Министерство науки и высшего образования
Российской Федерации

Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Воронежский государственный технический университет»

Кафедра автоматизированного оборудования
машиностроительного производства

СОВРЕМЕННЫЕ СИСТЕМЫ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ В АВТОМАТИЗИРОВАННОМ ПРОИЗВОДСТВЕ

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

к выполнению лабораторных работ
по направлению 15.03.01 «Машиностроение»
(профиль «Технологии, оборудование и автоматизация
машиностроительных производств»)
очной и заочной форм обучения

Воронеж 2021

УДК 658.52:681.3:681.5(07)
ББК 32.97:32.8я7

Составитель ст. преп. С. Л. Новокщенов

Современные системы управления базами данных в автоматизированном производстве: методические указания к выполнению лабораторных работ по направлению 15.03.01 «Машиностроение» (профиль «Технологии, оборудование и автоматизация машиностроительных производств») очной и заочной форм обучения /ФГБОУ ВО "Воронежский государственный технический университет"; сост.: С. Л. Новокщенов. - Воронеж: Изд-во ВГТУ, 2021. 33 с.

Методические указания содержат сведения, необходимые для выполнения лабораторных работ по дисциплине «Современные системы управления базами данных в автоматизированном производстве». В них изложены основные положения курса, приведены необходимые теоретические сведения для выполнения лабораторных работ.

Предназначены для студентов направления 15.03.01 «Машиностроение» очной и заочной форм обучения.

Методические указания подготовлены в электронном виде в файле МУ ССУБДвАП.pdf.

Ил. 12. Табл. 2. Библиогр.: 12 назв.

УДК 658.52:681.3:681.5(07)
ББК 32.97:32.8я7

Рецензент - А. В. Демидов, канд. техн. наук, доцент кафедры автоматизированного оборудования машиностроительного производства ВГТУ

Издается по решению редакционно-издательского совета Воронежского государственного технического университета

ВВЕДЕНИЕ

Основной целью выполнения лабораторных работ по курсу "Современные системы управления базами данных в автоматизированном производстве является практическое ознакомление с приемами работы с базами данных, методиками обработки реляционных баз данных.

Тематика и содержание работ составлены с учетом материальной базы лаборатории кафедры. Занятия в лаборатории проводятся под руководством преподавателя и лаборанта. Для проведения лабораторных занятий группа делится на подгруппы (по 10-12 человек), постоянный состав которых сохраняется до окончания всего лабораторного практикума. Лабораторные работы выполняются самостоятельно, необходимые записи ведутся в рабочих тетрадях. По результатам выполненных работ студент обязан:

1. Знать целевое назначение работы, уметь объяснить порядок и технику выполнения.

2. Знать устройство, приёмы управления и настройку оборудования и приборов, применяемых в работе.

3. Понимать физический и практический смысл полученных данных.

4. Предъявить отчет с необходимыми расчетами, эскизами, графиками и выводами по каждой работе.

Перед началом лабораторных работ студенты знакомятся с содержанием лабораторного практикума, организацией и режимом занятий, правилами техники безопасности.

По окончании работы рабочее место, оборудование, аппаратура и инструменты сдаются лаборанту.

Отчет по работе оформляется на писчей бумаге формата А4, графики и схемы при необходимости на миллиметровой бумаге или кальке. Отчёт брошюруется в общую тетрадь, записи в нем выполняются рукописно, рисунки, графики и схемы выполняется в соответствии с ЕСКД. Отчёт выполняется студентом индивидуально с индивидуальными выводами по работе. Отчет оформляется следующим образом:

1.Титульный лист, на котором указаны название вуза, кафедры дисциплины, группа и фамилия студента.

2. Назначение работы.

3. Цель работы.

4. Применяемое оборудование, приборы, датчики.

5. Последовательность и описание проводимых работ.

6. Результаты работы с таблицами и графиками.

7. Анализ результатов и выводов.

ОРГАНИЗАЦИЯ ЛАБОРАТОРНЫХ ЗАНЯТИЙ

Занятия в лаборатории проводятся под руководством преподавателя. Для проведения лабораторных занятий группа делится на подгруппы (по 10 - 12 человек), постоянный состав которых сохраняется до окончания всего лабораторного практикума. Лабораторные работы выполняется студентами самостоятельно. По результатам выполненных работ оформляется отчет. По окончании лабораторного практикума каждый студент должен сдать зачёт.

При сдаче зачёта студент обязан:

1. Знать целевое назначение работы и уметь объяснить порядок и технику её выполнения.
2. Знать устройство, приемы управления и настройку оборудования, приборов и программных средств, применяемых в работе.
3. Понимать физический и практический смысл полученных результатов.
4. Предъявить отчёт с записями со всеми необходимыми расчётами, эскизами, графиками и выводами по каждой выполненной работе.

ПОРЯДОК ВЫПОЛНЕНИЯ ЛАБОРАТОРНЫХ РАБОТ

Перед началом лабораторных занятий студенты знакомятся с содержанием лабораторного практикума, организацией и режимом занятий, правилами техники безопасности. Распределение обязанностей внутри подгруппы производится студентами с соблюдением принципа равного участия в работе каждого студента.

Студенты должны:

1. Изучить самостоятельно методику выполнения работы и ознакомиться с организацией рабочего места.
2. Ознакомиться под руководством преподавателя или лаборанта с устройством лабораторного оборудования и его управлением.
3. Категорически запрещается самостоятельный пуск оборудования и пользование без ведома преподавателя или лаборанта.
4. Изучить правила техники безопасности.
5. Произвести под руководством преподавателя или лаборанта настройку оборудования и приборов.
6. Выполнить самостоятельно необходимые учебные задания в соответствии с методикой. Результаты занести в рабочую тетрадь.
7. После окончания работы рабочее место сдать лаборанту.
8. Провести анализ полученных результатов и сделать выводы по работе. Оформить и сдать преподавателю отчет.

ТРЕБОВАНИЯ К ОТЧЕТУ

Отчет по работе оформляется на бумаге стандартного формата (формат А4). Отчет брошюруется в общую тетрадь. Отчет представляется в печатном виде. Коллективное составление и сдача отчетов не допускается.

Отчет по лабораторной работе должен быть выполнен в текстовом редакторе Microsoft Word 2003 или выше и содержать: титульный лист, название темы работы, цели работы, перечень технических и программных средств, необходимых для выполнения лабораторной работы; краткое описание исследуемого вопроса; алгоритм программы; исходные данные варианта; распечатку полученных в ходе расчета значений; выводы, содержащие анализ проведенной работы.

В выводах дается краткое объяснение сущности полученных результатов. Выводы должны быть краткими и отвечать на вопросы, поставленные в лабораторной работе.

