

Методические рекомендации по выполнению лабораторных работ по дисциплине «Эксплуатация и сопровождение ИС и сервисов»

ЛАБОРАТОРНАЯ РАБОТА №1

Тема: «Установка 1С Предприятие. Создание информационной базы»

Цель: научить создавать собственную информационную базу для работы, ознакомиться с интерфейсом программы и способами работы с основными объектами системы; сформировать умения по созданию пользователей системы, разграничению ролей. Ознакомить с приемами, применяемыми на начальном этапе ведения учета в 1С:Бухгалтерии.

1. Установка конфигурации

Практическое задание №1

Добавьте в список информационных баз свою собственную, назовите ее своим именем и фамилией.

Выполнение:

Пуск > Все программы > 1С Предприятие 8.1. > 1С Предприятие (рис. 1.1) > Добавить (рис. 1.2) > Добавление в список уже существующей базы/ Далее, это позволяет вам добавить в окно запуска программы существующую базу.

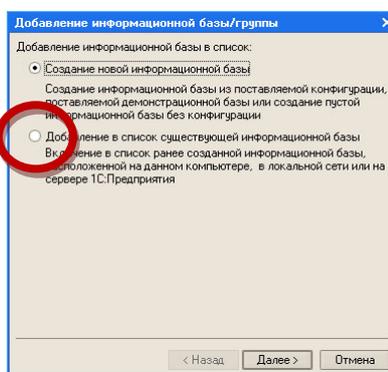
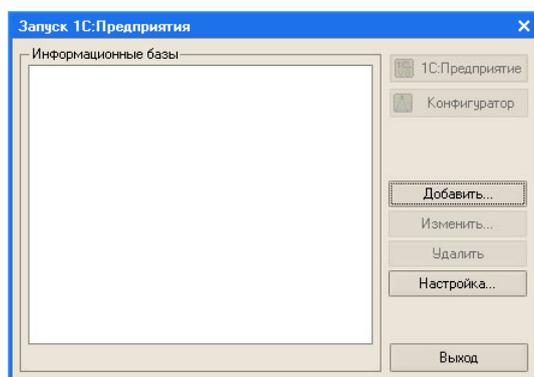


Рис. 1.1. Окно запуска программы Рис. 1.2. Выбор добавления новой информационной базы

Выбрав нужную базу из списка и нажав в очередной раз **Далее** мы попадаем в окно выбора названия для базы и типа ее расположения (рис. 1.3).

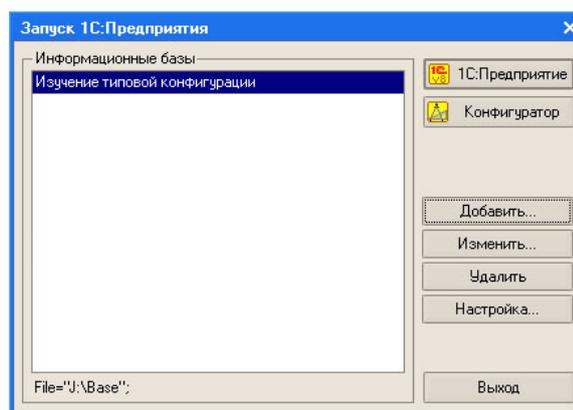
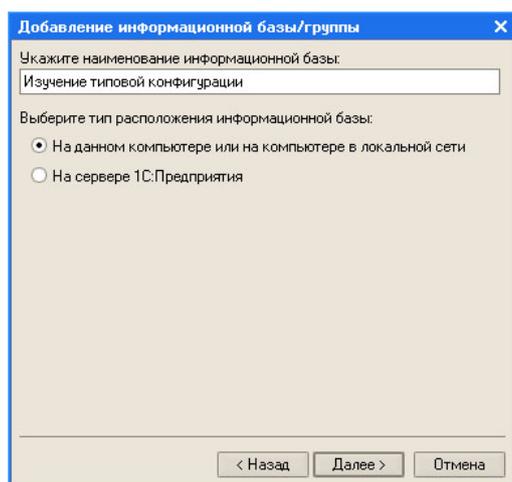


Рис. 1.3. Выбор названия и типа расположения базы Рис. 1.4. Стартовое окно после добавления информационной базы

В качестве названия базы введите "Изучение типовой конфигурации", а тип расположения выбрали **На данном компьютере или на компьютере в локальной сети/ Далее** (указать каталог для информационной базы - **адрес базы в сети на доске**) > Готово.

Теперь, если выделена та его строка, которая соответствует только что созданной информационной базе (рис. 1.4), в нем станут активными кнопки, позволяющие работать с этой базой (**Добавить, Изменить, Удалить, Настройки**).

Первый запуск

При первом запуске информационной базы программа предлагает произвести ее первоначальное заполнение.

Конфигурация производит некоторые начальные действия: установки значений констант, начальное заполнение некоторых справочников (в основном нормативно-справочной информацией).

1. Ввод сведений об организации

Практическое задание № 2

Необходимо ввести основную информацию об одной из организаций предприятия:

Наименование	«Лидер»
Префикс	Л
Полное наименование	Общество с ограниченной ответственностью «Лидер»
ИНН	7730123456
КПП	220505001
Сведения об основном расчетном счете	
Наименование	Основной р/с
Банк	АКБ «АВТ-БАНК»
Номер расчетного счета	40702810500000000125
Вид	Расчетный
Корр. счет	30101810100000000054
БИК	049250054
Валюта счета	Руб.
Адрес	Набережные Челны, пр. Мира, 84
Телефон	34-58-01

Выполнение:

Перед началом ведения учета необходимо заполнить сведения об организации (меню **«Предприятие – Организации»**):

1. Откройте форму для ввода основных сведений о своей организации. Для этого щелкните дважды по строке «Наша организация» (или нажмите кнопку , или выберите меню **«Действия – Изменить»** в командной панели формы списка).

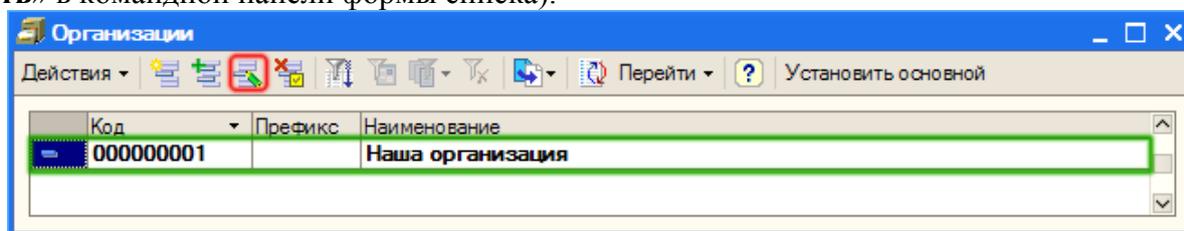


Рис. 1. Справочник Организации

В открывшемся окне заполните основные сведения об организации так, как показано на рисунке.

Концепция программы «1С: Бухгалтерия» подразумевает раздельное ведение учета по каждой из организаций, входящих в состав предприятия, в единой информационной базе. Все ключевые сведения, как о предприятии, так и об организациях, содержатся в справочниках, либо в регистрах сведений, сгруппированы в пункте **«Предприятие»**.

Рис. 2. Ввод сведений о собственной организации

В качестве организации программа «1С: Бухгалтерия 8.1» позволяет указать не только юридическое лицо, но и физическое лицо, в этом случае несколько изменяется состав реквизитов экранной формы справочника «Организации». После ввода наименования организации (краткого и полного), префикса нажать на кнопку в поле «Осн. банковский счет», в открывшемся окне «Банковские счета» нажать кнопку **Добавить** и заполнить сведения об основном расчетном счете по практическому заданию №1:

Рис. 3. Ввод сведений о банковском счете

Ввести сведения о банке, нажав на кнопку в поле «Банк»:

Рис. 4. Ввод сведений о банке

Для начала ведения учета необходимо заполнить еще несколько справочников, непосредственно связанных со структурой организации.

Практическое задание №3

Введите информацию о подразделениях и складах организации «Лидер»:

Список «Подразделения организации»	Производственный цех
	Главный склад

Выполнение:

1. Пункт меню «Предприятие – Подразделения организаций» – кнопка **Добавить**:

Рис. 5. Список подразделений

2. Пункт меню «Склад – Склады (места хранения)» – кнопка **Изменить**

В справочник «Склады» следует ввести главный склад. При создании нового склада программа требует обязательно указать вид склада, выберите «Оптовый». Тип цен указывать не надо.

Код	Наименование	Вид склада	Тип цен роз
000000001	главный склад	Оптовый	роз

Рис. 6. Справочник Склады (места хранения)

2. Ввод сведений о пользователях системы

Перед началом работы всем пользователям необходимо присвоить имя и пароль для разграничения прав доступа к отдельным объектам информационной базы, а также прав доступа к каждой организации, по которой в информационной базе ведется учет. Для ввода нового пользователя необходимо закрыть программу в режиме ведения бухгалтерского учета и запустить программу в режиме конфигурирования.

Меню «**Пуск – 1С Предприятие – Конфигуратор**» и через пункт меню «**Администрирование – Пользователи**» – добавить список пользователей, работающих с системой, выполняя практическое задание №4.

Практическое задание №4

Введите себя в качестве пользователя системы. По желанию установите пароль, роль – «Полные права», интерфейс – «Полный», рабочий язык «Русский». Закройте конфигуратор. Сохраните изменения и запустите «1С: Предприятие 8.1» в режиме учета.

Выполнение:

1. Закройте программу 1С: Предприятие. Бухгалтерия предприятия.

2. Откройте программу 1С: Предприятие в режиме конфигуратора:

Меню «**Пуск – Программы – 1С: Предприятие – Конфигуратор**».

