

Министерство образования и науки Российской Федерации
Федеральное агентство по образованию
Московский авиационный институт
(Государственный технический университет)

Кафедра 311

ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ

Методические указания к лабораторным работам для студентов
всех форм обучения по специальности 080801

Москва 2020

Содержание

Введение	3
Организационно-методические указания	3
Лабораторная работа №1. Принципы создания подсистемы ручного ввода информации в информационных системах	5
ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ	5
КОНТРОЛЬНЫЕ ВОПРОСЫ	9
Лабораторная работа №2. Принципы создания подсистемы вывода информации в информационных системах	10
ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ	10
ЗАДАНИЕ НА РАБОТУ	12
КОНТРОЛЬНЫЕ ВОПРОСЫ	12
Лабораторная работа №3. Принципы открытого обмена данными между независимыми компонентами программного обеспечения информационных систем.	13
ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ	13
Лабораторная работа №4. Основы организации распределенной работы пользователей	20
ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ	20
ЗАДАНИЕ НА РАБОТУ	21
Лабораторная работа №5. Использование информационных ресурсов Internet в разрабатываемых ИС	23
Лабораторная работа №6. Совместная разработка ИС. Элементы документирования разработки	26
ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ	26
ЗАДАНИЕ НА РАБОТУ	28
КОНТРОЛЬНЫЕ ВОПРОСЫ	28
Лабораторная работа №7. Создание ИС на основе типовых решений. Изменение информационных объектов и баз данных	29
ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ	29
Лабораторная работа №8. Создание ИС на основе типовых решений. Добавление печатных форм	33
ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ	33
ЗАДАНИЕ НА РАБОТУ	40
КОНТРОЛЬНЫЕ ВОПРОСЫ	40
Лабораторная работа №9. Создание ИС на основе типовых решений. Добавление новых информационных объектов	42
ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ	42
ЗАДАНИЕ НА РАБОТУ	44

Введение

В методических указаниях приведен материал для выполнения лабораторных работ по дисциплине «Проектирование информационных систем».

Основной целью, определяющей содержание этих лабораторных работ, является практическое ознакомление с некоторыми наиболее важными особенностями, методами и решениями в области проектирования программного обеспечения информационных систем.

Для достижения этой цели в лабораторных работах рассматриваются различные темы по основам, методам и особенностям разработки программного обеспечения информационных систем. Также в лабораторных работах студентам предлагается изучить несколько специфических задач, которые часто встречаются при проектировании современного ПО ИС.

Темы лабораторных работ и задания подобраны таким образом, чтобы выработать, закрепить у студентов некоторые необходимые практические профессиональные навыки при создании ПО ИС. Особенное внимание уделено теме создания ПО ИС на основе типовых решений как наиболее распространенной в настоящее время практической задаче в области разработки, внедрения и сопровождения ПО ИС.

В связи с различным объемом учебных часов по данной дисциплине количество лабораторных работ у специальности 080801 может быть различным в зависимости от формы обучения.

Организационно-методические указания

1. Объем каждой лабораторной работы, подготовка и порядок выполнения построены таким образом, чтобы все студенты выполнили работу и сдали отчеты.

2. Студенты обязаны изучить технику безопасности при работе на лабораторных установках до 1000 В.
3. Каждая лабораторная работа выполняется студентами самостоятельно. Все студенты предъявляют индивидуальные отчеты.
4. Отчеты по лабораторной работе оформляются в электронном виде, в файле формата Microsoft Word. Отчет должен содержать следующие данные: фамилия, имя отчество студента, номер группы, номер лабораторной работы, номер варианта, название работы, цель работы, формулировка задания на работу согласно варианту, результаты выполнения задания в виде изображений с экрана с результатами выполнения задания с комментариями, схемы и исходные тексты программ. Файл с отчетом, сжатый в формате Zip или Rar, сдается преподавателю. Имя архивного файла отчета должно содержать фамилию студента, выполнившего работу и номер лабораторной работы на русском языке.
5. Готовясь к лабораторному занятию, студент обязан изучить необходимый теоретический материал, пользуясь настоящими указаниями и рекомендованной литературой, подготовить макет отчета и дать ответы на контрольные вопросы.
6. Неподготовленные студенты к выполнению лабораторной работы не допускаются.
7. Студенты, не сдавшие отчет во время занятия, сдают его в назначенное преподавателем время.
8. Студент, не выполнивший лабораторную работу, выполняет ее в согласованное с преподавателем время.
9. Проверка знаний студентов производится преподавателем во время лабораторного занятия и при сдаче отчета.
10. При сдаче отчета студент должен показать знание теоретического материала в объеме, определяемом контрольными вопросами.

Лабораторная работа №1. Принципы создания подсистемы ручного ввода информации в информационных системах

(1 вариант)

Цель работы: Изучить принципы создания интерфейсов ПО подсистемы ввода информации.

Содержание работы:

1. Изучить теоретические сведения.
2. Разработать приложение согласно заданию.
3. Ответить на контрольные вопросы.

ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Подсистема ввода информации используется в составе каждой информационной системы. Во многих случаях для ввода информации используются различные автоматизированные системы, однако в отдельных видах ИС ввод данных производится операторами. Характерным видом таких информационных систем являются системы электронного документооборота. В таких системах первичная информация заносится оператором ПЭВМ и сохраняется в памяти для последующей обработки.

Подсистема ввода данных с помощью оператора требует тщательного проектирования интерфейса. Работу оператора по вводу данных отличает высокая трудоемкость, необходимость концентрации внимания длительное время без перерывов, высокая ответственность. Монотонность и однообразие действий обуславливают привлечение на эту работу специалистов невысокой квалификации. Поэтому интерфейсы таких систем должны обуславливать минимум действий оператора, максимум помощи и дополнительной информации при вводе, автоматический контроль вводимых данных и высокую отказоустойчивость алгоритмов.

Ниже перечислены **требования к интерфейсам** ручного ввода данных:

- 1) Четко различимый шрифт отображения вводимой текстовой и цифровой информации.
- 2) Использование нейтральных или ярких цветов шрифтов фонов, рамок и других элементов оформления для выделения вводимой, справочной и контрольной информации.
- 3) Рациональное, связанное по смыслу и направлению взгляда размещение информации на экране.
- 4) Последовательное автоматическое перемещение между отображаемыми элементами вводимой информации.
- 5) Система предупреждений и оповещений о пропущенных данных, нарушении формата данных, некорректных операциях.
- 6) Развитая логическая система автоматического заполнения данных или заполнения значениями.
- 7) Развитая логическая система контроля правильности введенных данных.
- 8) Средства оперативного отображения дополнительной справочной информации о введенных данных.
- 9) Система управления, позволяющая максимально быстро и эффективно осуществлять переходы, исправления введенных данных, дальнейшую передачу или вывод на печать.
- 10) Наличие алгоритмов защиты от некорректных действий («защита от дурака»).
- 11) Повышенная отказоустойчивость системы в целом и в особенности от конфликтов обработки неполных данных или данных неверного формата.