ТЕХНИКА БЕЗОПАСНОСТИ ПРИ РАБОТЕ СТУДЕНТОВ В ЛАБОРАТОРИИ

Для того чтобы уберечь себя и товарищей от несчастного случая, а государственное имущество от аварии, необходимо хорошо знать и полностью выполнять правила внутреннего распорядка, техники безопасности и пожарной безопасности. К лабораторным работам допускаются студенты, которые ознакомились с общими конкретными требованиями техники безопасности и прошли соответствующий инструктаж.

Проведение инструктажа и проверка знаний правил техники безопасности должны быть зарегистрированы соответствующими записями в лабораторном журнале. Конкретные требования техники безопасности при проведении той или иной работы изложены в описании к лабораторным работам.

Лабораторная работа №1

Построение реляционной базы данных средствами Microsoft Excel

(4 часа)

Цель работы: ознакомиться с принципами создания реляционных баз данных при помощи ЭВМ и специализированного программного обеспечения.

Технические средства и программное обеспечение:

1. IBM-PC или совместимый компьютер;
2. Операционная система Microsoft Windows;
3. Пакет офисных программ Microsoft Office;
4. База данных «Станки»;
5. Microsoft Visual Basic.NET.

Теоретические сведения:

База данных - набор сведений, хранящихся некоторым упорядоченным способом. Можно сравнить базу данных со шкафом, в котором хранятся документы. Иными словами, база данных - это хранилище данных. Сами по себе базы данных не представляли бы интереса, если бы не было систем управления базами данных (СУБД).

Система управления базами данных - это совокупность языковых и программных средств, которая осуществляет доступ к данным, позволяет их создавать, менять и удалять, обеспечивает безопасность данных и т.д. В общем СУБД - это система, позволяющая создавать базы данных и манипулировать сведениями из них.

Одними из самых простых баз данных являются ***реляционные***. Все данные представляются в виде простых таблиц, разбитых на строки и столбцы, на пересечении которых расположены данные.

Самым простым способом создания таблиц является использование табличных процессоров, таких, как Microsoft Excel (рис.1).

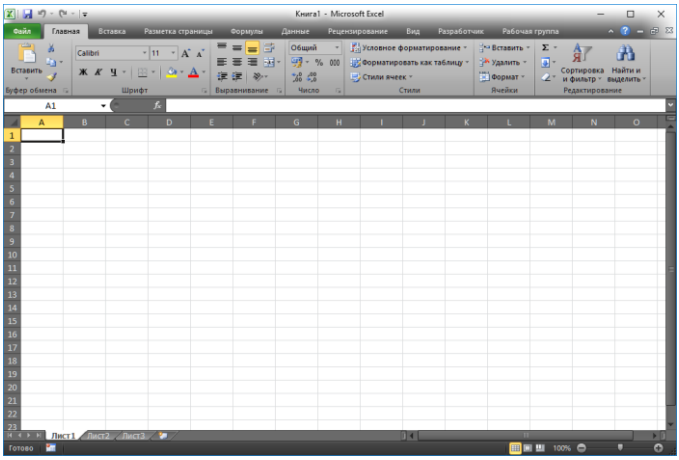

Рис. 1. Табличный процессор Microsoft Excel

Работать с такими программами достаточно просто, однако основную сложность при этом представляет организация таблицы, которая будет содержать необходимые данные.

Так, если рассматривать металлообрабатывающее оборудование, в частности, металлорежущие станки, то можно предложить следующую структуру такой таблицы (рис. 2).

1 Группа станка включает в себя классификацию станков в соответствии с их технологическим назначением, т.е. токарный, фрезерный, сверлильный, шлифовальный, карусельный, расточной и т.д.

2 Модель станка в общем виде представляет собой текстовую строку, содержащую обозначение модели станка.

3 Назначение представляет собой текстовую строку, содержащую технологическое назначение станка.

4 Техническая характеристика содержит необходимые для выбора станка конструктивные и технологические параметры.

1 Группа станка	2 Модель станка	3 Назначение
4 Техническая характеристика	5 Общий вид	

Рис. 2. Структура таблицы

5 Общий вид содержит фотографию общего вида или чертеж общего вида металлообрабатывающего станка.

Каждую станочную группу лучше всего разместить на отдельном рабочем листе, что бы в случае добавления данных о новом оборудовании было легче управлять данными. При этом достаточно будет просто программными средствами дописать сточку с данными ниже всех существующих.

Структуру базы данных рассмотрим на примере базы «Станки». Интерфейс показан на рис.3.

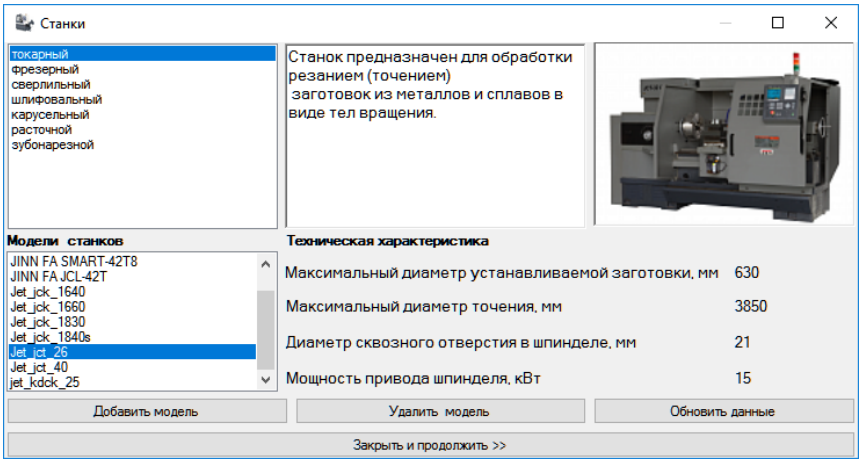

Рис.3. Интерфейс базы данных «Станки»

Структура базы данных в рассматриваемой системе выглядит следующим образом (рис. 4).

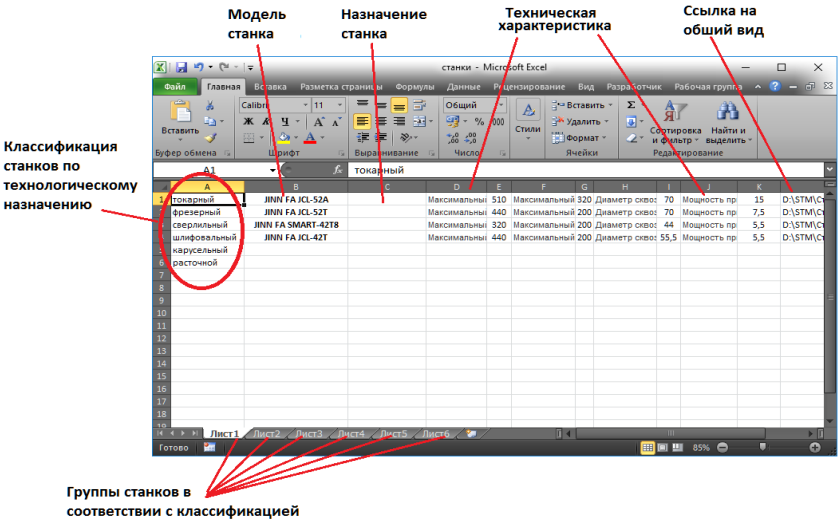

Рис. 4. Структура базы данных металлообрабатывающих станков

Т.е. база данных представляет собой обычную рабочую книгу Microsoft Excel, состоящую из нескольких рабочих листов, количество которых соответствует классификации металлообрабатывающего оборудования.

