецензируемого научного журнала.

Результаты процедуры повторного рецензирования статьи

В связи с политикой двойного слепого рецензирования личность рецензента не раскрывается.

Со списком рецензентов издательства можно ознакомиться [здесь](#).

Статья посвящена разработке прикладного пользовательского приложения для автоматизации составления таблицы контроля качества основного металла (ТБ1) в САПР КОМПАС-3D. В рамках работы рассматривается создание программного продукта, который ускоряет и упрощает процесс формирования документации в области проектирования оборудования для атомной энергетики.

Автор применяет метод разработки программного обеспечения с использованием языка Delphi. Статья подробно описывает последовательность действий по созданию и интеграции библиотеки для КОМПАС-3D, взаимодействие с его API, а также механизмы автоматизации работы с данными сборки. Программная часть включает описание ключевых процедур, таких как чтение данных из сборки, создание чертежей, а также функционал для редактирования таблиц.

Процесс проектирования и составления таблиц контроля качества в атомной энергетике является крайне важным, так как от точности и правильности документации зависит качество изготовления и безопасности эксплуатации оборудования. Разработка такого прикладного приложения отвечает актуальным потребностям отрасли, а также способствует ускорению и автоматизации сложных инженерных процессов. Это решение актуально в условиях современных требований к проектированию и документированию в атомной энергетике.

Научная новизна работы заключается в предложении автоматизированного подхода к формированию таблиц контроля качества в КОМПАС-3D, что значительно улучшает точность и снижает время на составление документации. В отличие от существующих решений, это приложение позволяет динамично изменять данные в процессе разработки и легко интегрировать новые функциональные возможности в приложение.

Статья написана четким и последовательным стилем, доступным для целевой аудитории, которая включает как разработчиков программного обеспечения, так и специалистов, занимающихся проектированием в КОМПАС-3D. Работа логично структурирована: введение, постановка задачи, описание методов разработки, реализация и результаты. Статья сопровождается подробными иллюстрациями, которые помогают визуализировать ключевые этапы разработки. Это делает материал не только теоретически значимым, но и практически полезным.

Автор тщательно подобрал литературу, включая как фундаментальные работы, так и современные исследования, связанные с разработкой приложений для САПР КОМПАС-3D. Библиография отражает разнообразие источников, что повышает научную ценность работы. Однако для повышения уровня работы рекомендовано добавить более актуальные источники, связанные с новыми технологиями в области автоматизации проектирования и разработки программного обеспечения.

Статья не содержит явных слабых мест. Применяемая методология и разработка показывают высокий уровень знаний автора в области программирования и проектирования в КОМПАС-3D. Возможные улучшения могут касаться детализации

алгоритмов обработки данных и их оптимизации, а также расширения функционала приложения для использования в более широких контекстах, помимо атомной энергетики.

Статья представляет собой ценный вклад в развитие программных средств для автоматизации инженерных процессов в атомной энергетике. Практическая значимость работы высока, так как разработанный инструмент позволяет повысить эффективность работы инженеров, а также снизить вероятность ошибок при составлении таблиц контроля качества. Работу можно рекомендовать как для специалистов в области проектирования, так и для ученых, занимающихся развитием программных систем для инженерных задач. Статья может быть интересна широкой аудитории, включая как профессионалов, так и студентов, изучающих программное обеспечение для САПР.

Работа заслуживает публикации с учетом ее научной значимости и практической полезности. Рекомендуется принять статью к публикации с незначительными доработками, связанными с улучшением анализа существующих решений в области автоматизации проектирования и расширением функционала описанного приложения.