ЗАДАНИЕ НА РАБОТУ

Разработать согласно требованиям и реализовать интерфейс автоматизированного рабочего места оператора по продажам для выполнения следующих действий:

1. Открыть базу клиентов

1. Ввод имени клиента, при вводе имени осуществляется поиск по списку клиентов, и если такой находится, то вывести:

- общий счет покупок клиента (сумма всех покупок)
- текущий счет клиента (деньги на счету фирмы для покупок);
- потолок кредита (кредитный лимит, который не изменяется);
- текущий долг клиента (;
- остаток кредита (разница между потолком кредита и текущим долгом);
- комментарий (о причине долга клиента, его надежности и т.п.).

2. Открыть базу товаров

3. Создать новый заказ

Нужна таблица склада товаров.

4. Выбор вида продажи:
 - наличный расчет (увеличивается общий счет покупок клиента, остальные счета не изменяются, количество товаров уменьшается);
 - безналичный расчет (увеличивается общий счет покупок клиента, уменьшается текущий счет клиента, остальные счета не изменяются, количество товаров уменьшается);
 - кредит (увеличивается общий счет покупок клиента, увеличивается текущий долг клиента, уменьшается остаток кредита, текущий счет клиента в этом случае нулевой или недостаточный для покупки (во втором случае он должен быть учтен при покупке и обнулится), количество товаров уменьшается; при приближении текущего долга клиента к потолку кредита (90% от потолка кредита) подсвечивается

красным текущий долг и остаток кредита; количество товаров уменьшается)

- бартер, т.е. обмен товаров без использования денег (никакие счета не изменяются, на складе количество одних товаров должно увеличиться, количество других уменьшиться, стоимость товаров, количество которых увеличилось, должно быть равно стоимости товаров, количество которых уменьшилось);
- взаимозачет (уменьшается текущий долг клиента, увеличивается остаток кредита, остальные счета не изменяются, количество товаров увеличивается).

5. Открыть базу товаров.

6. В новом заказе в форме в разделе таблица выбор товара из перечня, а также ввод количество товаров, которые пользователь заказывает.

7. Ввод перечня товаров и продаваемого количества в табличной части, с автоматическим указанием:

- цены (из списка товаров); -?
- суммы за единицу; -? Сумма за единицу заказа, сумма всего заказа, состоящего из списка товаров, общая цена за список товаров.
- выводом текущего остатка.
 - Если введенное количество товаров меньше текущего остатка или равно ему, то цвет шрифта остатка должен быть одним (например, зеленым).
 - Если введенное количество товаров больше текущего остатка, то цвет шрифта остатка должен быть другим (например, красным), а также должно сработать звуковое предупреждение. (в случае наличия устройства воспроизведения звука)

8. Автоматический подсчет общей суммы купленных товаров для каждого клиента. Запрос на формирование таблицы, состоящей из полей таблиц, а также из дополнительного вычисляемого поля, где подсчитана общая

сумма для каждого заказа клиента. Построение отчета на основе данной таблицы.

9. Формирование предупреждения, если общая сумма купленных товаров для клиента превышает потолок его кредита за минусом текущего долга клиента. При формировании нового заказа в случае, если выбрать способ покупки с помощью кредита.

10. Формирование предупреждения если есть строки с незаполненным товаром, нулевой ценой и нулевым количеством. Данное предупреждение реализовывается в разделе заполнения базы товаров. Открыть базу товаров.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Как устанавливаются и удаляются связи между таблицами?
2. Как удалить/установить тип связи?
3. Как обеспечить целостность данных? Как работают каскадное удаление, добавление?
4. Как создать шаблон ввода телефонного номера в соответствии со следующими условиями:
 - Вводится только 10 цифр (не больше);
 - Все остальные знаки (скобки, дефисы) появляются при вводе автоматически;
 - Ввод букв невозможен;

То есть в таком формате: +7(000)-000-00-00.

Лабораторная работа №2. Принципы создания подсистемы вывода информации в информационных системах

(1 вариант)

Цель работы: Изучить принципы создания интерфейсов подсистемы вывода информации.

Содержание работы:

1. Изучить теоретические сведения.
2. Разработать приложение согласно заданию.
3. Ответить на контрольные вопросы.

ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

В различных информационных системах используются современные интеллектуальные методы анализа оперативно поступающей информации, однако все еще часто при высокой скорости поступления и большой ответственности за принятие решения применяются операторы-эксперты. Их задачей является визуальный контроль текущих значений параметров, их динамики, текущий и ретроспективный анализ состояния и составление прогнозов. В составе информационной системы предприятия для повышения эффективности работы таких специалистов используются специальные автоматизированные рабочие места. К интерфейсам таких АРМ предъявляются жесткие требования, так как пользователь большое время проходит за визуальным анализом данных, а также необходима возможность эффективного использования математических методов для анализа выборок из текущих данных.

К таким интерфейсам предъявляются дополнительные **требования**, к которым относятся:

- 1) Четко различимый шрифт текстовой и цифровой информации;
- 2) Использование нейтральных или ярких цветов шрифтов фонов, рамок и других элементов оформления для выделения существенной информации.
- 3) Рациональное, связанное по смыслу и направлению взгляда размещение информации на экране.
- 4) Развитая логическая система предупреждений и оповещений о штатных событиях.
- 5) Логическая система активных предупреждений и оповещений о нештатных событиях.
- 6) Система управления, позволяющая максимально быстро и эффективно осуществлять необходимые действия с поступающей информацией.
- 7) Повышенная отказоустойчивость.

ЗАДАНИЕ НА РАБОТУ

Разработать и реализовать согласно требованиям интерфейс программы, предназначенной для использования в качестве автоматизированного рабочего места специалиста по контролю и анализу поступающей информации одного параметра согласно заданию.

В данном АРМ необходимо предусмотреть следующие возможности:

1. отображение значений поступающих данных: текущего значения, $N1=2..10$ предыдущих;
2. отображение полученной информации на экране в виде графика;
3. возможность выбора значений из группы текущих данных в группу выборочных значений указанным в задании способом, больше заданной границе, меньше заданной границе и кратно определенному значению;
4. отображения выборочных значений в виде графика; дополнительно показать красной линией среднее значение.
5. отображение диаграммы выборочных значений;
6. визуальная и звуковая индикация предупреждения при изменении текущего значения более чем на $N2=5..40$ процентов;
7. визуальная и звуковая индикация тревоги при выходе текущего значения из заданной области допустимых значений.

Рекомендуемый стек технологий:

Frontend: html

Backend: python, django, matplotlib, numpy, render.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Перечислите, прокомментируйте и приведите примеры соблюдения требований к интерфейсам вывода информации.

Лабораторная работа №3. Принципы открытого обмена данными между независимыми компонентами программного обеспечения информационных систем.

(6 вариантов)

Цель работы: Изучить возможности обмена данными между функционально-различными компонентами информационной системы через открытый интерфейс.

Содержание работы:

1. Изучить теоретические сведения.
2. Разработать приложение согласно заданию.
3. Ответить на контрольные вопросы.

ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

При создании информационных систем часто возникает необходимость организации обмена данными в режиме реального времени.