Баха данных содержит информацию о современных моделях станков с системой управления ЧПУ.

Задание:

1. Предложить структуру базы данных, содержащую данные о металлорежущих станках, изготавливаемых на них деталях и средств автоматизации.

Лабораторная работа №2

Основы работы в Microsoft Visual Basic.NET

(4 часа)

Цель работы: ознакомиться с основами программирования в среде Microsoft VisualBasic.NET, создать простое Windows-приложение.

Технические средства и программное обеспечение:

1. IBM-PC или совместимый компьютер;
2. Операционная система Microsoft Windows;
3. Пакет офисных программ Microsoft Office;
4. САПР ТП ОМД «Триумф»;
5. Microsoft Visual Basic.NET.

Теоретические сведения:

Для управления данными можно использовать системы управления базами данных (ССУБД), построенные на базе специализированных программ, таких как Microsoft Access, или созданных самостоятельно при помощи современных языков визуального программирования, таких, как Microsoft Visual Basic.NET (рис.5).

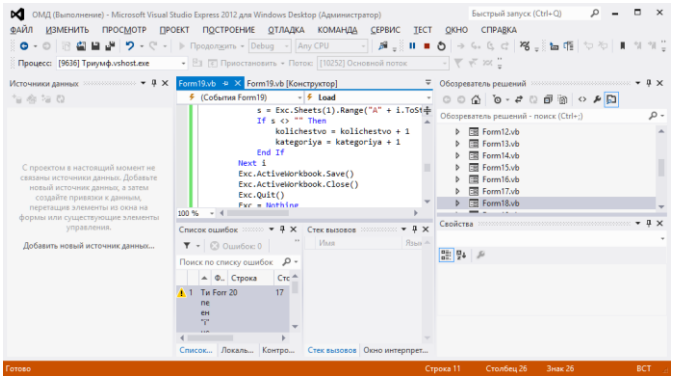

Рис. 5. Интерфейс Microsoft Visual Basic.NET

Применение современных средств программирования предполагает знания основ работы с языковыми средствами, а так же методов и средств объектно-ориентированного программирования.

Visual Basic .NET (VB.NET) — объектно-ориентированный язык программирования, который можно рассматривать как очередной виток эволюции Visual Basic (VB), реализованный на платформе Microsoft .NET.

Основам работы с Visual Basic.NET посвящено множество интернет-ресурсов, наиболее подробным из которых является сервер разработчика –

https://docs.microsoft.com/ru-ru/dotnet/welcome

Microsoft Visual Basic.NET работает со следующими структурами:

- Классы – ***class***;
- Переменные – ***dim***. Переменные могут быть следующих основных типов:
  - ***single***;
  - ***integer***;
  - ***decimal***;
- Константы – ***const***;
- Процедуры – ***sub***;
- Функции – ***function***.

Базовые команды, осуществляющие управление программными данными, в целом аналогичны большинству языков программирования, но в отличие от них имеют более простой синтаксис.

- **Abs (функция)** - возвращает абсолютное значение числа
- **And (операция)** - логическое и
- **AppActivate (оператор)** - активизирует окно приложения
- **Array (функция)** - создает массив из параметров и возвращает его какпеременную типа Variant
- **Asc (функция)** - возвращает числовой код первого символа строки аргумента
- **Atn (функция)** - возвращает арктангенс числа в радианах
- **Beep (оператор)** - проигрывает звуковой сигнал через динамик компьютера
- **Call (оператор)** - передает управление процедуре модуля (Sub), функциимодуля (Function) или подпрограмме DLL
- **CBool (функция)** - приводит выражение к типу Boolean
- **CByte (функция)** - преобразует выражение к типу Byte
- **CCur (функция)** - преобразование выражения к типу Currency
- **CDate (функция)** - преобразование выражения к типу Date
- **CDbl (функция)** - преобразование к типу Double
- **ChDir (оператор)** - изменяет текущий каталог на устройстве
- **ChDrive (оператор)** - изменяет текущее устройство
- **Choose (функция)** - возвращает значение из списка аргументов с определеннымпорядковым номером

- **Chr (функция)** - возвращает символ, связанный с определенным числовымкодом
- **CInt (функция)** - преобразование выражения к типу Integer
- **CLng (функция)** - преобразование выражения к типу Long
- **Close (оператор)** - закрывает файл, открытый оператором Open
- **Command (функция)** - возвращает командную строку, используемую длязапуска Visual Basic или приложения на Visual Basic
- **Const (оператор)** - объявления констант
- **Cos (функция)** - возвращает косинус числа
- **Create Object (функция)** - создать OLE Automation объект
- **CSng (функция)** - преобразование выражения к типу Single
- **CStr (функция)** - преобразование выражения к типу String
- **CurDir (функция)** - возвращает текущий каталог логического устройства
- **CVar (функция)** - преобразование выражения к типу Variant
- **CVErr (функция)** - возвращает подтип ошибки, для определенного пользователемномера ошибки
- **Date (оператор)** - устанавливает значение системной даты
- **Date (функция)** - возвращает значение системной даты
- **DateAdd (функция)** - возвращает переменную типа Variant, содержащуюдату, отличающуюся от заданной на определенный интервал времени
- **DateDiff (функция)** - возвращает число временных интервалов между двумядатами
- **DatePart (функция)** - возвращает определенную часть заданной даты
- **DateSerial (функция)** - возвращает дату для заданного года, месяцаи дня
- **DateValue (функция)** - возвращает дату
- **Day (функция)** - возвращает число от 1 до 31, соответствующее текущемудню месяца
- **DDB (функция)** - возвращает значение амортизационных потерь за определенныйпериод