3. Для добавления нового пользователя открыть меню «**Администрирование – Пользователи**» – **Добавить**, в окне «Пользователи» нажать **Добавить** и во вкладке Основные ввести имя пользователя (краткое и полное) и пароли (в учебной версии пароль не вводить):

Во вкладке «Прочие» подключить доступные роли «Полные права», Основной интерфейс – «Полный» и язык – «Русский»:

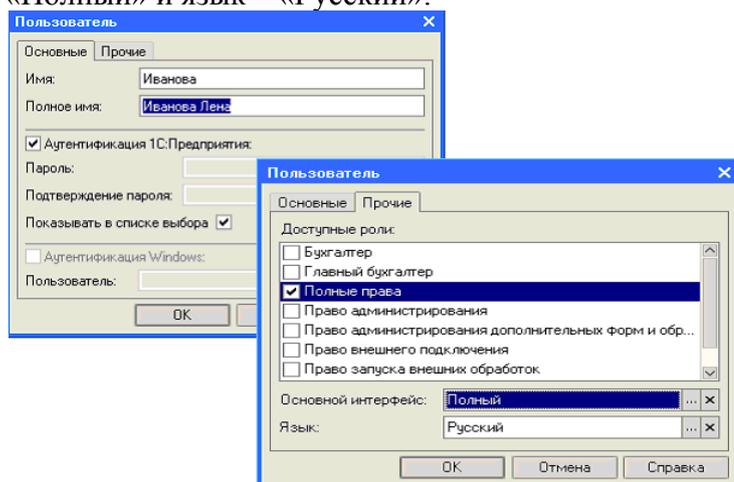


Рис. 7. Добавление пользователя системы

«1С: Бухгалтерия» позволяет одному пользователю установить несколько ролей, имеющих в конфигурации, и одна и та же роль может быть установлена для разных пользователей. Помимо ролей в конфигураторе назначается основной интерфейс. Интерфейс является объектом конфигурации, в котором определяются наборы команд главного меню и панелей инструментов. Результат выполнения на рис.12:

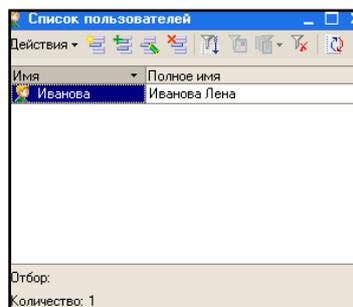


Рис. 8. Список пользователей системы

4. Закройте конфигуратор и откройте программу в режиме бухгалтерского учета.

После того, как Бухгалтерия *предприятия* запущена в пользовательском режиме, прежде чем откроется окно программы, мы увидим окно запроса пароля.

После первого *входа в систему* с новым именем и паролем, *пользователь*, которого идентифицируют данное имя и пароль, будет внесен в справочник **Пользователи** (**Операции > Справочники > Пользователи**).

Посмотреть список активных *пользователей* можно, выполнив команду **Сервис > Активные пользователи**.

Пункт меню **Сервис > Журнал регистрации** хранит сведения о действиях *пользователя*.

Проверочное задание: Создайте нового *пользователя* системы, назовите его **Бухгалтер**. В качестве роли установите "Бухгалтер (основная роль)", в качестве основного интерфейса - "Бухгалтерский", язык - русский. Проверьте работоспособность нового *пользователя*, войдя в систему под его именем. Попробуйте выполнить команду **Сервис > Журнал регистрации**. Если все сделано верно - система не даст вам выполнить эту команду, выведя сообщение о недостаточности прав доступа.

Практическое задание №5

Для введенного в режиме конфигурирования *пользователя* установите следующие *параметры*:

Основная организация	Лидер ООО
Основная ставка НДС	18%
Основной ответственный	(указать пользователя)
Основной склад	Главный склад
Запрашивать подтверждения при закрытии программы	Да
Показывать в документах счета учета	Да

Выполнение:

Меню «**Сервис – Настройки пользователя**» ввести предложенные параметры:

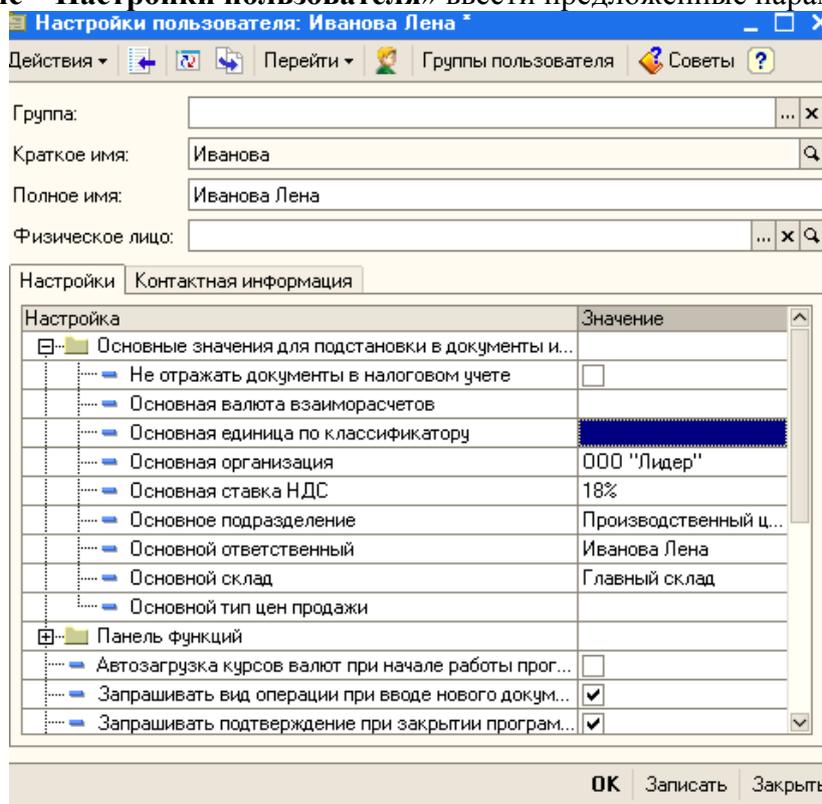


Рис. 9. Установление параметров для пользователя

4. Настройка параметров учетной политики организации

Согласно законодательству РФ, организация может выбрать один из способов отражения в учете тех или иных операций, предусмотренных нормативными документами, пользователю предоставляется возможность осуществить настройку системы, указав соответствующие параметры

в регистрах сведений «Учетная политика». Доступ к регистрам сведений осуществляется через пункт меню «**Предприятие – Учетная политика – Учетная политика организаций**».

Практическое задание №5

Введите параметры учетной политики для организации «ООО Лидер» за 2014 г. (данные, которые отсутствуют в форме ввода, но имеются в таблице можно не вводить, т. к. они используются для разных релизов программы «1С Бухгалтерия»):

Для целей бухгалтерского учета:	
Способ оценки МПЗ (Запасы)	ФИФО
ОС и НМА	Линейный
Общие сведения	Система налогообложения -общая
Способ учета выпуска готовой продукции	Без использования счета 40
Способ расчета себестоимости	По подразделениям
Отключите «использовать метод директ-костинг»	
Способ оценки товаров в рознице	По стоимости приобретения
Для целей налогового учета:	
Способ оценки МПЗ	ФИФО
Налоговый период	Квартал
Метод определения налоговой базы НДС	По отгрузке
Сначала оплачиваются ценности, по которым НДС	Может быть принят к вычету
Порядок учета расходов по налогам с ФОТ	На счетах, соответствующим счетам бухгалтерского учета

Выполнение:

Данный этап является одним из основных этапов настройки и продолжением настройки параметров учетной политики. Неправильно произведенная настройка аналитического учета и ее последующее изменение в процессе ведения учета может отрицательно сказаться на работоспособности системы. Для выполнения настройки выбрать пункт меню «**Предприятие – Настройка параметров учета**»

Необходимо заполнить правильно вкладки:

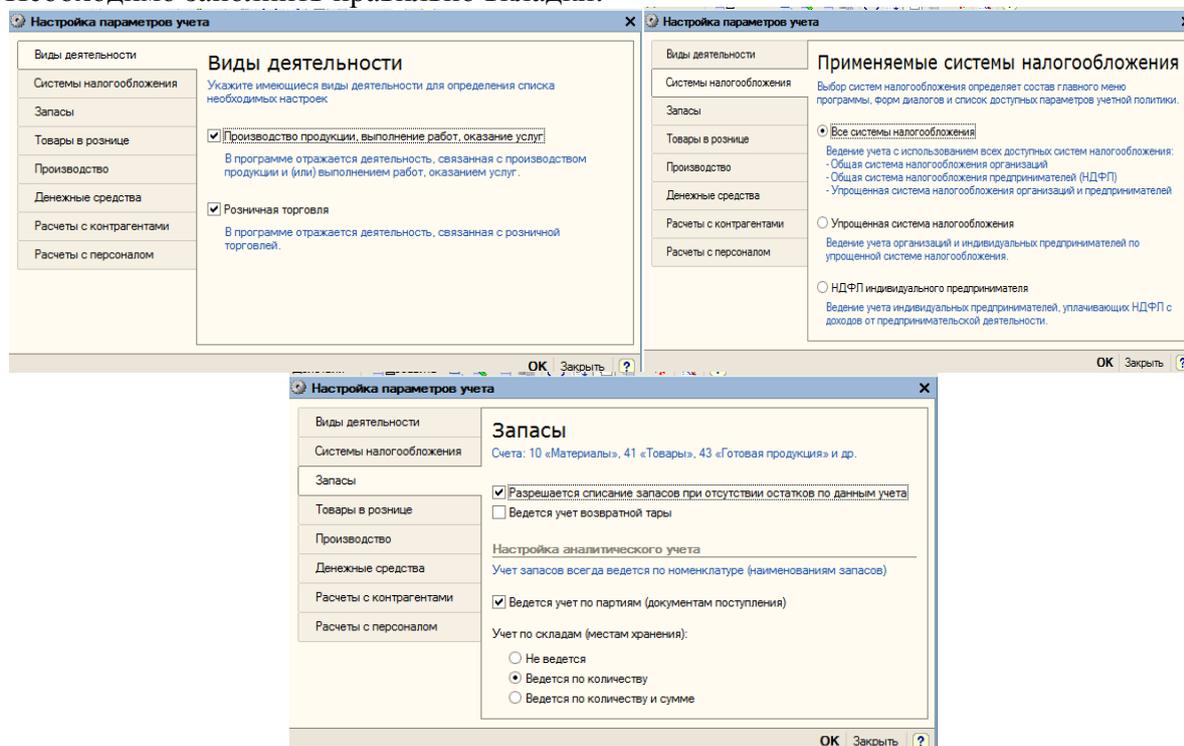


Рис. 10. Настройка параметров учета. Виды деятельности. Система налогообложения. Запасы