Как правило, такая необходимость возникает в следующих случаях:

- 1) При организации совместной работы существующей и новой частей ИС.
- 2) При распределении процессов обработки данных между специализированными компонентами ПО ИС по функциональному признаку.
- 3) При пространственном удалении средств обработки данных между специализированными компонентами ПО ИС по функциональному признаку.

Когда необходимые для связывания компоненты ПО ИС являются законченными программными продуктами, разработанные квалифицированными специалистами, то в большинстве случаев такие программы используют для обмена данными с другими программами распространенные стандартизированные методики, протоколы и интерфейсы.

Для обмена данными используются различные стандарты открытого интерфейса. Например, такие как OLE и DDE. Эти интерфейсы широко используются в операционной системе Microsoft Windows в частности для обмена данными между программами пакета Microsoft Office.

Интерфейс OLE используется для внедрения объектов из других приложений в текущее рабочее окно, при этом отображение и действия с этим внедренным объектом выполняются средствами приложения, в котором этот объект был создан, остальная часть рабочего окна управляется приложением которому принадлежит рабочее окно.

Интерфейс DDE используется для установки связи DDE с документом, созданным в другом приложении Windows. В случае изменения связанных данных новые данные вставляются в документ при обновлении поля, имеющего связь через DDE.

Однако некоторые специализированные программы должны иметь возможность обмена информацией с различным ПО независимо от вида операционной системы и по универсальным правилам, малокритичными к ресурсам ПЭВМ и ОС.

Например, программы систем безналичных платежей типа Банк-клиент имеют возможность загружать и выгружать информацию из и в файлы произвольного формата, так как пользователи одной версии программы банк-клиент имеют различные операционные системы и бухгалтерские программы которым необходимо взаимодействовать с программой банк-клиент.

При использовании методов обмена данными произвольного формата данные передаются обычно через временный файл. Программа источник выгружает данные в файл, программа приемник загружает данные из файла. Главным условием взаимодействия ПО является согласование формата временного файла между программой источником и приемником.

Довольно часто для промежуточного файла выбирается один из популярных форматов файлов баз данных (например dBase III, dBase IV). Однако, при необходимости передачи между программами дополнительного

параметра без остановки их работы, при добавлении дополнительного реквизита (поля) или изменения параметров (типа) существующего поля в одной из программ программе источнике программа приемник не сможет открыть файл данных, так как изменился его формат.

Наиболее универсальным является способ передачи данных через текстовые файлы с разделителями: служебными символами, которыми разделяются поля данных. В этом случае при изменении формата отдельных полей или количества полей загрузка программой приемником данных все равно состоится.

В настоящее время получает все более широкое распространение метод передачи данных Microsoft XML, который во временных файлах передает не только данные, но и правила конвертации типов данных из формата программы источник в формат программы приемник.

Однако в большинстве программ используется вывод данных через текстовые файлы с полями фиксированной ширины с указанием типов данных.

ЗАДАНИЕ НА РАБОТУ

Разработать демонстрационную установку для показа возможности обмена данными между тремя программами. Две из них должны быть написаны студентом. Первая программа генерирует данные и передает их для обработки второй программе. Вторая программа должна получить эти данные, обработать их и передать для визуализации третьей программе. Назовем первую программу источником, вторую - сервером третью - визуализатором.

Первая программа передает данные задаваемого в ней формата в задаваемом количестве и в задаваемой последовательности.

Для этого в программе должны быть предусмотрены возможности по созданию структуры данных и заполнению данными.

В программе-источнике должны быть возможности (подобно программам СУБД) редактировать структуру информации, указав код поля, номер по порядку, наименование, тип значения, точность и признак того будут ли эти данных передаваться. Ограничимся простыми типами – строка, дата и число.

Для заданной структуры данных должна быть возможность заполнить ее значениями в таблице и выгрузить во внешний файл для последующей обработки. При выгрузке также должна передаваться структура данных.

Программа-сервер должна выполнить две команды пользователя: загрузка данных от источника и передача обработанных данных на визуализатор.

Данные из программы-сервера в программу - визуализатор передаются посредством открытого интерфейса OLE. В роли визуализатора используем программу Microsoft Excel, которая в данном случае используется для представления полученных данных в виде диаграммы.

В программе Microsoft Excel заранее подготавливается таблица, в которой выделяется группа ячеек, куда будут динамически помещаться получаемые данные. Создается и конфигурируется диаграмма, которая будет строиться по данным из выделенных ячеек.

В результате работы: генерируемые первой программой данные должны обрабатываться второй и результаты отображаться в третьей программе и одновременно по этим же данным в листе Microsoft Excel должна динамически строиться диаграмма.

Для работы с OLE объектами в языке Паскаль существует стандартная функция CreateOleObject. Она возвращает значение созданного OLE объекта, через который можно обращаться к внутренним функциям связанного приложения. Для связи с Excel необходимо указать аргументом этой функции : 'Excel.Application'.

Управление Microsoft Excel для обработки полученных данных из разработанной программы-источника осуществляется с помощью команд встроенного в Microsoft Office объектно-ориентированного языка Visual Basic.

Ниже приведен фрагмент исходного кода на языке Visual Basic реализующий открытие в Excel заданного файла и занесение значений в ячейки с указанными координатами.

```
Workbooks("cogs.xls").Worksheets("sheet1").cell(2,6).Value=3,14;
```

Заполнение значений ячеек таблицы происходит при невидимости заполняемой таблицы, поэтому после его завершения необходимо установить значение 1 для стандартного флага видимости Visible.

Далее приводятся примеры команд языка Visual Basic используемые для управления Excel через интерфейс OLE в редакторе текста модуля Delphi.

```
var
vExcel, xlvariant: OLEVariant;
begin
vExcel := CreateOleObject('Excel.Application');
vExcel.Workbooks.Open('BigBook.xls', False);
...
vExcel.Cells[1, 1].Value := 124512;
vExcel.Cells[1, 2].Value := 'С добрым утром';
vExcel.ActiveSheet.Range['B5:B100'].Select;
vExcel.Selection.Columns.AutoFit;
...
xlVariant.ActiveSheet.Range['B5:B100'].Select;
{Левая рамка}
xlVariant.Selection.Borders[7].LineStyle := 1;
xlVariant.Selection.Borders[7].Weight := 2;
xlVariant.Selection.Borders[7].ColorIndex := -4105;
vExcel.Visible := True;
```

Подробнее о способах связывания Delphi и Excel можно узнать на Internet-сайте по адресу http://www.mcsa.ru/f5_p.shtml

ВАРИАНТЫ ЗАДАНИЙ

Реализовать отображение динамики следующих процессов:

Для вариантов 1-3 в исходных данных должны указываться дата, курс на дату, название биржи давшей такой курс наименование источника информации о курсе. Вид обработки – усреднение значений курса на каждую дату по всем биржам.

1. Биржевой курс валюты Доллар.
2. Биржевой курс валюты Евро.
3. Биржевой индекс S&P.

Для вариантов 4,5 исходными данными являются: номер датчика, место расположения датчика, номер зоны (сектора) расположения, дата и время показания, значение параметра. Вид обработки вывод средних значений показаний по каждой зоне на задаваемую пользователем дату и время.

4. Температура атмосферного воздуха.
5. Температура пара в камере реактора.