- **Declare (оператор)** - на уровне модуля объявляет ссылки ко внешним подпрограммамв DLL
- **Deftype (операторы)** - устанавливает тип данных по умолчанию на уровнемодуля для переменных, параметров подпрограмм, а также возвращаемыхзначений для функций и операторов Property Get, начинающихся с определенныхсимволов
- **Dim (оператор)** - объявляет переменные и выделяет память под них
- **Dir (функция)** - возвращает имя файла или каталог, подходящий дляданного шаблона или атрибута файла, или метку тома устройства
- **DoEvents (функция)** - прерывает выполнение приложения
- **Do... Loop (оператор)** - повторяет блок команд до тех пор, пока условиеверно или до тех пор, пока условие не станет верным
- **End (оператор)** - заканчивает подпрограмму или блок команд
- **Environ (функция)** - возвращает строку, связанную с переменной окруженияоперационной системы
- **EOF (функция)** - возвращает значение, указывающее, достигнут ликонец файла
- **Eqv (оператор)** - проверяет логическое равенство двух выражений
- **Erase (оператор)** - повторно инициализирует элементы массивов фиксированногоразмера и перераспределяет память под динамические массивы
- **Error (оператор)** - эмулирует возникновение ошибки
- **Error (функция)** - возвращает текст сообщения данного номера ошибки
- **Exit (операторы)** - осуществляет выход из циклов Do ... Loop, For... Next, функции и процедур
- **Exp (функция)** - возвращает экспоненту числа
- **FileAttr (функция)** - возвращает режим открытия или номер (handle) файла
- **FileCopy(оператор)** - копирует файл
- **FileDateTime (функция)** - возвращает дату и время создания или последнеймодификации файла

- **FileLen (функция)** - возвращает длину файла в байтах
- **Fix (функция)** - возвращает целую часть числа
- **For Each...Next (оператор)** - повторяет одну и ту же последовательностькоманд для каждого элемента массива или коллекции
- **For...Next (оператор)** - повторяет последовательность команд определенноечисло раз
- **Format (функция)** - форматирует выражение в соответствии с заданнымформатом
- **FreeFile (функция)** - возвращает следующий не занятый номер файла дляиспользования в операторе Open
- **Function (оператор)** - объявляет имя, аргументы и код подпрограммы,возвращающей значение (функции)
- **FV (функция)** - возвращает значение ренты, основываясь на периодическихвзносах и постоянной норме капиталовложений
- **Get (оператор)** - читает данные из открытого файла в переменную
- **GetAttr (функция)** - возвращает атрибуты файла, каталога или меткитома
- **GetObject (функция)** - возвращает OLE Automation объект для файла сданным расширением
- **GoSub... Return (оператор)** - выполняет подпрограмму
- **GoTo (оператор)** - передает управление определенной строке подпрограммыбез возврата контроля
- **Hex (функция)** - возвращает строку, представляющую шестнадцатеричноезначение числа
- **Hour (функция)** - возвращает целое число в диапазоне 0 - 23 включительно,представляющее определенный час дня
- **If...Then... Else (оператор)** - выполнение групп команд в зависимостиот значения выражения
- **Iff (функция)** - возвращает одно из двух значении в зависимости от значениявыражения
- **Imp (операция)** - импликация двух выражений
- **Input (функция)** - возвращает символы из файла, открытого для последовательногодоступа или как двоичный файл

- **Input # (оператор)** - считывает данные из открытого файла в переменные
- **InputBox (функция)** - показывает диалоговое окно ввода, ожидает вводатекста и возвращает содержимое введенного текста, после закрытия окна
- **InStr (функция)** - возвращает позицию первой найденной подстроки в строке
- **Int (функция)** - возвращает целую часть числа
- **Is (операция)** - сравнение двух ссылок на объекты
- **IsArray (функция)** - возвращает булево значение, указывающее, являетсяли данная переменная массивом
- **IsDate (функция)** - возвращает булево значение, указывающее, можетли выражение быть преобразовано к типу Date
- **IsEmpty (функция)** - возвращает булево значение, указывающее, инициализированоли значение данной переменной
- **IsError (функция)** - возвращает булево значение, указывающее, являетсяли выражение значением кода ошибки
- **IsMissing (функция)** - возвращает булево значение, указывающее,был ли передан данный необязательный параметр в подпрограмму
- **IsNull (функция)** - возвращает булево значение, указывающее, несодержит ли выражение недопустимое (Null) значение
- **IsNumeric (функция)** - возвращает булево значение, указывающее,может ли данное выражение рассматриваться как число
- **IsObject (функция)** - возвращает булево значение, указывающее, являетсяли выражение объектом OLE Automation
- **Kill (оператор)** - удаляет файл
- **LBound (функция)** - возвращает значение нижней границы индекса массива
- **LCase (функция)** - возвращает строку в нижнем регистре
- **Left (функция)** - возвращает определенное число символов с начала строки
- **Len (функция)** - возвращает число символов строки или число байт,необходимых для хранения переменной

- **Let (оператор)** - присваивает значение выражения переменной или свойству
- **Like (операция)** - сравнение двух строк
- **Line Input # (оператор)** - считывает строку из файла в переменную
- **Load (оператор)** - загружает в память форму или элемент управления
- **LoadPicture (функция)** - загружает графический образ в объекты: Form,
- **Loc (функция)** - возвращает текущую позицию чтения/записи в открытомфайле
- **Lock (оператор)** - контролирует доступ других процессов ко всему или части открытого файла
- **LOF (функция)** - возвращает размер в байтах открытого файла
- **Log (функция)** - возвращает натуральный логарифм числа
- **LSet (оператор)** - копирует строку в строковую переменную, а такжекопирует значение переменной одного специализированного типа в переменнуюдругого специализированного типа
- **LTrim (функция)** - возвращает копию строки без лидирующих пробелов
- **Mid (оператор)** - замещает определенное число символов в строке на символыиз другой строки
- **Mid (функция)** - возвращает определенное число символов с определеннойпозиции строки
- **Minute (функция)** - возвращает целое число в диапазоне 0 - 59, представляющееминуту часа
- **MkDir (оператор)** - создает новый каталог
- **Mod (операция)** - возвращает остаток от деления двух чисел
- **Month (функция)** - возвращает целое число в диапазоне 1 - 12, представляющееномер месяца
- **MsgBox (функция)** - показывает сообщение в диалоговом окне, ожидаетвыбор одной из кнопок пользователем и возвращает значение, указывающее,какая кнопка была выбрана
- **Name (оператор)** - переименовывает файл или каталог
- **Not (операция)** - логическое отрицание