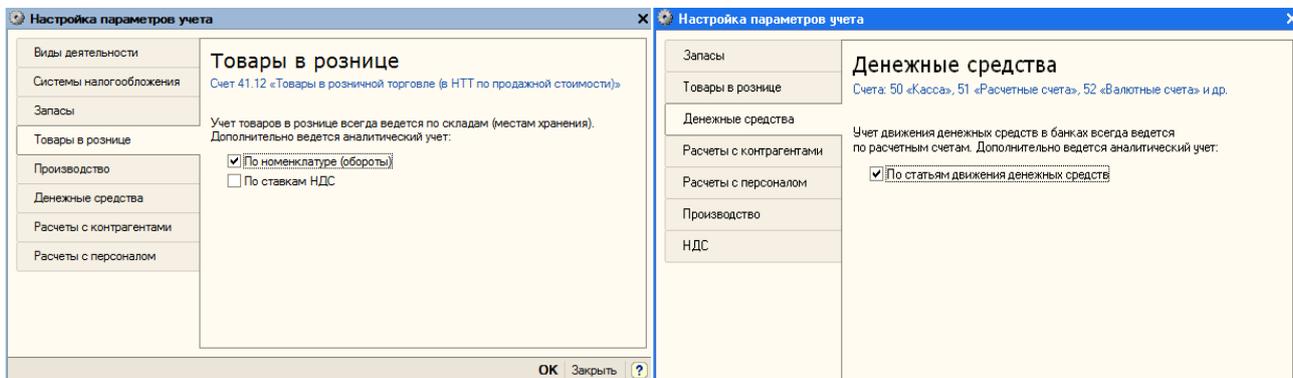


Рис. 11. НПУ Товары в рознице .НПУ Денежные средства

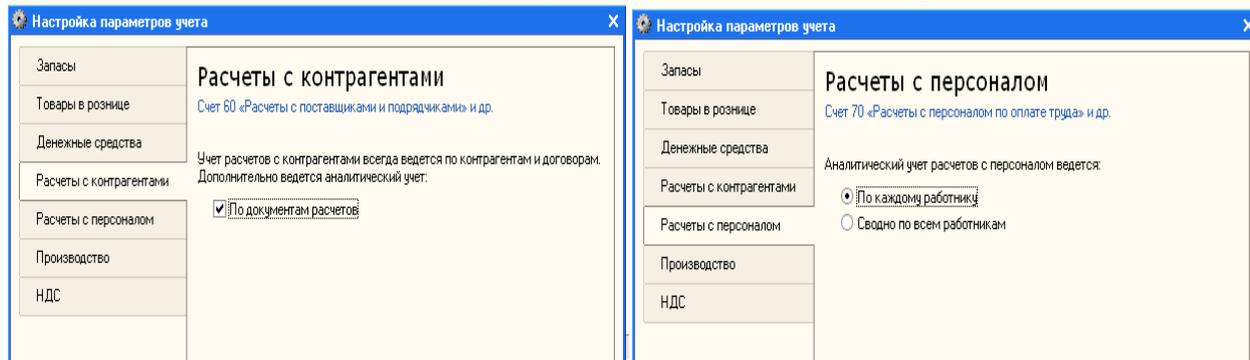


Рис. 121. НПУ Расчеты с контрагентами НПУ Расчеты с персоналом

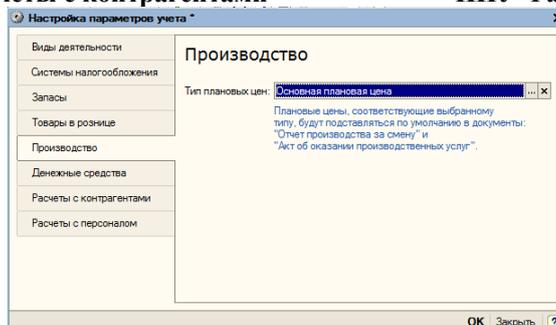


Рис. 132. Настройка параметров учета. Производство

Настройки являются общими для всех организаций информационной базы.

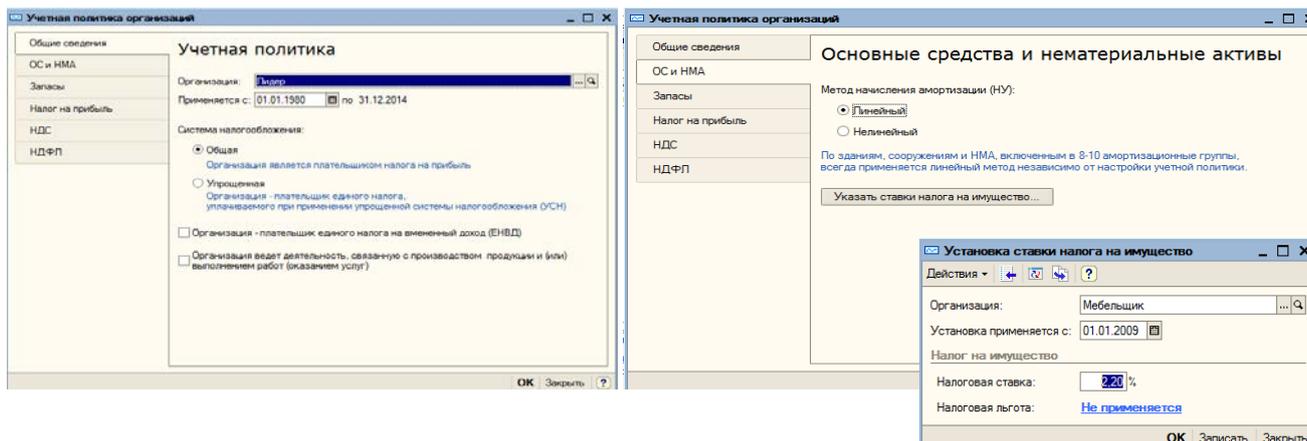
6. Учетная политика для целей налогового и бухгалтерского учета

Предприятие > Учетная политика > Учетная политика организаций позволяет открыть *реестр сведений Учетная политика организаций*. Это периодический реестр - при изменениях *учетной политики* создается новая запись, которая содержит дату начала действия новой *учетной политики*.

Практическое задание №6

Заполните *учетную политику* вашей организации по образцу (рис.23)

Выполнение:



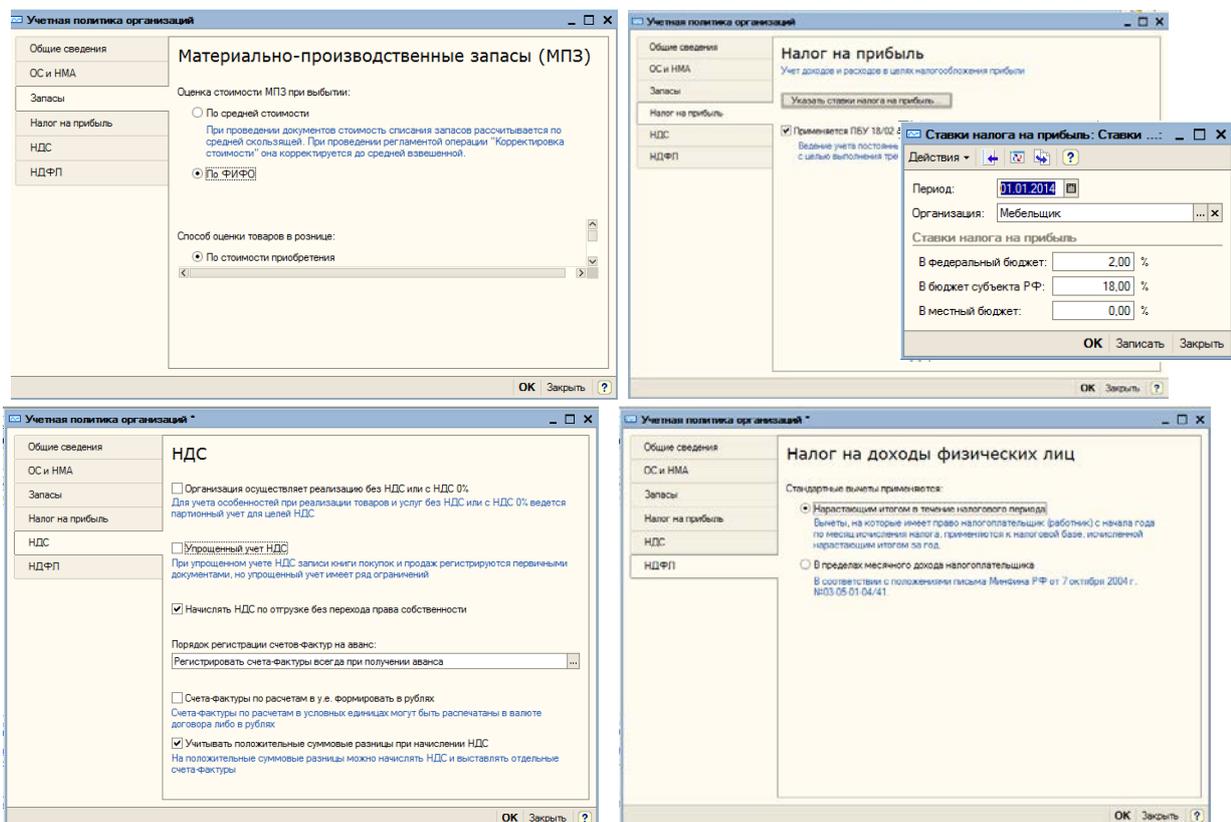


Рис 23. Учетная политика организаций

Учетная политика (по персоналу)

Предприятие > Учетная политика > Учетная политика (по персоналу). Откроется окно регистра сведений **Учетная политика организации по персоналу**. Создадим новый элемент регистра ([рис. 24](#)).

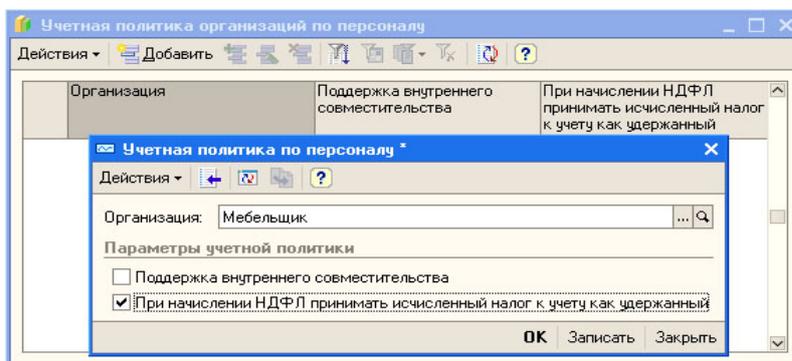


Рис. 24. Настройка учетной политики организации по персоналу

Несмотря на немалый объем проделанных работ, первоначальная подготовка базы к ведению учета еще не закончена. В следующей работе мы займемся завершением подготовки базы к решению учетных задач.