Для варианта 6 исходными данными являются: наименование товара, группа товара, проданное количество, продажная цена, закупочная цена, скидка. Вид обработки определение суммы прибыли по каждой группе товаров. (Прибыль одной продажи равна: Цена продажи без скидки * количество минус цена закупки * количество)

6. Продажи товаров через кассовый аппарат.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. В каких случаях необходимо организовывать обмен данными между компонентами информационной системы.
2. Какие вы знаете способы по обмену данными.

Лабораторная работа №4. Основы организации распределенной работы пользователей

(1 вариант)

Цель работы: изучить методы ограничения доступа к базам данных на основе авторизации пользователей. Научиться создавать приложения с разграничением доступа.

Содержание работы:

1. Изучение методических рекомендаций.
2. Создание приложения согласно заданию.

ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

В большинстве современных информационных систем обработки данных используются различные методы защиты информации от несанкционированных действий. Несанкционированные действия могут выполняться, как правило, или посторонними пользователями или пользователями, не имеющими достаточной квалификации для выполнения определенных действий.

Для определения пользователя, начавшего работу с системой, используются различные методы авторизации. Простейший состоит в том, что каждому пользователю назначается определенное идентифицирующее имя и персональный пароль. Вход в систему ограничивается только зарегистрированными пользователями после правильного ввода его пароля.

В системе создаются наборы прав – перечни разрешенных действий пользователя определенного типа для работы с каждым фрагментом системы. Таким образом, набор прав - это список возможных действий пользователя. Для каждого из этих действий, в зависимости от необходимости, устанавливается флаг разрешения или запрета относительно полномочий конкретного пользователя. Списки прав обычно хранятся в зашифрованном виде в недоступном для просмотра участке дискового пространства.

Каждому пользователю назначается определенный набор прав, в соответствии с которым он работает в системе. Распознавание прав пользователей осуществляется с помощью авторизации.

В случае определенного количества раз неправильно введенных комбинаций имен пользователя и пароля, применяются различные методы активной защиты, к числу которых относится полный запрет на вход в систему, привлечение внимания к рабочему месту с которого производятся неверные попытки входа, скрытый вызов охраны и др.

Так системы ограждаются от посторонних и предохраняются от случайных сбоев, вызванных некорректными действиями.

ЗАДАНИЕ НА РАБОТУ

Необходимо создать приложение для работы со списком данных личной информации о работниках некоторой фирмы с различными ограничениями права доступа пользователей.

В каждой строке списка сотрудников содержится следующая информация:

- фамилия;
- имя, отчество;
- должность;
- адрес;
- телефон.

Необходимо реализовать 4 набора прав по следующим критериям:

1. Директор. Есть все права (любые изменения списка сотрудников: добавление, изменение, удаление). Вход по паролю.
2. Заместитель директора. Есть все права, кроме удаления и добавления сотрудников. Вход по паролю.
3. Секретарь. Права на просмотр всех данных, нет прав на изменение любых данных. Вход по паролю.

4. Гость. Просмотр только фамилии, имени, отчества каждого сотрудника, должности, рабочего телефона, без возможности просмотра адреса и личного телефона. Вход без пароля.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Какими способами осуществляется авторизация доступа в программу.
2. Какие виды ограничения доступа вы знаете.

Лабораторная работа №5. Использование информационных ресурсов Internet в разрабатываемых ИС

(2 варианта)

Цель работы: изучить особенности использования информационных ресурсов Internet в разрабатываемых информационных системах в режиме on-line. Научиться создавать приложения, использующие при работе информацию из Internet.

Содержание работы:

1. Изучение методических рекомендаций.
2. Создание приложения согласно заданию.

ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Глобальная информационная сеть Internet содержит множество постоянно обновляемой информации и данных, которые можно, а иногда и жизненно необходимо использовать в разрабатываемом ПО ИС.

Существует несколько способов получения информации из Internet:

- on-line считывание из HTML страниц Web –серверов;
- on-line загрузка файлов с данными с Web–серверов или FTP –серверов;
- получение данных из автоматически рассылаемой электронной почты;
- получение данных по каналам on-line общения (ICQ и др.);
- другие способы.

Наиболее оперативным способом из перечисленных является считывание данных непосредственно из серверов – из текстов Web -страниц или подготовленных для считывания файлов.

Многие современные среды проектирования имеют возможности для интеграции в разрабатываемые программы средств для доступа к Web-серверам и обмену с ними информацией.

В среде проектирования Delphi6 для связи с on-line html страницей используется стандартный компонент TNMHTTP. У него есть свойства:

- NMHTTP1.Proxy – адрес Proxy сервера.
- NMHTTP1.ProxyPort – порт.
- NMHTTP1.Body – хранит html код страницы полученный методом get.

и методы:

- NMHTTP1.Get(адрес html страницы); - получение html кода.

Адрес прокси сервера можно найти, используя DOS команду [ipconfig](#).

Если прокси сервер не установлен, поля Proxy и ProxyPort нужно оставить пустыми.

Просмотрев код html файла, можно выделить повторяющиеся элементы кода, в тэгах которого содержится нужная информация:

Пример фрагмента считанного кода:

```
<TD class="ind"><A HREF=" ../daily/AUD.rus.shtml?show=3M"
class="n"> AUD</A></TD>
<TD class=W0>1</TD>
<TD class=custom><B>Австралийский доллар</B></TD>
<TD class=W0>21.129</TD>
<TD class=W0R>-0.1169</TD>
<TD class=W0R>-0.5502</TD>
<TD class=W0>36</TD>
```

Нужная информация подчеркнута. Следовательно, нам необходимо отфильтровать блоки html кода в которых содержатся данные. Для получения html кода используется вышеописанный компонент TNMHTTP.

ЗАДАНИЕ НА РАБОТУ

Разработать программу для on-line чтения курсов валют.

Данная программа должна получать и показывать значения текущих курсов валют по выбору пользователя. Информация о курсах валют должна автоматически считываться с сайта www.rbc.ru ссылка «Курсы ЦБ РФ» (<http://stock.rbc.ru/demo/cb.0/intraday/>).

Выбор валюты и получение значение курса должно выполняться непосредственно в разработанной программе, как потенциальной части экономической ИС.

Вариант 1. Необходимо реализовать показ курсов одной из трех валют заранее установленных в программе по выбору пользователя.

Вариант 2. Необходимо реализовать показ курсов любых валют из списка валют по вышеуказанной ссылке.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Приведите примеры задач в которых необходимо on-line считывание программ из Internet.
2. Какие способы получения информации из Internet можно использовать для автоматического использования в ПО ИС.

Лабораторная работа №6. Совместная разработка ИС.

Элементы документирования разработки

(1 вариант)

Цель работы: Изучить основы взаимодействия разработчиков ИС.

Содержание работы:

1. Изучение методических рекомендаций.
2. Создание приложения согласно заданию.

ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

При проектировании информационных систем обычно используется несколько разработчиков. Отдельные исполнители создают отдельные компоненты технического обеспечения, разные люди создают программное обеспечение. При этом процесс разработки может быть сильно растянут во времени, а также могут использоваться уже давно существующие наработки и компоненты.