- **Now (функция)** - возвращает текущие значения даты и времени
- **Oct (функция)** - возвращает строку, представляющую восьмеричное представлениечисла
- **On Error (оператор)** - устанавливает обработчик ошибок и задает местоположениеподпрограммы обработки; используется также для отмены обработки ошибокподпрограммой обработчика
- **On..GoSub, On...GoTo (операторы)** - передача управления на одну изнескольких определенных строк (меток), в зависимости от значения выражения
- **Open (оператор)** - скрывает файл для ввода/вывода
- **Option Base (оператор)** - используется для объявления значения нижнейграницы размерности индексов массивов по умолчанию
- **Option Compare (оператор)** - используется на уровне модуля для объявленияметода сравнения по умолчанию при сравнении строк
- **Option Explicit (оператор)** - используется на уровне модуля для установкипроверки наличия объявлений для всех переменных в данном модуле
- **Option Private (оператор)** - используется на уровне модуля для указания,что весь модуль является Private
- **Or (операция)** - логическое ИЛИ
- **Partition (функция)** - возвращает строку, указывающую, сколько развстретились числа из заданного диапазона
- **Print # (оператор)** - записывает форматированные данные в файл
- **Private (оператор)** - используется на уровне модуля для объявления Privateпеременных и выделяет место в памяти для их хранения
- **Property Get (оператор)** - объявляет имя, аргументы и код подпрограммыполучения значения свойства
- **Property Let (оператор)** - объявляет имя, аргументы и код процедурыустановки значения свойства
- **Property Set (оператор)** - объявляет имя, аргументы и код процедурыустановки ссылки на объект
- **Public (оператор)** - используется на уровне модуля для объявления Publicпеременных и выделяет место в памяти для их хранения
- **Put (оператор)** - записывает переменную в файл

- **QBColor (функция)** - возвращает RGB код, соответствующий номеру цвета
- **Randomize (оператор)** - инициализирует генератор случайных чисел
- **RGB (функция)** - возвращает целое число, представляющее значение RGBкода
- **ReDim (оператор)** - используется на уровне подпрограммы для переопределенияразмера динамических массивов и выделения под них места в памяти
- **Rem (оператор)** - вставка комментариев в программу
- **Reset (оператор)** - закрывает все открытые программой файлы
- **Resume (оператор)** - продолжает выполнение программы после завершенияпроцедуры обработчика ошибок
- **Right (функция)** - возвращает определенное число символов с правой стороныстроки
- **RmDir (оператор)** - удаляет каталог
- **Rnd (функция)** - возвращает случайное число
- **RSet (оператор)** - копирует правую часть строки в строковую переменную
- **RTrim (функция)** - возвращает копию строки без конечных пробелов
- **SavePicture (оператор)** - сохраняет в файл графический образ объектаForm, элементов управления Picture Box или Image
- **Second (функция)** - возвращает целое значение в диапазоне 0 - 59,представляющее секунду в минуте
- **Seek (оператор)** - устанавливает позицию для следующей операции чтенияили записи в открытый файл
- **Seek (функция)** - возвращает текущую позицию чтения/записи открытогофайла
- **Select Case (оператор)** - выполняет одну или несколько команд, в зависимостиот значения выражения
- **SendKeys (оператор)** - посылает одно или несколько нажатий клавиш активномуокну, как если бы они были введены пользователем с клавиатуры

- **Set (оператор)** - связывает ссылку на объект с переменной или свойством
- **SetAttr (оператор)** - устанавливает атрибуты файла
- **Sgn (функция)** - возвращает знак числа
- **Shell (функция)** - запускает внешнюю программу на выполнение
- **Sin (функция)** - возвращает значение синуса угла
- **Space (функция)** - возвращает строку, содержащую определенное числопробелов
- **Spc (функция)** - позиционирование в строке вывода
- **Sqr (функция)** - подсчет значения квадратного корня числа
- **Static (оператор)** - используется на уровне модуля для объявления переменныхи выделяет место в памяти для их хранения. Переменные сохраняют значениядо завершения программы
- **Stop (оператор)** - приостанавливает выполнение программы
- **Str (функция)** - возвращает строковое представление числа
- **StrComp (функция)** - возвращает результат сравнения строк
- **StrConv (функция)** - возвращает преобразованную строку
- **String (функция)** - возвращает строку заданной длины из одинаковых символов
- **Sub (оператор)** - объявляет имя, параметры и тело процедуры
- **Switch (функция)** - подсчитывает значения списка выражении и возвращаетзначение или выражение, связанное с выражением из списка, значениекоторого равно True
- **Tab (функция)** - позиционирование в строке вывода
- **Tan (функция)** - возвращает значение тангенса угла
- **Time (оператор)** - устанавливает значение системных часов
- **Time (функция)** - возвращает значение типа Date, указывающее текущеесистемное время
- **Timer (функция)** - возвращает число секунд, прошедших после полуночи
- **TimeSerial (функция)** - возвращает значение типа Date, содержащеевремя для заданного часа, минуты и секунды
- **Time Value (функция)** - возвращает значение типа Date, содержащеевремя суток

- **Trim (функция)** - возвращает копию строки без начальных и конечных пробелов
- **Type (оператор)** - объявляет на уровне модуля специализированный типданных
- **TypeName (функция)** - возвращает строку информации о заданной переменной
- **UBound (функция)** - возвращает значение наибольшего индекса для даннойразмерности массива
- **UCase (функция)** - возвращает строку, преобразованную в верхний регистр
- **Unload (оператор)** - выгружает форму или элемент управления из памяти
- **Unlock (оператор)** - контролирует доступ других процессов ко всему иличасти открытого файла
- **Val (функция)** - возвращает числовое представление строки
- **VarType (функция)** - возвращает значение, указывающее тип переменной
- **Weekday (функция)** - возвращает целое число, представляющее день недели
- **While...Wend (оператор)** - выполняет в цикле последовательность команддо тех пор, пока верно условие
- **Width # (оператор)** - назначает ширину строки вывода для операции записив открытый файл
- **With (оператор)** - выполняет последовательность команд для конкретногообъекта или переменной специализированного типа
- **Write # (оператор)** - записывает данные в файл
- **Xor (операция)** - исключающее ИЛИ
- **Year (функция)** - возвращает целое число, представляющее год

Задание

1. Освоить базовые приемы работы со средствами Microsoft Visual Basic.NET.

2. Составить простое приложение средствами языка Microsoft Visual Basic.NET с применением формы (табл. 1).

Таблица 1

№	Задание	№	Задание	№	Задание
1	сложение двух чисел	13	ввод и сохранение текста	25	ввод и сохранение текста
2	вычитание двух чисел	14	деление двух чисел	26	сложение двух чисел
3	умножение двух чисел	15	сложение двух чисел	27	вычитание двух чисел
4	деление двух чисел	16	умножение двух чисел	28	умножение двух чисел
5	ввод и сохранение текста	17	ввод и сохранение текста	29	сложение двух чисел
6	сложение двух чисел	18	деление двух чисел	30	умножение двух чисел
7	вычитание двух чисел	19	ввод и сохранение текста		
8	умножение двух чисел	20	сложение двух чисел		
9	сложение двух чисел	21	вычитание двух чисел		
10	умножение двух чисел	22	умножение двух чисел		
11	вычитание двух чисел	23	деление двух чисел		
12	сложение двух чисел	24	ввод и сохранение текста		

Лабораторная работа №3

Создание программного обеспечения управления данными реляционной базы данных

(4 часа)

Цель работы: ознакомиться с методикой создания Windows-приложений средствами Microsoft VisualBasic.NET.