Контрольные вопросы

1. Алгоритм создания собственной информационной базы.
2. Алгоритм создания списка пользователей системы.
3. В каком справочнике хранится информация о пользователях системы, как открыть данный справочник?
4. Как посмотреть список активных пользователей?
5. Где хранятся сведения о действиях пользователя?
6. Где можно изменить дату в программе 1С:Бухгалтерия?
7. Опишите алгоритм удаления объекта из системы.

ЛАБОРАТОРНАЯ РАБОТА № 2

Тема: «Администрирование 1С Предприятие. Администрирование пользователей. Резервное копирование и восстановление данных. Обновление системы»

Цель: сформировать умения по выполнению роли администратора информационной системы. Научить производить администрирование пользователей, выполнять резервное копирование и восстановление данных и обновление системы.

1.2. Основные возможности и особенности внедрения

Программа 1С:Бухгалтерия предназначена для ведения бухгалтерского и *налогового учета*, для проведения расчетов по *заработной плате* и формирования регламентированной *бухгалтерской отчетности*. Учетные механизмы, заложенные в программу, соответствуют бухгалтерскому и налоговому законодательству, принятому в Российской Федерации. Программа поддерживает общую систему налогообложения, налогообложение по УСН и ЕНВД.

1С:Бухгалтерия применяется для *автоматизации* учета в самых разных организациях. Например, в справочнике внедренных решений, который можно найти по адресу <http://1c.ru/rus/partners/solutions/>, есть примеры множества внедрений программы. Для того чтобы получить представление о круге задач, для которых программа подходит лучше всего или понять, сможете ли вы эффективно автоматизировать с ее помощью учет в вашей организации - вы можете посмотреть этот справочник. Если окажется, что достаточно много организаций вашей направленности автоматизированы с помощью 1С:Бухгалтерии - это верный признак того, что программа, скорее всего, подойдет и вам. Ну, а обилие других *конфигураций* в списке внедрений обычно однозначно указывает на то, что вам для *автоматизации* лучше всего подыскать одну из таких *конфигураций*.

Обычно если организацию в целом устраивают возможности программы, ее устанавливают либо без изменений, либо с незначительными настройками вроде добавления дополнительных счетов в *план счетов*, изменения форм и логики работы некоторых документов, создания дополнительных и переработки существующих отчетов. Если же организация отличается особенной спецификой ведения учета, нередко учет в ней автоматизируют с помощью других *конфигураций*, работающих на *платформе* 1С:Предприятие. Например, такое часто встречается в ресторанном и гостиничном бизнесе, а так же во многих других видах деятельности.

Разработчик поддерживает 1С:Бухгалтерию в актуальном состоянии - то есть, регулярно обновляя ее, вы можете быть уверены в том, что учет в вашей организации соответствует последним изменениям в законодательстве.

Бухгалтерский учет, несмотря на достаточно жесткое и подробное нормативное регулирование, допускает множество "вольностей", особенностей. Если вы занимались курсом "Основы бухгалтерского учета" - вы должны помнить, что в Российской Федерации существует четыре уровня нормативного регулирования учета, один из которых - это стандарты самих организаций. Особенности учета в каждой из организаций являются причиной неоднозначности при выборе способа *автоматизации* учета с помощью 1С:Бухгалтерии.

Если вы начинаете вести учет в новой организации, не имеющих "учетных традиций" и собираетесь применять для своих целей 1С:Бухгалтерию, наиболее быстрым, логичным и дешевым способом *автоматизации* учета будет именно использование типовой *конфигурации*, для небольшого предприятия вполне подойдет базовая версия, хотя ее ограничения достаточно сильно сужают круг ее применения. Воспользовавшись типовой *конфигурацией*, вы получите готовое решение для *автоматизации* учета за вполне прогнозируемую сумму, равную стоимости программы, стоимости подписки на обновления, и, скорее всего, стоимости работы специалистов, которые установят систему и произведут ее первоначальную настройку. Как правило, организации, автоматизирующие учет, заключают договора на обслуживание программ с организацией-партнером 1С (1С - Франчайзи), которая занимается обновлением системы, консультированием бухгалтеров по сложным вопросам учета и другими подобными вопросами.

Если же вы собираетесь автоматизировать учет в существующей организации, а в особенности - если это организация достаточно большая, имеющая какие-либо особенности в ведении учета, которые не предусмотрены типовой *конфигурацией*, вам предстоит встать перед нелегким выбором. Выбрав автоматизацию учета с помощью типовой *конфигурации*, вы принудите организацию отказаться от ее "учетных традиций" и вести некий "усредненный" вариант учета. Такой выбор, как уже было сказано, довольно экономичен, но нередко он

неприемлем и приводит ко второму варианту - то есть - к использованию измененной *конфигурации*. У подхода к *автоматизации*, предусматривающего изменение типовой *конфигурации*, есть свои плюсы и минусы. Плюсы заключаются в том, что программу можно настроить так, как это нужно вашей организации. А минусы - в дополнительных затратах на настройку и в дополнительных сложностях с обновлением программы. Для настройки *конфигурации* под ваши нужды можно либо воспользоваться услугами специализированных организаций - обычно это уже упомянутые выше 1С - Франчайзи - фирмы, которые заключают договор с 1С и занимаются автоматизацией учета, поддержкой пользователей и массой других дел, либо принять на работу в организацию программиста, способного грамотно настроить программу. Еще один минус заключается в том, что для обновления измененной *конфигурации* вам так же понадобятся услуги квалифицированного программиста.

Доработка типовой *конфигурации* под учетные нужды конкретной организации - это распространенная практика. Например, как быть, если документы, принятые организацией для оформления каких-либо фактов хозяйственной жизни, отличаются от типовых, если вам нужны особые отчеты, если вы ведете учет по некоторым счетам в нескольких *аналитических разрезах*, количество которых превышает стандартные три, заложенные в *конфигурации*? Ответов на эти вопросы может быть несколько. Первый - дорабатывать типовую *конфигурацию* и быть готовыми к дополнительным затратам на поддержание ее в актуальном состоянии. Второй - подыскать другую *конфигурацию*, которая способна удовлетворить ваши нужды. Причем, не исключено, что в вашем случае неплохим решением будет работа в нескольких *конфигурациях* - в одной, например, для ведения бухгалтерского учета, а в другой - для целей расчета заработной платы.

Сейчас у вас может возникнуть вполне закономерный вопрос о том, есть ли практическая польза тем бухгалтерам, которые собираются работать в измененных *конфигурациях* 1С:Бухгалтерии, от изучения особенностей типовой *конфигурации*. Ответ на этот вопрос вполне однозначен: да, есть. Всё дело в том, что и в типовой, и в измененной *конфигурациях*, вам придется работать с одними и теми же объектами системы - документами, справочниками, отчетами, регистрами, *журналами документов* и т.д. И если вы разобрались с функционированием типовой *конфигурации* и в состоянии вести с ее помощью учет - вы сможете освоить измененную *конфигурацию* в достаточно короткие сроки.

А сейчас мы приступаем к практической работе с программой. Начиная работу, мы исходим из того, что у вас уже установлена *платформа* и необходимый шаблон *конфигурации*.

1.3. Установка конфигурации из шаблона

После того, как программа установлена, в *USB*-порту ПК размещен электронный ключ, дающий возможность работы с программой, вы можете приступать к ведению учета. Но прежде чем начинать работу, вам понадобится произвести еще некоторые манипуляции. В частности - создать новую *информационную базу*. Для этого вам нужно познакомиться с окном запуска программы, которое появляется после выполнения команды (в *Windows XP*) Пуск > Все программы > 1С Предприятие 8.1. > 1С Предприятие ([рис. 1.1](#)).

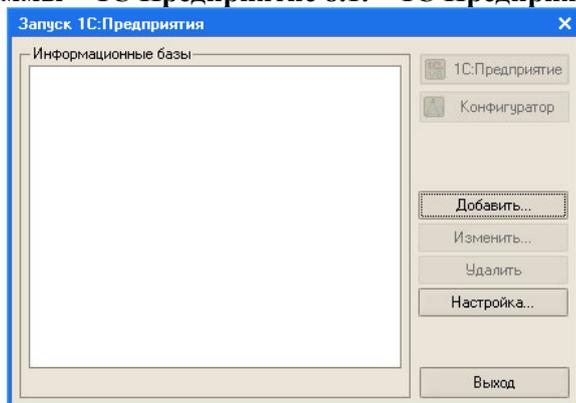


Рис. 1.1. Окно запуска программы

На рисунке это окно пустое. Для того, чтобы создать новую *информационную базу*, нужно нажать на кнопку **Добавить**. После этого появится окно ([рис. 1.2](#)) с вопросом о том, хотите ли вы добавить новую базу или добавить в список существующую.

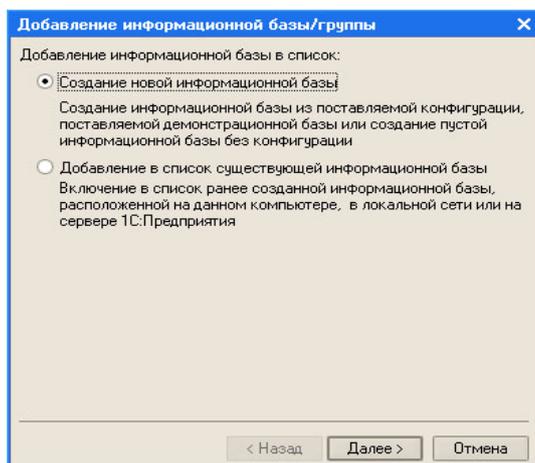


Рис. 1.2. Выбор добавления новой информационной базы

В нашем случае нужно выбрать пункт **Создание новой информационной базы** и нажать кнопку **Далее**. Пункт **Добавление в список существующей информационной базы** позволяет вам добавить в окно запуска программы существующую базу. Это может быть нужным тогда, когда вы, например, работали с некоторой базой данных в одном месте и хотели бы продолжить работу с ней в другом. Для этого вам нужно скопировать каталог *информационной базы*, перенести его на ваш компьютер, после чего воспользоваться данным пунктом этого окна.

Возможность подключения *информационных баз* может понадобиться вам и в других случаях.

Нажав кнопку **Далее**, мы попадаем в следующее окно добавления *информационной базы* ([рис. 1.3](#)).