Поэтому очень важно чтобы все участники проектирования могли не только грамотно оставлять техническую информацию о своей разработке, но и использовать такую информацию, оставленную коллегами из другого подразделения, организации или предшественников, или оставленную самим разработчиком, но давно.

При использовании «чужих» наработок в создании информационной системы пользуются имеющейся технической документацией и другими описаниями.

В данной работе будут рассмотрены некоторые элементы для облегчения дальнейшего использования созданного ПО и создания технической документации.

Программисты являются категорией разработчиков, которым очень много приходится использовать сторонние решения при разработке, внедрении и сопровождении программ.

Основным способом фиксирования информации о разрабатываемой программе являются комментарии в исходных текстах. Грамотно составленные комментарии значительно повышают восприятие исходного текста и назначения программы и ускоряют нахождение ошибок.

Кроме того, комментарии используются для автоматизированной обработки текстов программ. Одним из первых видов служебных комментариев было оставление в них команд компилятору.

Также комментарии используются для автоматизированного составления технической документации на исходные тексты программ.

Некоторые основные правила составления комментариев:

- 1) Комментарии должны быть достаточно подробными и понятными для чтения другим специалистам.
- 2) Комментарии должны иметь графическое выделение символами для легкого визуального отделения комментариев от основного текста программы.
- 3) Комментарии при изменении готовой программы должны содержать код специалиста, выполнившего изменение, дату изменения, причину изменения и первоначальный фрагмент текста программы.
- 4) Каждый фрагмент исходного текста, содержащий сложные или специфические действия (расчеты, обработки, операции с БД или памятью, ссылки на другое ПО) должен содержать грамотное и подробное описание выполняемых действий.
- 5) Каждая глобальная переменная должна иметь комментарий.
- 6) Каждый модуль должен иметь комментарий.
- 7) Каждая подпрограмма должна иметь развернутый комментарий, содержащий назначение этой подпрограммы, описание, тип, назначение, и условия наличия значения каждой формальной и фактической переменной.

ЗАДАНИЕ НА РАБОТУ

На основе любой программы по ранее выполненным лабораторным работам (например, текст программы из пятой или четвертой лабораторной работы) снабдить комментариями (каждую строчку) и привести в соответствие исходный код (SQL или VBA) программы в соответствии с вышеперечисленными правилами. Код должен быть рабочим.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Назовите причины грамотного комментирования исходных текстов программ.
2. Перечислите предложенные правила составления комментариев.

Лабораторная работа №7. Создание ИС на основе типовых решений. Изменение информационных объектов и баз данных

(2 варианта)

Цель работы: изучить основные принципы модификации готового программного обеспечения и практически попробовать изменить типовую конфигурацию программы 1С Предприятие.

Содержание работы:

1. Изучение методических рекомендаций.
2. Изменение ПО и БД ИС согласно заданию.

ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

В настоящее время в мире существует огромное количество разработанного, внедренного и работающего программного обеспечения ИС. Одной из самых распространенных задач и, как следствие, востребованных способностей для программистов является умение сопровождать существующее ПО без его остановки и использовать готовое ПО для создания новых систем.

При модификации эксплуатируемого ПО необходимо **СТРОГО** придерживаться следующих правил:

1. Перед изменением ПО необходимо сделать **две резервных копии** ПО, всех его файлов, настроек и баз данных, для этого на некоторое время необходимо остановить работу ПО и всех его пользователей.
2. Любые изменения ПО необходимо осуществлять на первой резервной копии, особенно контролируя, чтобы ПО резервной копии никаким образом не влияло на работу основной программы и ее баз данных.
3. Перед модификацией ПО необходимо получить и изучить всю возможную информацию о данном ПО и реестр изменений, выполненных предыдущих модификациях.

4. При выполнении любого даже самого незначительного изменения ПО необходимо выполнить синтаксический контроль и тщательно протестировать все функции программ использующих измененный фрагмент.
5. При выполнении модификаций, после каждой успешно выполненной подзадачи или функциональной части, необходимо делать резервную копию ПО и БД текущей резервной копии, для того чтобы неправильные последующие действия не испортили работу с самого начала.
6. Каждую успешную модификацию (подзадачу, функциональный элемент) необходимо фиксировать в реестре изменений с указанием назначения изменения, мест изменения ПО и структуры БД, даты и специалиста выполнившего изменение.
7. После выполнения всех модификаций еще раз протестировать все элементы программ, которые могут использовать измененные фрагменты.
8. Показать пользователям и объяснить новые изменения в работе ПО.
9. Настоять и проконтролировать чтобы пользователи самостоятельно выполнили функции программ подвергшиеся изменению на первой резервной копии.
10. Удаться, если при действиях пользователя измененные функции не работают.
11. Заменить основную программу на модифицированную.

Несоблюдение этих правил может привести к катастрофическим последствиям для ПО и БД, организации в которой они эксплуатируются и лично для программиста выполнявшего модификацию.

ЗАДАНИЕ НА РАБОТУ

Выполнить изменения типовой конфигурации программы «1С Бухгалтерия версии 4.*» согласно заданию и выполнить действия в

соответствии с вышеперечисленными правилами. Перед выполнением заданий прочитайте замечания после вариантов.

Варианты заданий:

1) Реорганизация учета основных средств.

Добавить для основных средств новый параметр (реквизит) «Код внутреннего учета» в базу данных и диалоговые окна;

Составить обработку по автоматическому заполнению этого параметра значениями по правилу «Код внутреннего учета»= «Код подразделения»+ «Порядковый номер ОС в подразделении указанном как место хранения ОС».

Добавить в диалог ввода информации об основном средстве кнопку «Назначить внутренний код». При нажатии пользователем этой кнопки для новых вводимых элементов справочника автоматически назначается внутренний код по вышеперечисленному правилу.

2) Проверка правильности заполнения справочника контрагентов по ИНН.

Реализовать автоматическую проверку на наличие непустого значения реквизита ИНН при интерактивной записи элементов справочника контрагенты с выдачей предупреждения об ошибке при пустом значении.

Составить обработку для автоматической проверки наличия в базе данных в справочника «Контрагенты» двух контрагентов с одинаковым ИНН. При обнаружении таких элементов должно быть выдано сообщение о наименованиях и кодах найденных элементов и дубликаты должны быть помечены на удаление.

Добавить в диалог ввода информации о контрагенте кнопку «Проверить ИНН». При нажатии пользователем этой кнопки для новых вводимых элементов справочника автоматически выполняется проверка всех элементов справочника на наличие элемента с таким же значением реквизита ИНН или вхождением такого же значения как подстроки.

Результат проверки должен быть показан в форме предупреждения.