Технические средства и программное обеспечение:

1. IBM-PC или совместимый компьютер;
2. Операционная система Microsoft Windows;
3. Пакет офисных программ Microsoft Office;
4. САПР ТП ОМД «Триумф»;
5. Microsoft Visual Basic.NET.

Теоретические сведения:

Данные, хранящиеся в таблице, необходимо отобразить на форме приложения, создаваемого средствами Microsoft Visual Basic.NET. В качестве примера рассмотрим следующую структуру (рис.6).

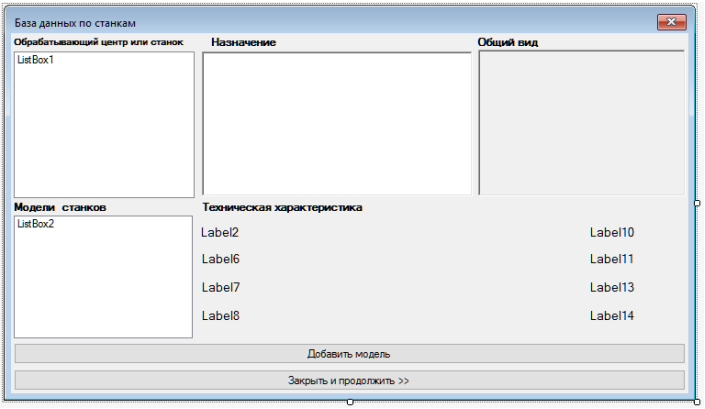

Рис. 6. Форма управления реляционной базой данных

Программная форма содержит следующие объекты:

- ***Listbox*** – представляет собой элемент управления, позволяющий пользователю выбрать единственное значение из множества, представленного в виде списка;

- ***RichTextbox*** – элемент управления, имеющий возможность отображения текста с элементами форматирования;
- ***Picturebox*** - элемент управления, позволяющий отображать изображение в формате jpg, png и т.д. с возможностями настройки вида отображения;
- ***Label*** – объект, позволяющий отобразить любую текстовую информацию (как числа, так и текст) .

Все объекты предназначенные для вывода определенных типов данных. Пример работы системы управления базы данных показана на рис. 7.

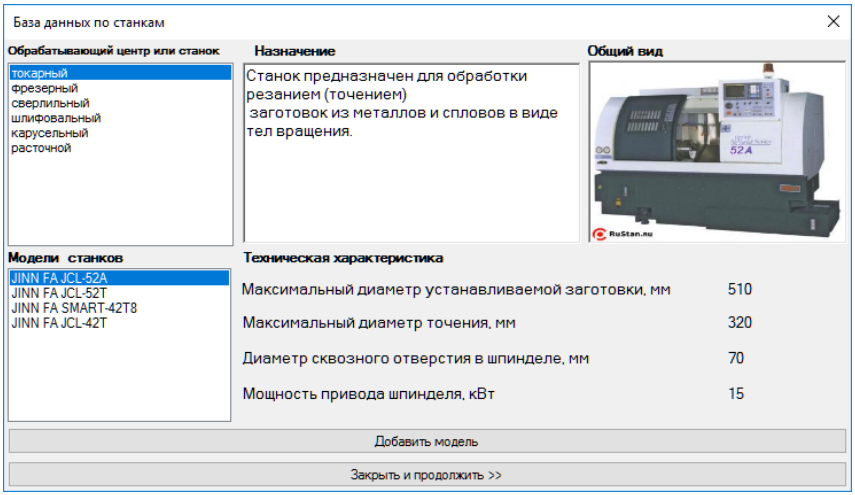

Рис. 7. Управление базами данных

Объекты при создании формы выбираются из «Панели элементов» (рис. 8).

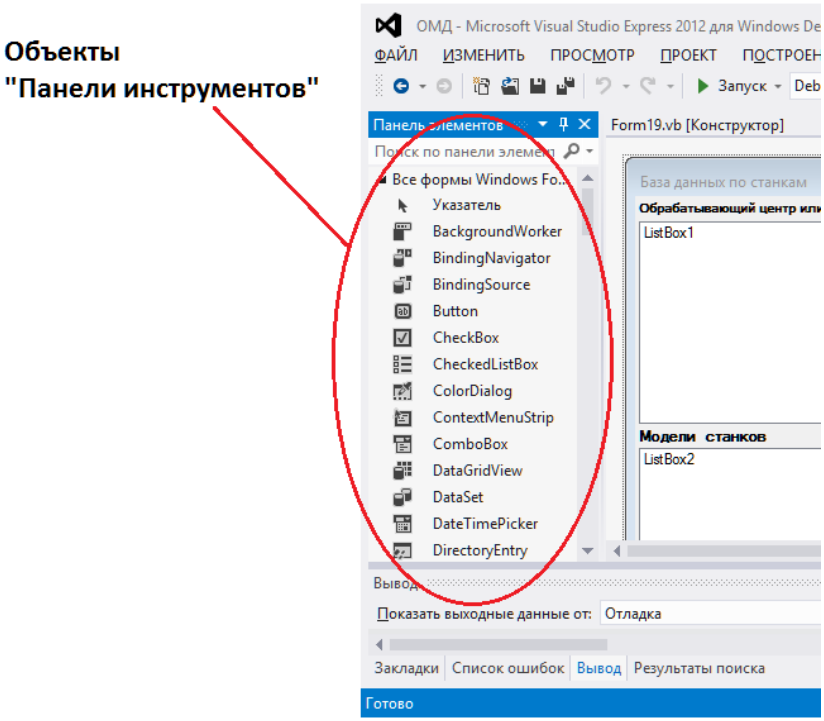

Рис. 8. Объекты «Панели инструментов» Microsoft Visual Basic.NET

И размещаются на форме, где создается приложение, в необходимом порядке.

Как заставить отображать данные выбранных элементов, рассмотрим в последующих работах.

Задание

1. Создать интерфейс системы управления базой данных.
2. Тему для создания интерфейса выбрать из табл. 2.

Таблица 2

№	Задание	№	Задание	№	Задание
1	База данных станков	11	База данных деталей	21	База данных технологических инструментов
2	База данных приспособлений	12	База данных приспособлений	22	База данных станков
3	База данных деталей	13	База данных станков	23	База данных приспособлений
4	База данных приспособлений	14	База данных технологических инструментов	24	База данных станков
5	База данных средств автоматизации	15	База данных средств автоматизации	25	База данных технологических инструментов
6	База данных технологических инструментов	16	База данных деталей	26	База данных деталей
7	База данных средств автоматизации	17	База данных станков	27	База данных приспособлений
8	База данных технологических инструментов	18	База данных приспособлений	28	База данных станков
9	База данных деталей	19	База данных технологических инструментов	29	База данных средств автоматизации
10	База данных технологических инструментов	20	База данных станков	30	База данных приспособлений

Лабораторная работа №4

Создание инструментов чтения данных
реляционной базы данных

(4 часа)

Цель работы: создать инструменты чтения данных средствами Microsoft VisualBasic.NET.