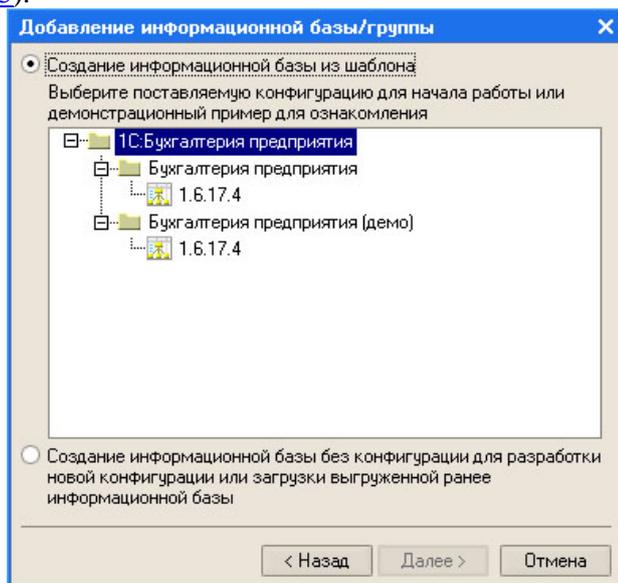


Рис. 1.3. Окно создания информационной базы

Здесь нас интересует вариант **Создание информационной базы из шаблона**. В списке шаблонов можно найти множество групп шаблонов в зависимости от того, какие из них установлены в вашей системе, а открыв группу - выбрать интересующую вас *конфигурацию*.

Если у вас есть конфигурация, которая представляет собой отдельный файл с расширением .CF, вы должны создавать новую информационную базу немного другим путем, подробности о котором мы рассмотрим ниже.

Выбрав нужную базу из списка и нажав в очередной раз **Далее** мы попадаем в окно выбора названия для базы и типа ее расположения ([рис. 1.4](#)).

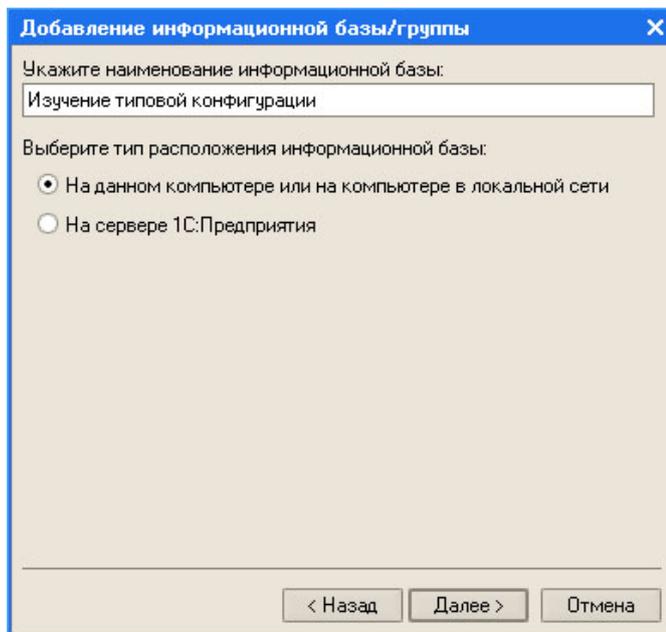


Рис. 1.4. Выбор названия и типа расположения базы

В качестве названия базы мы ввели "Изучение типовой *конфигурации*", а тип расположения выбрали **На данном компьютере или на компьютере в локальной сети**. Сетевая версия программы, использующая для хранения баз данных сервер 1С:Предприятия допускает выбор соответствующего варианта.

Очередное нажатие на **Далее** приводит нас к окну, где нужно указать каталог для *информационной базы* ([рис. 1.5](#)).

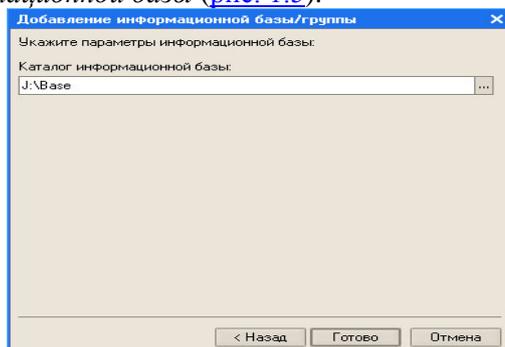


Рис. 1.5. Выбор каталога и языка

Теперь осталось лишь нажать на кнопку **Готово** и через некоторое время, необходимое на создание *информационной базы*, вы снова увидите стартовое окно. Теперь, если выделена та его строка, которая соответствует только что созданной *информационной базе* ([рис. 1.6](#)), в нем станут активными кнопки, позволяющие работать с этой базой.

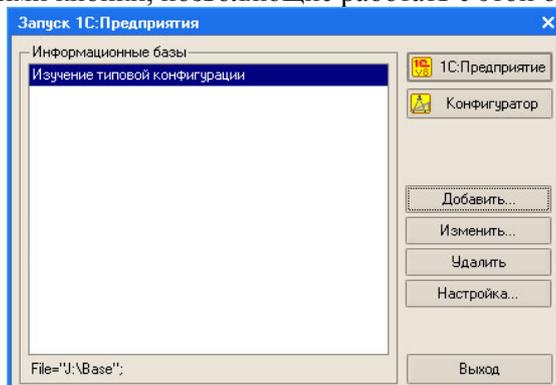


Рис. 1.6. Стартовое окно после добавления информационной базы

Кнопка *Конфигуратор* предназначена для *запуска системы* в режиме *конфигурирования*. В данном курсе мы коснемся вопросов *конфигурирования* лишь в небольшом объеме - это отдельная тема, которой будут посвящены особые курсы.

Кнопка **1С:Предприятие** запускает программу в пользовательском режиме - именно он нас и интересует.

Обратите внимание на то, что при выделении *информационной базы*, находящейся в списке, в нижней части окна можно видеть путь, по которому эта база расположена.

Если вы хотите удалить базу из списка - воспользуйтесь кнопкой **Удалить**. Однако, удаляя базу из списка вы должны понимать, что вы не удаляете файлы базы данных, расположенные на диске - то есть - если вам действительно нужно удалить базу с компьютера (скажем, вы временно поработали со своей базой на чужом компьютере и хотите удалить ее оттуда) - вы должны найти и удалить каталог базы и все файлы, содержащиеся в нем. База данных 1С:Бухгалтерии довольно объёмна - после создания размер папки с ней равен нескольким сотням *мегабайт*. При ведении учета размеры базы растут с вводом учетных данных.

Кнопка **Изменить** позволяет изменить параметры выделенной *информационной базы* или группы.

Кнопка **Настройка** предназначена для настройки диалогового окна запуска программы (рис. 1.7).

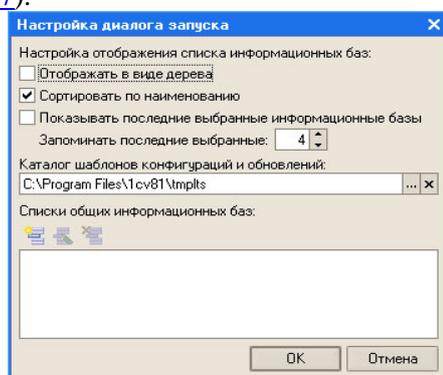


Рис. 1.7. Настройка параметров окна запуска программы

Если установить в окне настройки флажок **Отображать в виде дерева**, список баз в окне запуска программы будет отображаться в виде, напоминающем отображение папок в Проводнике Windows. Записи об *информационных базах* могут принадлежать группам, группы можно создавать с помощью стандартного окна добавления в список новой *информационной базы*. Если нажать на кнопку **Добавить** в режиме отображения списка баз в виде дерева, появившееся окно будет содержать, помимо команд создания новой базы или добавления в список существующее базы, команду создания новой группы. Создавая новую группу, нужно задать ей имя.

После того, как новая группа создана, в нее можно помещать записи об *информационных базах*, перетаскивая их мышью на нужную группу в окне запуска программы.

1.4. Установка конфигурации из CF-файла

Выше мы упоминали о том, что *конфигурацию* можно импортировать из файла с расширением .CF. Такой файл вам могут, например, дать на дружественном предприятии для того, чтобы вы могли установить у себя последнюю версию *конфигурации* и экспериментировать с ней. При подобном развитии событий нужно действовать следующим образом.

Сначала, в уже известном вам окне, которое появляется после запуска программы (рис. 1.1) следует нажать на кнопку **Добавить**.

В следующем окне выбрать пункт **Создание новой информационной базы**.

В следующем окне нужно выбрать пункт **Создание информационной базы без конфигурации для разработки новой конфигурации или загрузки выгруженной ранее информационной базы**.

В следующем окне указываем наименование базы.

Далее - указываем место хранения базы данных.

После завершения перечисленных действий в списке *информационных баз* появится новая пустая база. Теперь нужно выделить ее и нажать на кнопку *Конфигуратор*. Будет открыто окно *конфигуратора* (рис. 1.8).

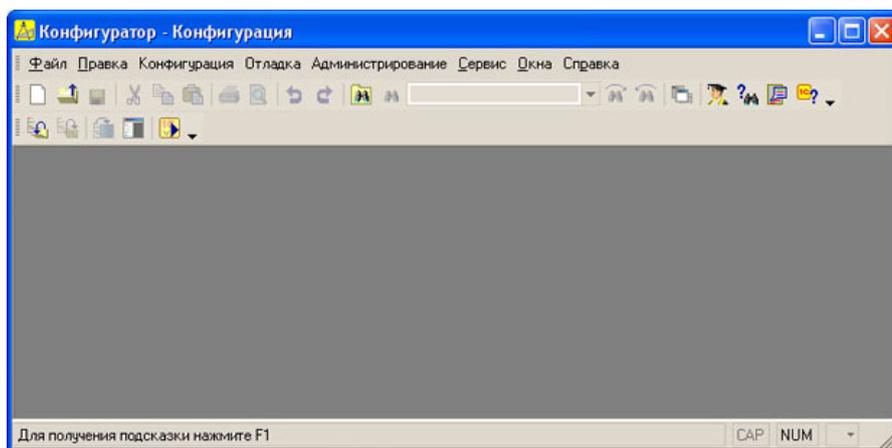


Рис. 1.8. Окно конфигуратора

При первом открытии окно пустое и сейчас нам понадобится выполнить некоторые действия для загрузки в пустую *конфигурацию* имеющегося у вас CF-файла.

Выполните команду меню **Конфигурация > Открыть конфигурацию**. Окно программы изменится - в его левой части будет открыто дерево *конфигурации*. Оно нас пока не интересует. Теперь можно выполнить команду **Конфигурация > Загрузить конфигурацию из файла**. Эта команда позволяет полностью заместить существующую (в нашем случае пустую) *конфигурацию конфигурацией*, сохраненной в CF-файле. В появившемся окне, [рис. 1.9](#), вам нужно указать путь к файлу, в котором хранится *конфигурация*, которую вы хотите загрузить.