Замечания:

- первые шаги по конфигурированию в программе «1С Предприятие 7.7» кратко описаны в теоретических сведениях к лабораторной работе №8,9.
- автоматические разовые изменения базы данных удобно осуществлять реализуя алгоритмы создавая новые внешние отчеты.
- кнопки обычно добавляются рядом с полями ввода к которым они относятся по смыслу в формах элемента справочника.
- для удаления пробелов слева и справа при сложении строк используется функция СОКРЛП().
- событие «Запись элемента справочника» перехватывается и обрабатывается предопределенной процедурой ПРИЗАПИСИ().
- анализ содержимого справочника один из элементов которого открыт для редактирования осуществляется через ссылку создаваемую функцией СОЗДАТЬ ОБЪЕКТ(«Справочник.XXXXX») и перебором значений стандартным образом с помощью пары команд типа ВЫБРАТЬXXXX и ПОЛУЧИТЬXXXX
- при обработке таблиц данных полезно для наглядности сообщать об обработанных элементах.
- в данной лабораторной работе для демонстрации работы созданных алгоритмов внести новые записи в базу данных самостоятельно.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Перечислите правила модификации эксплуатируемого ПО.
2. Объясните возможные последствия несоблюдения каждого из пунктов этих правил.

Лабораторная работа №8. Создание ИС на основе типовых решений. Добавление печатных форм (2 варианта)

Цель работы: изучить методы работы в конфигурации программы MS Access с объектом «таблица». Научиться создавать простейшие печатные формы в системе. *скорректировать цель? как?*

Содержание работы:

1. Изучение методических рекомендаций.
2. Создание печатной формы согласно заданию.

ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

О печати данных в Access

Печать данных напрямую из таблиц или запросов хорошо подходит для простых списков данных. Формы подходят для просмотра или печати данных одной записи или набора связанных записей, но не крупных наборов данных или сводок. Отчеты позволяют сравнивать, обобщать и сводить крупные наборы данных, а затем выводить их на печать в привлекательном виде.

Основные сведения об отчетах в Access

Отчет — это объект базы данных, удобный для представления ее информации с одной из следующих целей:

- отображение или распространение сводки данных;
- архивирование моментальных снимков данных;
- предоставление сведений об отдельных записях;
- создание надписей.

Составные части отчета

Макет отчета разбит на разделы, которые можно просмотреть в режиме конструктора. Ниже перечислены типы разделов и указано назначение каждого из них:

Раздел	Отображение при печати	Использование раздела
Заголовок отчета	В начале отчета	<p>В заголовок включается информация, обычно помещаемая на обложке, например, эмблема компании, название отчета или дата.</p> <p>Если в заголовке отчета помещен вычисляемый элемент управления, использующий статистическую функцию Sum, сумма рассчитывается для всего отчета.</p> <p>Заголовок отчета печатается перед верхним колонтитулом.</p>
Верхний колонтитул	Вверху каждой страницы	Верхний колонтитул используется в тех случаях, когда нужно, чтобы название отчета повторялось на каждой странице.
Заголовок группы	В начале каждой группы записей	<p>Используется для печати названия группы. Например, если отчет сгруппирован по изделиям, в заголовках групп можно указать их названия.</p> <p>Если поместить в заголовок группы вычисляемый элемент управления, использующий статистическую функцию Sum, сумма будет рассчитываться для текущей группы.</p> <p>Заголовок группы может состоять из нескольких разделов в зависимости от добавленных уровней группирования.</p>
Область данных	Отображается один раз для каждой строки в источнике записей	Здесь размещаются элементы управления, составляющие основной содержание отчета.
Примечание группы	В конце каждой группы записей	Примечание группы можно использовать для печати сводной информации по группе.
Нижний колонтитул	Внизу каждой страницы	Используется для нумерации страниц и для печати постраничной информации.
Примечание отчета	В конце отчета	Примечание отчета можно использовать для печати итогов и другой сводной информации по всему отчету.

Создание отчета в Access

Для создания отчета в Access нужно выполнить следующие действия:

1. Выбор источника записей

Источником записей для отчета может быть таблица, именованный или внедренный запрос. Источник записей должен содержать все строки и столбца данных, которые требуется отобразить в отчете.

- Если нужные данные содержатся в существующей таблице или запросе, выделите эту таблицу или запрос в области навигации и перейдите к действию 2.
- Если источник записей еще не создан, выполните одно из указанных ниже действий.
 - Перейдите к действию 2 и воспользуйтесь инструментом Пустой отчет;
Или
 - создайте таблицы или запрос, которые будут содержать нужные данные, выберите их в области навигации и перейдите к действию 2.

2. Выбор инструмента отчета

Инструменты отчета расположены на вкладке **Создать** в группе **Отчеты**. В следующей таблице описаны такие параметры:

Инструмент	Описание
Отчет	Позволяет создать простой табличный отчет, содержащий все поля из источника записей, который выбран в области навигации.
Конструктор отчетов	Открывает в режиме конструктора пустой отчет, в который можно добавить необходимые поля и элементы управления.
Пустой отчет	Позволяет открыть пустой отчет в режиме макета и отобразить область задач «Список полей», из которой можно добавить поля в отчет.

Мастер отчетов	Служит для вызова пошагового мастера, с помощью которого можно задать поля, уровни группировки и сортировки и параметры макета.
Наклейки	Вызывает мастер, в котором можно выбрать стандартный или настраиваемый размер подписей, набор отображаемых полей и порядок их сортировки.

3. Создание отчета

- 1) Для выбора требуемого инструмента нажмите соответствующую кнопку панели инструментов. После появления мастера следуйте всем его командам и на последней странице нажмите кнопку Готово. Access отображает отчет в режиме макета.
- 2) Отформатируйте отчет, чтобы добиться желаемого внешнего вида.
 - Измените размер полей и подписей, выделяя их и перетаскивая края.
 - Расположите поля в нужном порядке, выделяя их (и соответствующие подписи, если они есть) и перетаскивая в нужное место.
 - Также можно, щелкая поля правой кнопкой мыши, с помощью команд контекстного меню объединять или разбивать ячейки, удалять и выделять поля и выполнять другие задачи форматирования.

Добавление группировки, сортировки и итогов

Чтобы быстро добавить в отчет группировку, сортировку или итоги, щелкните правой кнопкой мыши поле, к которому необходимо применить соответствующую функцию, и выберите нужную команду в контекстном меню.

Кроме того, когда отчет открыт в режиме макета или конструктора, можно добавить эти функции с помощью области "Группировка, сортировка и итоги".

1. Если область "Группировка, сортировка и итоги" не открыта, на вкладке **Конструктор** в группе **Группировка и итоги** выберите команду **Группировка**.
2. Нажмите кнопку **Добавить группировку** или **Добавить сортировку** и выберите поле, по которому требуется сгруппировать или отсортировать данные.
3. Чтобы задать дополнительные параметры или добавить итоги, в строке группировки или сортировки выберите команду **Больше**.

Выделение данных с помощью условного форматирования

В приложении Access есть инструменты выделения данных в отчетах. Правила условного форматирования можно добавить для каждого элемента управления или их группы, а в клиентских отчетах можно использовать гистограммы для сравнения данных.