Технические средства и программное обеспечение:

1. IBM-PC или совместимый компьютер;
2. Операционная система Microsoft Windows;
3. Пакет офисных программ Microsoft Office;
4. САПР ТП ОМД «Триумф»;
5. Microsoft Visual Basic.NET.

Теоретические сведения:

Для отображения станочных групп предусмотрен список Listbox1, отображение элементов в котором осуществляется при загрузке формы в процедуре Form1_Load. Рассмотрим пример заполнения списка значениями из реляционной базы данных.

```
Private Sub Form18_Load(sender As Object, e As EventArgs)
Handles MyBase.Load
   Exc1 = CreateObject("Excel.Application")
Exc1.Workbooks.Open("D:\STM\Станки\станки.xlsx").Activate()
   For i = 1 To 50
       s = Exc1.Sheets(1).Range("A" + i.ToString).Value()
        If s <> "" Then
           ListBox1.Items.Add(s)
        End If
   Next i
   Exc1.ActiveWorkbook.Save()
   Exc1.ActiveWorkbook.Close()
        Exc1.Quit()
        Exc1 = Nothing
        Label2.Text = " "
        Label6.Text = " "
        Label7.Text = " "
        Label8.Text = " "
        Label10.Text = " "
        Label11.Text = " "
        Label13.Text = " "
        Label14.Text = " "
    End Sub
```

Листинг 1

Данная процедура позволяет отобразить структуру данных в следующем виде (рис. 9).

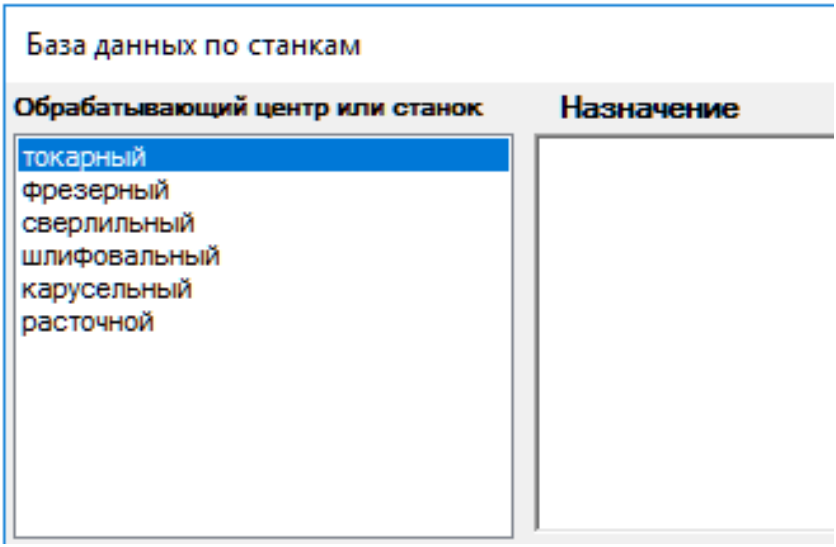

Рис. 9. Структура данных

Выбор объекта в списке осуществляет команда

отображение = ***Listbox1.SelectedIndex*** = присвоение

При этом любой элемент управления позволяет ввести или отобразить значение переменной.

Отображение технической характеристики осуществляется с помощью инструмента Label, который имеет свойство Text. Листинг примера заполнения данных показан ниже (Листинг 2).

```
        stanok = ListBox2.SelectedIndex + 1
         Dim i As Integer

         Exc = CreateObject("Excel.Application")
Exc.Workbooks.Open("D:\STM\Станки\станки.xlsx").Activate()

         For i = 1 To 50
             s = Exc.Sheets(kategoriya).Range("B" +
             i.ToString).Value()
             If s <> "" Then
                 s6(i) = Exc.Sheets(kategoriya).Range("L" +
                 i.ToString).Value()
```

```
            s2(i) = Exc.Sheets(kategoriya).Range("E" +
            i.ToString).Value()
            s3(i) = Exc.Sheets(kategoriya).Range("G" +
            i.ToString).Value()
            s4(i) = Exc.Sheets(kategoriya).Range("I" +
            i.ToString).Value()
            s5(i) = Exc.Sheets(kategoriya).Range("K" +
            i.ToString).Value()
        End If
    Next i

    Exc.ActiveWorkbook.Save()
    Exc.ActiveWorkbook.Close()
    Exc.Quit()
    Exc = Nothing

    If (s6(stanok) <> "") Then
        PictureBox1.Load(s6(stanok))
    Else
        PictureBox1.Load("D:\STM\noimage.jpg")
    End If

    Label10.Text = s2(stanok)
    Label11.Text = s3(stanok)
    Label13.Text = s4(stanok)
    Label14.Text = s5(stanok)
```

Листинг 2

Работает данный код следующим образом (рис.10).

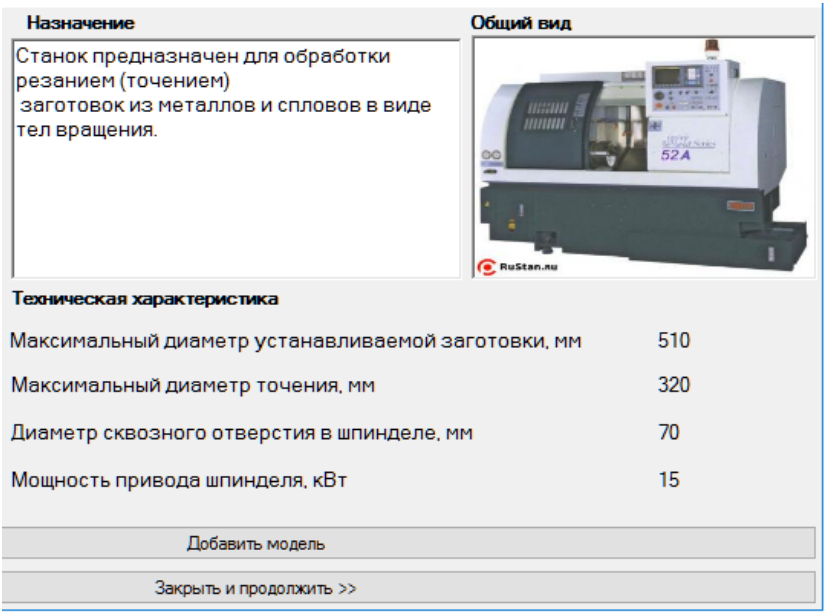

Рис. 10. Работа кода

Рассматриваемые методы позволяют получать данные из Microsoft Excel без подключения дополнительных библиотек, с применением объектов базового семейства ***System***.