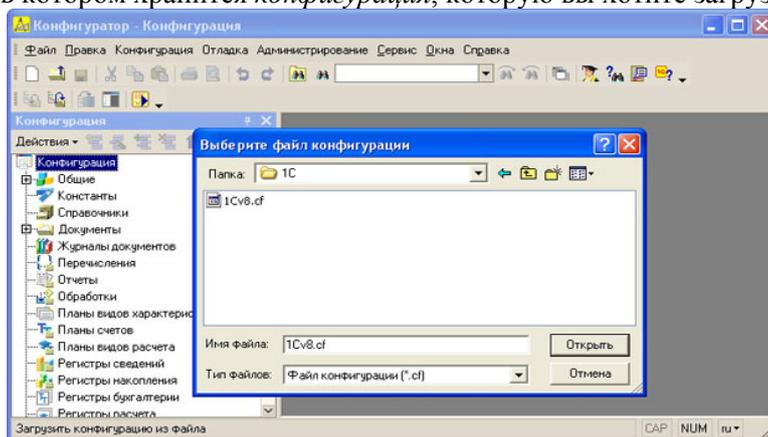


Рис. 1.9. Загрузка конфигурации из CF-файла

Теперь вам нужно лишь нажать на кнопку **Открыть** и дождаться, когда программа загрузит *конфигурацию*. Это может занять довольно много времени. Если система будет задавать вам вопросы (в частности - об обновлении *конфигурации* базы данных, о принятии изменений в структуре информации *конфигурации*) - отвечайте на них утвердительно.

В левом нижнем углу окна программы отображаются служебные сообщения о ходе обновления *конфигурации*. После того, как обновление завершится (прекратится вывод служебных сообщений о загрузке базы), вы можете закрыть окно *конфигуратора* и приступить к запуску программы в пользовательском режиме. Однако, на практике установка *информационной базы* одним из вышеперечисленных способов, обычно не ограничивает дела, которые нужно сделать до первого запуска *конфигурации*. В частности, нельзя забывать об обновлениях.

1.5. Обновление конфигурации

На практике, после установки *конфигурации*, обычно возникает необходимость в ее обновлении. Обновления распространяются в виде файлов с расширением .CFU. Такой файл копируется в ходе установки обновления в папку, содержащую, в соответствии с настройками программы, файлы шаблонов *конфигураций* и обновлений. Например, в подавляющем большинстве случаев это - папка **C:\Program Files\1cv81\tmlpls\1c**.

Имея файл обновления *конфигурации*, вы должны проверить, можно ли с его помощью обновить вашу *конфигурацию*. Обычно в файлах, сопровождающих обновления, можно найти подробные сведения о том, для обновления каких именно релизов *конфигураций* они предназначены.

Прежде чем обновлять *конфигурацию*, нужно создать резервную копию вашей *информационной базы*. Если вы обновляете только что созданную *информационную базу*, можно обойтись и без этого шага, но для того, чтобы застраховаться от негативных последствий возможных сбоев при обновлении "живой" базы лучше всего сделать резервную копию. Для этого можно воспользоваться различными методиками. В частности, при использовании файлового варианта *информационной базы* можно просто скопировать папку, содержащую *информационную базу*. Можно, открыв *конфигуратор* с нужной базой, выполнить команду **Администрирование > Выгрузить информационную базу**. Файлы выгрузки *информационной базы* имеют расширение .DT. Для того, чтобы загрузить *информационную базу*, можно воспользоваться командой **Администрирование > Загрузить информационную базу**.

Для того, чтобы обновить *конфигурацию*, нужно открыть ее в режиме *конфигуратора*, после чего выполнить следующую последовательность действий.

1. Выполнить команду меню **Конфигурация > Открыть конфигурацию** ;
2. Выполните команду **Конфигурация > Поддержка > Обновить конфигурацию** ;
3. Появится окно **Обновление конфигурации** (рис. 1.10).

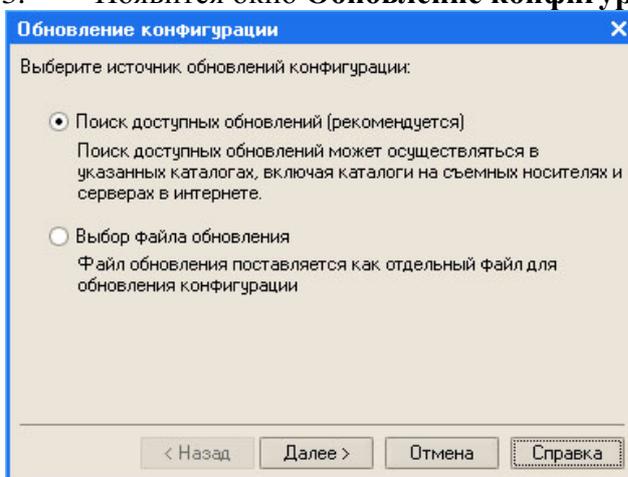


Рис. 1.10.

В этом окне следует выбрать позицию **Поиск доступных обновлений** в том случае, если вы хотите, чтобы система автоматически просканировала некоторые папки в поисках доступных обновлений. Это целесообразно делать, например, в том случае, если вы только что установили свежие обновления, которые были загружены в вышеупомянутую папку tmp\lts.

Если у вас есть нужный .CFU-файл и вы не хотите, чтобы система искала его сама, можете установить переключатель в позицию **Выбор файла обновления** ;

4. После выбора нужного файла обновления на следующем этапе, вы запускаете процесс обновления, который может быть достаточно длительным.

Когда у вас есть *информационная база* с конфигурацией, которая вас устраивает, вы можете запустить 1С:Предприятие, выбрать в стартовом окне нужную базу и нажать на кнопку 1С:Предприятие. Обычно первый запуск программы с новой конфигурацией занимает некоторое время - система выполняет подготовительные действия. А пока происходят все эти действия, давайте обсудим некоторые положения, касающиеся особенностей устройства 1С:Бухгалтерии.

1.6. Основные объекты конфигурации с точки зрения пользователя

1С:Бухгалтерия - это система, предназначенная для ввода, хранения и обработки учетных данных. Для этих целей она имеет набор компонентов (*объектов конфигурации*), которые позволяют выполнять все необходимые операции. Основным объектом, которым вам придется пользоваться при работе в программе, называется **Документ**. Именно с помощью документов данные попадают в систему. Документы заполняются пользователями, при их сохранении проводятся операции по внесению данных этих документов в систему, а на основе этих данных выполняются все остальные учетные процедуры.

Можно сказать, что 1С:Бухгалтерия предусматривает ведение учета "от документа", что, в общем-то, достаточно логично. Документы в системе 1С:Бухгалтерия очень похожи на документы, с которыми приходится иметь дело любому бухгалтеру. Можно сказать, что это - электронные аналоги обычных документов. При необходимости документы (вернее - их печатные формы) можно распечатывать на принтере. Объект **Журнал документов** используется для группировки однородных документов и выполняет функции, схожие с папкой, в которую складывают обычные бумажные документы.

Любая бухгалтерия занимается не только документами. Как известно, документы, или, как их принято называть в бухгалтерской практике - первичные учетные документы (или даже просто "первичка") это - лишь источник бухгалтерской информации. Автоматизированная бухгалтерия - не исключение. После того, как документ введен в систему, он формирует движения по регистрам. Эта довольно сложная для начинающего фраза, на самом деле, означает лишь то, что данные из документов (обычно - определенным образом обработанные) записываются в специальные таблицы, которые называют регистрами. Существует несколько разновидностей регистров.

Регистры накопления используются для накопления какой-либо информации, как правило - в числовом выражении. Например - для хранения данных о приходе товаров на склад и их выбытии с возможностью получения остатков и оборотов.

Регистры сведений можно применять для хранения информации, которая меняется со временем. Например, в такой регистр вполне можно записывать цену товаров, полученных от поставщиков для того, чтобы анализировать изменение их стоимости во времени.

Регистры бухгалтерии предназначены для ведения бухгалтерского учета.

Регистры расчета нужны для выполнения расчетных операций, например - для расчета заработной платы сотрудников организации. В конфигурации **Бухгалтерия предприятия** регистры расчета не используются, но если вы собираетесь в будущем заниматься программированием для *платформы* 1С:Предприятие, вам нужно, по крайней мере, знать об их существовании.

Не только документы и регистры предназначены для организации учета. Перечислим еще некоторые объекты системы, с которыми вам придется иметь дело.

Для хранения данных, которые изменяются очень редко, или, в идеале, не изменяются вовсе, используются **Константы**. Например, в константе может храниться номер версии конфигурации, валюта регламентированного учета.

Для хранения различной информации справочного характера используют **Справочники**. Так, справочники могут хранить сведения о сотрудниках организации, о материалах и товарах, о контрагентах и т.д.

Объекты системы с говорящим названием **Отчеты** используют для получения различных отчетов. А так называемые **Обработки** - для выполнения различных операций с данными.

Лабораторная работа №3 «Разработка тестового сценария проекта»

Цель: получить навыки разработки тестовых сценариев.

Теоретические вопросы

- Оценка стоимости и причины ошибок в программном обеспечении.
- Виды и методы тестирования.
- Понятие теста.
- Требования к разработке тестовых сценариев.
- Правила разработки тестовых сценариев.

Задание № 1

Написать программу решения квадратного уравнения $ax^2 + bx + c = 0$.

Задание № 2

Найти минимальный набор тестов для программы нахождения вещественных корней квадратного уравнения $ax^2 + bx + c = 0$. Решение представлено в таблице.

Но-мер теста	a	b	c	Ожидаемый результат	Что проверяется
1	2	-5	2	$x_1=2, x_2=0,5$	Случай вещественных корней
2	3	2	5	Сообщение	Случай комплексных корней
3	3	-12	0	$x_1=4, x_2=0$	Нулевой корень
4	0	0	10	Сообщение	Неразрешимое уравнение
5	0	0	0	Сообщение	Неразрешимое уравнение
6	0	5	17	Сообщение	Неквадратное уравнение
7	9	0	0	$x_1=x_2=0$	Нулевые корни

Таким образом, для этой программы предлагается минимальный набор функциональных тестов, исходя из 7 классов выходных данных.

Заповеди по отладки программного средства, предложенные Г. Майерсом.