Добавление условного форматирования для элементов управления:

1. Щелкните отчет в области навигации правой кнопкой мыши и выберите пункт **Режим макета**.
2. Выберите нужные элементы управления, а затем на вкладке **Формат** в группе **Форматирование элемента управления** выберите команду **Условное форматирование**.
3. В диалоговом окне **Диспетчер правил условного форматирования** выберите команду **Создать правило**.
4. В диалоговом окне **Новое правило форматирования** выберите значение в поле **Выберите тип правила**.
 - Чтобы создать правило, которое будет проверяться отдельно для каждой записи, выберите вариант **Проверьте значения в текущей записи или используйте выражение**.
 - Чтобы создать правило для сравнения записей друг с другом с помощью гистограмм, выберите вариант **Сравнить с другими записями**.

1. В поле **Изменение описания правила** укажите правило, которое будет применяться при форматировании, и тип форматирования, а затем нажмите кнопку **ОК**.
2. Если нужно создать еще одно правило для этого элемента управления или группы элементов, повторите процедуру, начиная с действия 4.

Предварительный просмотр и печать отчета

Предварительный просмотр

1. Щелкните правой кнопкой мыши имя отчета в области навигации и выберите команду **Предварительный просмотр**. Команды на вкладке **Предварительный просмотр** можно использовать, чтобы:
 - напечатать отчет;
 - изменить размер или макет страницы;
 - изменить масштаб или просмотреть сразу несколько страниц;
 - обновить данные в отчете;
 - экспортировать отчет в другой формат файла.
2. Нажмите кнопку **Заккрыть режим предварительного просмотра**.

Печать отчета

Печать отчета без предварительного просмотра:

- Щелкните правой кнопкой мыши имя отчета в области навигации и выберите команду **Печать**. Отчет будет отправлен на принтер, используемый по умолчанию.
- Чтобы открыть диалоговое окно, в котором можно выбрать принтер, указать число копий и задать другие параметры, выберите команду **Печать**.

Добавление в отчет кнопки «Печать»

1. Открыть отчет в режиме конструктора.
2. Найти в элементах управления элемент «Кнопка», поставить ее в нужное место отчета.
3. Выбрать пункты **Работа с отчетом**, **Печать отчета**, затем выбрать отчет, который будет печататься, далее выбрать текст или рисунок, который будет отображаться на кнопке, затем дать кнопке имя и нажать кнопку **Готово**.

О печатных формах документов в 1С. Сформируйте «Отчет по всем сотрудникам» (Меню «Отчеты»).

Как формируется каждый отчет и правила вывода на печать.

Правила создания печатных форм.

Создание печатной формы для справочника.

Шаги создания и справка по каждому из них.

Что такое справочники в 1С

Справочник – это агрегатный тип данных, средство для работы со списками однородных элементов данных. Название и структура каждого конкретного справочника определяются при его создании в конфигураторе. У любого справочника существуют два реквизита, которые создаются автоматически – «Код» и «Наименование». Реквизиты справочников могут быть периодическими, т.е. иметь значения, связанные с датой. При изменении значения периодического реквизита старое значение сохраняется, при этом новое значение начинает действовать с указанной даты, старое – до указанной даты. Свойства справочника редактируются в окне редактирования «Справочник». Для визуального представления справочника существуют «Форма элемента», «Форма группы» и «Формы списка» (их может быть несколько).

ЗАДАНИЕ НА РАБОТУ

1. Добавить печатные формы в типовой конфигурации MS Access согласно варианту:

Вариант 1. Создать печатную форму для печати для справочника «Материалы» с полями: *(в аксесе ведь не справочники, а просто таблицы)*

- Код;
- Наименование;
- Счет Учета.

Вариант 2. Создать печатную форму для печати сумм начисленной зарплаты сотрудникам из документа «Начисление зарплаты» с полями:

- № п/п,
- Сотрудник,
- Начислено.

2. Реализовать печатную форму по образцу. *(что это вообще означает? создать отчет? печатная форма это отчет? в чем отличие между «создать» и «реализовать»?)*

Но то есть по сути: создаю таблицы с указанными данными и создаю по ним отчет, который возможно отправить на печать, верно? Нужно добавить это в задание?)

3. В форме подсвечивать: *в форме или в получившемся отчете?*

- 1) Выше верхнего порога – один цвет;
- 2) Ниже нижнего порога – другой цвет;
- 3) Между порогами – третий;
- 4) Кратно / Не кратно какому-либо числу – четвертый цвет.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Объясните основные принципы вывода на печать в программе 1С предприятие 7.7.
2. Где создаются шаблоны печатных форм.

3. Перечислите команды языка системы 1С «Предприятие» для работы с печатными формами.

Лабораторная работа №9. Создание ИС на основе типовых решений. Добавление новых информационных объектов

(1 вариант)

Цель работы: Изучить основы добавления новых информационных объектов на примере конфигурирования программы MS Access.

Содержание работы:

1. Изучить теоретические сведения.
2. Изменить программу согласно заданию.

ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Задача добавления новых информационных объектов [какие здесь подразумеваются объекты?] в эксплуатируемую ИС довольно актуальна. Как правило такая задача возникает при необходимости расширить функции ПО ИС для охвата новых или ранее не учитываемых областей деятельности.

Добавить то же самое, но об Аксесе. Для этого нужно выяснить, что есть объекты в аксесе.

Рассмотрим создание новых объектов в системе «1С Предприятие 7.7».

Для изменения конфигурации информационной базы программу необходимо запустить в режиме «Конфигуратор». Затем в главном меню Раздел «Конфигурация» - «Открыть конфигурацию» открывает среду настроек: «Метаданные». Она имеет вид дерева, в котором объединена структура объектов системы со структурой базы данных.

Реализацию задания разделим на четыре этапа:

1. Изменение структуры информационной базы и БД.
2. Вывод новых объектов в главное меню.
3. Составление диалогов.
4. Формирование вычисляемых полей.

Для создания новых объектов, выбирается соответствующий раздел дерева: **Справочники, Документы, или Журналы**. Затем с помощью контекстной команды (правая кн. мыши или раздел «Действия» главного меню) «Новый <объект>» создается требуемое.

Сначала создаются справочники, потом журналы, потом документы

Если запускается окно конструктора, снять флаг использовать конструкторы, закрыть окно конструктора и повторить действие.

В открывшемся окне в рамках задания важно указать для всех объектов : идентификатор (системное имя объекта) и синоним (пользовательское имя объекта). Для нового документа также нужно указать нужный журнал, в котором будут храниться документы.

Затем, выбрав этот созданный объект, выполняем контекстную команду «Новый реквизит» для задания вводимых реквизитов. Для новых реквизитов важно указать : идентификатор, синоним и тип значения.

Изменить параметры реквизита можно через контекстную команду «Свойства».

Записать изменения можно командой «Записать» главного меню.

Следующим шагом, в окне «Конфигурация» выберем закладку (внизу) «Интерфейсы». В списке интерфейсов выберем интерфейс «Бухгалтер». Выполнив контекстную команду «Редактировать меню» переходим в редактор главного меню.

Добавление объектов осуществляется путем выполнения контекстной команды свойства на элементе <новый элемент> в соответствующем разделе редактора. При задании свойств необходимо задать Наименование и Объект (выбрать нужный объект из списка). Для документа нужно также указать команду «Ввести».

Теперь рассмотрим, как создать диалоговые окна для созданных объектов. Выбрав необходимый объект в дереве, выполняется контекстная команда «Редактировать форму» и открывшийся макет формы вставляются все созданные реквизиты.