Инструменты редактирования данных

Процедуры сохранения добавленных баз данных выполняется с помощью аналогичных рассмотренных в предыдущей работе методов, только вместо отображения данных ведется их запись.

Интерфейс добавления данных может выглядеть следующим образом (рис. 11).

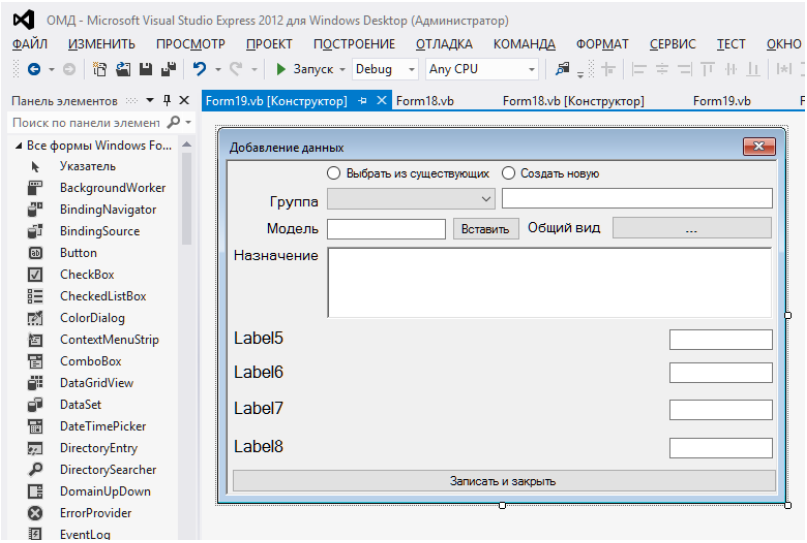

Рис. 11. Интерфейс диалога добавления данных

Пример листинга процедуры записи данных показан в листинге 3, а пример его работы на рис. 12.

```
        Exc = CreateObject("Excel.Application")

Exc.Workbooks.Open("D:\STM\Станки\станки.xlsx").Activate()
        Dim i As Integer

        For i = 1 To kolichestvo
            Exc.Sheets(i).Range("A" + (kolichestvo +
            1).ToString).Value() = TextBox1.Text
        Next

        Exc.ActiveWorkbook.Save()
        Exc.ActiveWorkbook.Close()
        Exc.Quit()
        Exc = Nothing
        Me.Close()
```

Листинг 3

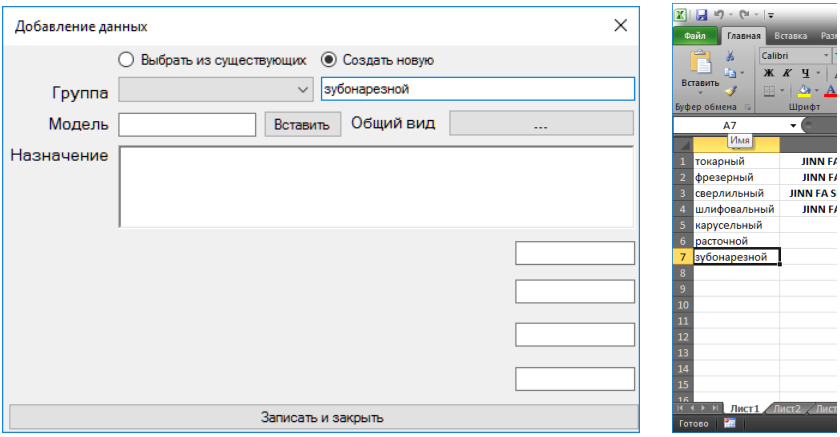

Рис. 12. Пример работы процедуры ввода данных

Задание

1. Создать процедуры чтения и отображения информации, содержащейся в базе данных.
2. Создать процедуры добавления нового оборудования в существующую базу данных.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Новокщенов С.Л. Современные системы управления базами данных в автоматизированном производстве [Электронный ресурс]. – Электрон. текстовые и граф. данные (1,9 Мб) / С.Л. Новокщенов, М.В. Кондратьев, В.И. Корнеев. – Воронеж: ФГБОУ ВО «Воронежский государственный технический университет», 2015. – 1 электрон. опт. диск (CD-ROM);
2. Тарасов С. В. СУБД для программиста. Базы данных изнутри – М.: СОЛОН-Пресс, 2021 – с., ил.
3. Келлехер Дж Наука о данных: Базовый курс / Джон Келлехер, Брендан Тирии; Пер. с англ. – М.; Альпина-Паблишер, 2020. – 222 с.

4. Нагао М., Катаяма Т., Узмура С. Структура и базы данных: Пер. японс. – М.: Мир, 1986. – 197 с., ил.
5. Карпова И. П. Базы данных. Курс лекций и материалы для практических заданий. – Учебное пособие. – М.: Питер, 2013. – 240 с.
6. Коннолли, Томас. Базы данных: проектирование, реализация и сопровождение. Теория и практика / Томас Коннолли, Каролин Бегг ; [перевод с английского Р. Г. Имамутдиновой, К. А. Птицына]. - 3-е изд. - Москва [и др.] : Вильямс, 2018. - 1439 с.
7. Норенков И.П. Информационная поддержка наукоемких изделий (CALS-технологии) / И.П. Норенков, П.К. Кузьмик. — М.: Изд-во МГТУ им. Н.Э. Баумана, 2002.
8. Колчин А.Ф.Управление жизненным циклом продукции / Колчин А.Ф., Овсянников М.В., Стрекалов А.Ф., Сумароков С.В. – М.: Анахарсис, 2002.
9. Имитационное моделирование: ученик для вузов / Ткаченоков С.В.; под ред. Ткаченоков С.В. – М. 2006 г.
10. Шабов И.К. Моделирование процессов / Шабов И.К. – 2000 г.
11. Российская энциклопедия CALS. Авиационно-космическое машиностроение / Под ред. А.Г. Братухина. – М.: ОАО НИЦ АСК, 2008.
12. Судов Е.В. Технологии интегрированной логистической поддержки изделий машиностроения / Судов Е.В., Левин А.И., Петров А.В., Чубарова Е.В. – М.: "Информбюро", 2006.

ОГЛАВЛЕНИЕ

СОВРЕМЕННЫЕ СИСТЕМЫ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ В АВТОМАТИЗИРОВАННОМ ПРОИЗВОДСТВЕ

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

к выполнению лабораторных работ
по направлению 15.03.01 «Машиностроение»
(профиль «Технологии, оборудование и автоматизация машиностроительных производств»)
очной и заочной форм обучения

Составитель
Новокщенов Сергей Леонидович

В авторской редакции

Подписано к изданию 29.11.2021.
Уч.-изд. л. 2,1

ФГБОУ ВО «Воронежский государственный технический университет»
394026 Воронеж, Московский просп., 14