Заповедь 1. Считайте тестирование ключевой задачей разработки ПС, поручайте его самым квалифицированным и одаренным программистам, нежелательно тестировать свою собственную программу.

Заповедь 2. Хорош тот тест, для которого высока вероятность обнаружить ошибку, а не тот, который демонстрирует правильную работу программы.

Заповедь 3. Готовьте тесты как для правильных, так и для неправильных данных.

Заповедь 4. Документируйте пропуск тестов через компьютер, детально изучайте результаты каждого теста, избегайте тестов, пропуск которых нельзя повторить. *Заповедь 5.* Каждый модуль подключайте к программе только один раз, никогда не изменяйте программу, чтобы облегчить ее тестирование.

Заповедь 6. Пропускайте заново все тесты, связанные с проверкой работы какой-либо программы ПС или ее взаимодействия с другими программами, если в нее были внесены изменения (например, в результате устранения ошибки).

Задание № 3

Разработайте набор тестовых сценариев (как позитивных, так и негативных) для следующей программы:

Имеется консольное приложение (разработайте самостоятельно). Ему на вход подается 2 строки. На выходе приложение выдает число вхождений второй строки в первую. Например:

Строка 1	Строка 2	Вывод
абвгабвг	аб	2
стстсап	стс	2

Набор тестовых сценариев запишите в виде таблицы, приведенной выше.

Задание № 4

Оформить отчет.

Лабораторная работа №4 «Разработка тестовых пакетов»

Цель: получить навыки разработки тестовых пакетов.

Теоретические вопросы

- Системные основы разработки требований к сложным комплексам программ.
- Формализация эталонов требований и характеристик комплекса программ.
- Формирование требований компонентов и модулей путем декомпозиции функций комплексов программ.
- Тестирование по принципу «белого ящика».

Задание № 1

В Древней Греции (II в. до н.э.) был известен шифр, называемый "квадрат Полибия". Шифровальная таблица представляла собой квадрат с пятью столбцами и пятью строками, которые нумеровались цифрами от 1 до 5. В каждую клетку такого квадрата записывалась одна буква. В результате каждой букве соответствовала пара чисел, и шифрование сводилось к замене буквы парой чисел. Для латинского алфавита квадрат Полибия имеет вид:

	1	2	3	4	5
1	A	B	C	D	E
2	F	G	H	I, J	K
3	L	M	N	O	P
4	Q	R	S	T	U
5	V	W	X	Y	Z

Пользуясь изложенным способом создать программу, которая:

- а) зашифрует введенный текст и сохранит его в файл;
- б) считает зашифрованный текст из файла и расшифрует данный текст.

Задание № 2

Спроектировать тесты по принципу «белого ящика» для программы, разработанной в задании № 1. Выбрать несколько алгоритмов для тестирования и обозначить буквами или цифрами ветви этих алгоритмов. Выписать пути алгоритма, которые должны быть проверены тестами для выбранного метода тестирования. Записать тесты, которые позволят пройти по путям алгоритма. Протестировать разработанную вами программу. Результаты оформить в виде таблиц:

Тест	Ожидаемый результат	Фактический результат	Результат тестирования
...

Задание № 3

Проверить все виды тестов и сделать выводы об их эффективности

Задание № 4

Оформить отчет.

Лабораторная работа №5 «Использование инструментария анализа качества»

Цель: получить навыки использования инструментария анализа качества.

Теоретические вопросы

- Общие требования к качеству функционирования сложных программных комплексов.
- Требования к характеристикам качества сложных программных комплексов.
- Требования к эффективности использования ресурсов ЭВМ программным комплексом в реальном времени.
- Проверка корректности функциональных требований к сложным комплексам программ.

Задание № 1

Написать программу, генерирующую массив вещественных чисел в диапазоне от –10 до 10 и определяющую все минимальные положительные элементы.

Задание № 2.

Оценить эффективность разработанной программы:

	Исходная программа		Улучшенная программа	
	Недостатки	Количественная оценка	Улучшения	Количественная оценка
Время выполнения				
Оперативная память				
Внешняя память				

Задание № 3

Оценить качество разработанной программы

	Правильность	Универсальность	Проверяемость	Точность результатов
Недостатки				
Оценка				

Задание № 4

Оформить отчет.

Лабораторная работа №6 «Анализ и обеспечение обработки исключительных ситуаций»

Цели: получение навыков анализа и обеспечения обработки исключительных ситуаций.

Теоретические вопросы

- Исключения в C++.
- Установленные исключения.
- Спецификация исключения.

Задание № 1

Написать программу, в которой обрабатываются следующие исключительные ситуации: "отрицательное значение возраста" и "год рождения больше текущего":

```
#include <iostream>
#include <ctime>
using namespace std;

int AgeCalc(int year)
{
    if (year <= 0)
    {
        throw "ERROR: negative value of the birth year!!!";
    }

    struct tm *CDate;
    time_t tt = time(NULL);
    CDate = localtime(&tt);

    if (year > (1900 + CDate->tm_year) )
    {
        throw "ERROR: The birth year value is greater than current year value!!!";
    }

    return 1900 + CDate->tm_year - year;
}

int main()
{
    int BYear = 1980;
    int PAge = 0;

    try{
        PAge = AgeCalc(BYear);
    }
    catch (const char * s)
    {
        cout << s << endl<<endl;
    }
    catch(...)
    {
        cout << "Unknown exception" << endl<<endl;
    }

    cout << "For birth year " << BYear << " the age is " << PAge << endl;

    return 0;
}
```

Задание № 2

Составить программу циклического вычисления значений функций, определенных из таблицы вариантов заданий. Значения R должны вводиться с клавиатуры. R1 и R2 – вещественные, R3 – комплексное. Предусмотреть вывод подсказок в виде (например):

Funkciya $\sin(x)$

Q – Vyihod iz programmyi

Vvedite chislo ili Q:

Для вычисления значений функции написать функцию, вычисляющую требуемые по заданию значения. При разработке функции разрешается использовать функции модуля math.h.

Предусмотреть анализ всей введенной информации на ошибки, обработку ошибок реализовать с использованием обработчиков try... в зависимости от варианта задания. Предусмотреть вывод имени функции, в которой произошла ошибка. Вывод на экран и чтение с клавиатуры организовать при помощи стандартных потоков ввода/вывода/ошибки.

Вывести исходные данные и результат в виде (например):

$\text{Sin}(R) = \text{rez};$

Где rez –результаты вычисления (вещественный).

Вариант	Функция	Обработчики
1	$\text{Sin}(R1) \cdot (\pi) / R2 - R3$	Потеря разряда Деление на 0
2	$\text{Sin}(R2) / \pi \cdot R1 + R3$	Потеря разряда Переполнение
3	$\text{Tan}(R1) / R3 + \text{Cmod}(R3)$	Потеря разряда Прерывание
4	$\text{Arctan}(R1) \cdot R2 + R3$	Потеря разряда Переполнение
5	$\text{Ln}(R1 - R2) \cdot R2 - R3$	Обл.опр. арг. Исчезновение порядка

Задание № 3

Реализуйте класс «очередь» из строк. Реализуйте методы для вставки в очередь удаления. Породите и обработайте ошибки динамического выделения памяти, переполнения очереди.

Задание № 4

Оформите отчет.

Лабораторная работа №7 «Функциональное тестирование»

Цель: получение навыков проведения функционального тестирования.

Теоретические вопросы

- Особенности функционального тестирования программного обеспечения (тестирования «черного ящика»).
- Ошибки, выявляемые при функциональном тестировании.
- Задачи, решаемые при функциональном тестировании.

Задание № 1

Опишите методы формирования тестовых наборов при использовании стратегии "черного ящика":

Эквивалентное разбиение	
Анализ граничных значений	
Анализ причинно-следственных связей	
Предположение об ошибке	

Задание № 2

Пусть необходимо выполнить тестирование программы, определяющей точку пересечения двух прямых на плоскости. Попутно, она должна определять параллельность прямой одной из осей координат.

В основе программы лежит решение системы линейных уравнений $Ax + By = C$ и $Dx + Ey = F$.

1. Используя метод эквивалентных разбиений, получаем для всех коэффициентов один правильный класс эквивалентности (коэффициент – вещественное число) и один неправильный (коэффициент – не вещественное число). Откуда можно предложить 7 тестов:

- 1) все коэффициенты – вещественные числа;
- 2)– 7) поочередно каждый из коэффициентов – не вещественное число.

2. По методу граничных условий можно считать, что для исходных данных граничные условия отсутствуют (коэффициенты – "любые" вещественные числа); для результатов – получаем, что возможны варианты: единственное решение, прямые сливаются (множество решений), прямые параллельны (отсутствие решений).

Следовательно, можно предложить тесты, с результатами внутри области и с результатами на границе.

3. По методу анализа причинно-следственных связей определяем множество условий.
 - а) для определения типа прямой;
 - б) для определения точки пересечения.

Выделяем три группы причинно-следственных связей (определение типа и существования первой линии, определение типа и существования второй линии, определение точки пересечения) и строим таблицы истинности.

К уже имеющимся тестам добавляются:

а) проверки всех случаев расположения обеих прямых – 6 тестов по первой прямой вкладываются в 6 тестов по второй прямой так, чтобы варианты не совпадали, – 6 тестов;

б) выполняется отдельная проверка несовпадения условия $x \square = 0$ или $y = 0$ (в зависимости от того, какой тест был выбран по методу граничных условий) – тест также можно совместить с предыдущими 6 тестами;

4. По методу предположения об ошибке добавим тест, при котором все коэффициенты – нули. Всего получили 20 тестов по всем четырем методикам. Если еще попробовать вложить независимые проверки, то возможно число тестов можно еще сократить.

Задание № 3

Разработать программу определения вида треугольника, заданного длинами его сторон: равносторонний, равнобедренный, прямоугольный, разносторонний.

Предлагаемые тесты свести в таблицу.

Номер теста	Назначение теста	Значения исходных данных	Ожидаемый результат	Реакция программы	Вывод

Задание № 4

Разработать программу решения уравнения $ax^2 + bx + c = 0$, где a, b, c – любые вещественные числа.

Предлагаемые тесты свести в таблицу.

Номер теста	Назначение теста	Значения исходных данных	Ожидаемый результат	Реакция программы	Вывод

Задание № 5

Оформить отчет.

Лабораторная работа №8 «Тестирование безопасности»

Цель: получение навыков тестирования безопасности информационной

Теоретические вопросы

- Тестирование восстановления.
- Тестирование безопасности.
- Технологии тестирования безопасности.
- Тестирование безопасности – оценка уязвимости программного обеспечения к различным атакам.