Далее можно добавлять реквизиты в форму с помощью кнопки с желтым крестом в правой нижней части экрана. Текстовые поля добавляются с помощью кнопки с изображением **T**.

ЗАДАНИЕ НА РАБОТУ

1. Создать в программе MS Access следующие объекты для реализации учета пробега автотранспорта:

1) Новый справочник «Автомобили» с полями:

- Марка автомобиля;
- Гос. номер автомобиля;
- Год выпуска;
- Норма расхода литров на 1 км.

2) Новый справочник «Водители» с полями:

- Сотрудник – типа спр. «Сотрудники»;
- Автомобиль – типа спр. «Автомобили».

3) Новый документ «Путевой лист» с полями:

- Водитель. Выбирается из списка.
- Автомобиль. Отображается при выборе водителя.
- Время выезда. Вводится.
- Время заезда. Вводится.
- Начальный километраж. Вводится.
- Конечный километраж. Вводится.
- Пробег. Вычисляется как разница между километражами:
[конечный километраж] – [начальный километраж].
- Расход топлива. Вычисляется как произведение пробега и нормы расхода топлива:

$$[\text{пробег}] \cdot [\text{норма расхода литров на 1 км}].$$

2. Должно иметься следующее:

1) Один автомобиль, у которого два водителя;

- 2) У одного водителя два автомобиля;
- 3) Для водителей подсчитывать пробег;
- 4) Для автомобилей подсчитывать расход топлива;
- 5) У водителя с двумя автомобилями пробег должен быть больше остальных;
- 6) У автомобиля с двумя водителями расход топлива должен суммироваться для обоих водителей;

3. Создать диаграмму по водителям, где:

- 1) Средний пробег по всем водителям представлен в виде горизонтальной красной линии;
- 2) Пробеги остальных водителей – столбики синего цвета;
- 3) Если меняется среднее или пробег, то меняется соответствующий столбик или красная линия;

4. Создать диаграмму по автомобилям, где:

- 1) Средний расход топлива представлен в виде горизонтальной красной линии;
- 2) Расходы топлива автомобилей представлены в виде столбиков синего цвета;
- 3) Если меняется средний расход топлива или что-то в отдельности, то меняется соответствующее отображение на графике.

Добавить дополнительные требования – какие?

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Перечислите последовательность действий и существенные параметры для создания нового справочника в конфигурации программы «1С Предприятие 7.7».

2. Перечислите последовательность действий и существенные параметры для создания нового документа.

Лабораторная работа №10. Создание ИС на основе типовых решений. Создание дополнительных бухгалтерских отчетов

(3 варианта)

Цель работы: Изучить основы создания дополнительных отчетов в структуре предметно-ориентированной ИС, на примере программы 1С Предприятие 7.7.

Содержание работы:

1. Изучить теоретические сведения.
2. Реализовать изменения в ИС согласно заданию.

ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Система 1С Предприятие является эффективным инструментом для бухгалтеров и управленцев. Типовые конфигурации поставляемые с системой обладают очень широким спектром возможностей. Однако на каждом предприятии существуют некоторые особенности ведения учета документооборота и отчетности, которые не реализованы в типовых конфигурациях. Поэтому необходимо уметь добавлять дополнительные функции.

При использовании системы 1С Предприятие часто возникает необходимость добавить новый отчет или преобразовать данные стандартного отчета для представления в заданном виде.

Для облегчения этой задачи пользователям не владеющим внутренним языком 1С Предприятие 7.7 в системе предусмотрены специальные «конструкторы»- пошаговые интерактивные алгоритмы для создания простейших отчетов.

Для запуска конструктора бухгалтерских запросов необходимо создать новый отчет через конструктор отчетов: команда «Новый отчет» в разделе главного меню «Конструкторы» в режиме «Конфигуратор».

На шаге выбора конструктора структуры отчета (выбор конструктора макета отчета, конструктора запроса или конструктора бухгалтерского запроса) отметить «конструктор бухгалтерского запроса».

Данный конструктор позволяет составить отчет, содержащий бухгалтерский запрос и шаблон печатной формы по заданию.

После завершения создания конструктором отчета, необходимо отредактировать шаблон печатной формы для придания отчету требуемого вида.

ЗАДАНИЕ НА РАБОТУ

Добавить в конфигурацию отчеты согласно требованиям варианта.

Варианты заданий.

1. Составить отчет «Остатки товаров». Данный отчет должен выводить перечень товаров с указанием общего количества и стоимости товаров на всех складах (без детализации по складам) на заданное число с выводом общей суммы.
2. Составить отчет «Подотчетники». Данный отчет должен выводить список сотрудников с указанием суммы денежных средств полученных под отчет каждым сотрудником на заданное число, с выводом общей суммы.
3. Составить отчет «Издержки организации». Данный отчет должен выводить список статей издержек суммы величины каждой издержки за выбранный период.

Замечание. Все перечисленные варианты заданий имеют базовую структуру типа «Оборотка».

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Какие на Ваш взгляд имеют преимущества и недостатки использование шаблонов при создании новых бухгалтерских отчетов в системе 1С Предприятие 7.7.

Лабораторная работа №11. Проектирование элементов Case систем

(6 вариантов)

Цель работы: Изучить принцип функционирования и основы создания Case систем проектирования ПО.

Содержание работы:

1. Изучить теоретические сведения.
2. Разработать приложение согласно заданию.

ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Case системы являются мощным средством автоматизации проектирования ПО. Основной принцип таких систем заключается в визуальном конструировании структуры разрабатываемого ПО путем комбинирования готовых визуальных объектов в рабочем окне. Каждому визуальному объекту соответствует некоторый модуль или фрагмент исходного текста программы.

После того как завершено визуальное конструирование программы пользователь Case системы активирует процедуру генерации программного кода, в результате работы которой получается практически готовая программа, соответствующая заданной структуре.

Принцип Case моделирования используется не только при разработке ПО, но и при создании сложных математических моделей с использованием численных методов в математических программах и многих других областях.

ЗАДАНИЕ НА РАБОТУ

Разработать мини Case систему проектирования программ для реализации математических моделей. Система должна позволять моделировать функцию:

$$y = F_1(F_2(F_3(x)))$$

где F_i – это функция из набора:

$$\{y = \sqrt{x}; y = \frac{1}{x}; y = e^x\}$$

Разработанная программа должна позволять осуществлять визуальное комбинирование последовательности связи из экземпляров имеющихся в программе объектов: входом, выходом, и тремя видами вышеуказанных функций.

Программа должна содержать кнопку и механизм генерации исходного кода на языке VBA. В результате выполнения данного механизма, должен быть сгенерирован работоспособный исходный код программы для вычисления заданного в разработанной программе выражения.

Программа должна считать функцию от функции.

При вводе значений, не попадающих в область определения функции, должно выводиться сообщение об этом.

исправить язык синего текста на более читабельный и понятный? что должны иметься кнопки, каждая из которых делает то-то

КОНТРОЛЬНЫЕ ВОПРОСЫ

2. Укажите назначение и преимущества Case – систем проектирования.