Компьютерные системы очень часто являются мишенью незаконного проникновения. Под проникновением понимается широкий диапазон действий: попытки хакеров проникнуть в систему из спортивного интереса, месть рассерженных служащих, взлом мошенниками для незаконной наживы. Тестирование безопасности проверяет фактическую реакцию защитных механизмов, встроенных в систему, на проникновение. В ходе тестирования безопасности испытатель играет роль взломщика. Ему разрешено все:

- попытки узнать пароль с помощью внешних средств;
- атака системы с помощью специальных утилит, анализирующих защиты;
- подавление, ошеломление системы (в надежде, что она откажется обслуживать других клиентов);
- целенаправленное введение ошибок в надежде проникнуть в систему в ходе восстановления;
- просмотр несекретных данных в надежде найти ключ для входа в систему.

При неограниченном времени и ресурсах хорошее тестирование безопасности взломает любую систему. Задача проектировщика системы – сделать цену проникновения более высокой, чем цена получаемой в результате информации.

Задание № 1

Изучите и опишите одно из средств выявления уязвимостей:

Таблица 1. Обзор средств выявления уязвимостей, работающих на уровне кода

Наименование средства	Назначение	Поддерживаемые языки программирования	Примечание
Иностраные средства выявления уязвимостей			
Its4	Статически просматривает исходный код для обнаружения потенциальных уязвимостей защиты	C/c++	Отмечает вызовы потенциально опасных функций, таких, как strcpy/memcpy, и выполняет поверхностный семантический анализ, пытаясь оценить, насколько опасен такой код, а также дает советы по его улучшению
Rats(rough auditing tool for security)	Просматривает исходный текст, находя потенциально опасные обращения к функциям	C/c++, php, perl, python	глубокого семантического анализа в поисках дефектов, способных привести к переполнению буфера, полученных из mops
Flawfinder	Просматривает исходный текст, находя потенциально опасные обращения к функциям	C/c++	Выполняет поиск функций, которые чаще всего используются некорректно, присваивает им коэффициенты риска (опираясь на такую информацию, как передаваемые параметры) и составляет список потенциально уязвимых мест, упорядочивая их по степени риска

Flexelint (pc-lint)	Производит семантический анализ исходного кода, анализ потоков данных и управления	C/c++	В конце работы выдаются сообщения нескольких основных типов: – возможен нулевой указатель – проблемы с выделением памяти (например, нет free() после malloc()) – проблемный поток управления (например, недостижимый код); – возможно переполнение буфера, арифметическое переполнение; – предупреждения о плохом и потенциально опасном стиле кода
Parasoft c++ test Coverity	Формирование тестов анализа уязвимостей на уровне метода, класса, файла и проекта Используется для выявления и исправления дефектов безопасности и качества в приложениях критического назначения	C++ C/c++, java	Генерирует тестовый код, вызывая для его подготовки компилятор visual c++ Способен с минимальной положительной погрешностью обрабатывать десятки миллионов строк кода, обеспечивая 100-процентное покрытие трассы
Klocwork k7	Предназначен для автоматизированного статического анализа кода, выявления и предотвращения дефектов программного обеспечения и проблем безопасности	C/c++, java	Выявляет коренные причины недостатков качества и безопасности программного обеспечения
Codesurfer	Может применяться для поиска ошибок в исходном коде, для улучшения понимания исходного кода	C/c++	Позволяет проводить анализ указателей, использовать и определять переменные, зависимости данных, строить графы вызовов
Fxcop	Способен обнаружить более 200 недочетов (или ошибок) в следующих областях: – архитектура библиотеки; – правила именования; – производительность; – безопасность	C/c++	Откомпилированный код проверяется с помощью механизмов рефлексии, парсинга msil и анализа графа вызовов

Qaudit	Быстрый анализ исходных файлов на наличие переполнения буфера, ошибок форматной строки, запросов исполняемых вызовов, переменных среды, и функций, имеющих проблемы защиты	C/c++	Написать на интерпретируемом языке perl, прост в использовании
Российские средства выявления уязвимостей			
Ак-вс	Автоматизированный анализ исходных текстов, с целью выявления потенциально опасных сигнатур	C/c++, java, pascal,c#, php, assembler	Позволяет проводить статический анализ исходных текстов, динамический анализ, имеет базы сигнатур для каждого из поддерживаемых языков программирования
Аист-с	Автоматизированный анализ исходных текстов	C/c++	Позволяет проводить статический анализ исходных текстов
Ксайт	Автоматизированный анализ исходных текстов	C/c++	Позволяет проводить статический анализ исходных текстов
Uca	Предназначено для выявления потенциально опасных сигнатур	C/c++, pascal, perl, plm	Имеет базы сигнатур для каждого из поддерживаемых языков программирования
Viva64	Помогает отслеживать в исходном коде потенциально опасные фрагменты, связанные с переходом от 32-битных систем к 64-битным	C/c++	Помогает писать корректный и оптимизированный код для 64-битных систем

Задание № 2

Разработать приложение, интерфейс которого представлен на рисунке.

Коррекция

Имя:

Телефон:

Список:

- Валентина:333-33-33
- Василий:222-22-22
- Ирина:555-55-55
- Максим:111-11-11

Всего: 4

Задание № 3

Добавить в программу форму авторизации по имени и паролю.

Лабораторная работа №9 «Нагрузочное тестирование, стрессовое тестирование»

Цель: получение навыков проведения нагрузочного и стрессового тестирования.

Теоретические вопросы

- Особенности нагрузочного тестирования.
- Особенности стрессового тестирования.

Задание № 1

Разработать Компилятор простых арифметических выражений, например $2 + (-5) * (7 - 8)$.

Вход и выход осуществляются в виде строк.

Задание № 2

Разработать тестовый сценарий нагрузочного тестирования. Ответить на вопрос – сколько запросов в секунду может обработать приложение при условии, что они идут последовательно. Построить график зависимости времени ответа от количества параллельных запросов (рассматривать логарифмическую шкалу по основанию два, т.е. 1, 2, 4, 8, 16, 32 и т.д. запроса) Ответить на вопрос – какое максимальное количество параллельных запросов может обработать приложение без сбоев.

Задание № 3

Оформить отчет.

Лабораторная работа №10 «Тестирование интеграции»

Цель: получение навыков тестирования интеграции.

Теоретические вопросы

- Особенности тестирования интеграции.
- Методы интеграционного тестирования.
- Нисходящее тестирование интеграции.
- Восходящее тестирование интеграции.
- Сравнение нисходящего и восходящего тестирования интеграции

Задание № 1

Разработать приложение, состоящее из трех модулей:

- 1) главный модуль, считывающий из текстового файла координаты точек на плоскости;
- 2) модуль, содержащий функции расчета расстояния между двумя точками;
- 3) модуль, содержащий функцию, определяющую треугольник с максимальной площадью.

Задание № 2

Описать этапы нисходящего проектирования разработанного приложения.

Задание № 3

Описать этапы восходящего проектирования разработанного приложений.

Задание № 4

Оформить отчет.

Лабораторная работа №11 «Конфигурационное тестирование»

Цели: получение навыков проведения конфигурационного тестирования.

Теоретические вопросы

– Особенности конфигурационного тестирования.

Конфигурационное тестирование (Configuration testing). Проверяется работоспособность при различных конфигурациях, предполагает тестирование работы системы на различных платформах: различных вариантах аппаратной конфигурации, версиях операционной системы и окружения.

Задание № 1

Дана структура с именем ZNAK, состоящая из полей:

- фамилия, имя;
- знак Зодиака;
- дата рождения (массив из трех чисел).

Написать программу, которая выполняет следующие действия:

- ввод с клавиатуры данных в массив, состоящий из 8 элементов типа ZNAK, и занесение их в файл данных;
- чтение данных из файла и вывод их на экран;
- вывод на экран информации о людях, родившихся в месяц, значение которого введено с клавиатуры (если таких нет – вывести об этом сообщение);
- список должен быть упорядочен по знакам Зодиака.

Задание № 2

Описать и обосновать итоги тестирования работы разработанного приложения на различных платформах: различных вариантах аппаратной конфигурации, версиях операционной системы и окружения.

Лабораторная работа №12 «Тестирование установки»

Цель: получение навыков тестирования установки.

Теоретические вопросы

– Комплексное тестирование приложения.

Задание №1

Разработать приложение, интерфейс которого представлен на рисунке 1.

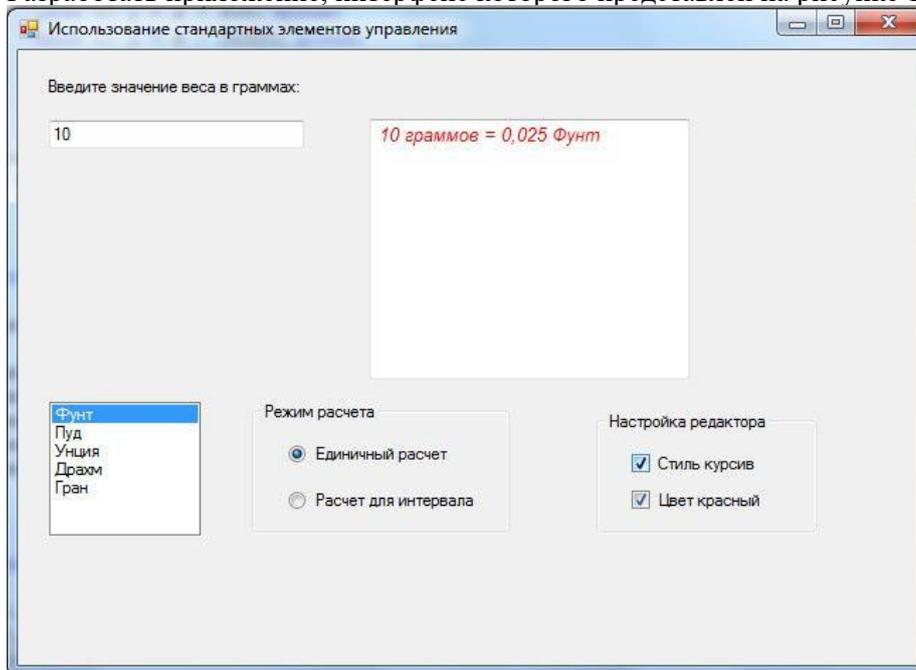


Рисунок 1. Интерфейс приложения

Задание №2

Провести комплексное тестирование разработанного приложения.

Задание №3

Оформить отчет.