



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ
ГИДРОМЕТЕОРОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра Прикладной информатики

БАКАЛАВРСКАЯ РАБОТА

На тему

«РАЗРАБОТКА АВТОМАТИЗИРОВАННОГО РАБОЧЕГО МЕСТА
ПРЕПОДАВАТЕЛЯ» НА ПРИМЕРЕ : «РАЗРАБОТКА ПРОГРАММ
ДЛЯ ЗАПОЛНЕНИЯ ЭКЗАМЕНАЦИОННЫХ БИЛЕТОВ»

Исполнитель

Петренко Станислав Олегович

Руководитель

Ст. преподаватель кафедры «Прикладная информатика»
Степанов Сергей Юрьевич

«К защите допускаю»
Заведующий кафедрой

(подпись)

кандидат технических наук
Слесарева Людмила Сергеевна

«22» 06 2016 г.

Санкт-Петербург
2016

ОГЛАВЛЕНИЕ

Введение	3
Глава 1 Основания для разработки и требования к системе.....	6
1.1 Описание предметной области	6
1.2 Обзор существующих информационных систем составления экзаменационных билетов.....	7
1.3 Анализ информационной структуры	9
1.4 Функциональные требования к разрабатываемой информационной системе	13
Глава 2 Инструменты и технологии реализации	15
2.1 Выбор языка программирования.....	15
2.2 Выбор средств разработки	24
Глава 3 Процесс разработки информационной системы.....	35
3.1 Разработка пользовательской формы.....	35
3.2 Разработка объекта.....	37
3.3 Реализация проекта	42
Заключение.....	49
Список источников литературы.....	52

Введение

Автоматизация— это одна из возможностей, предоставляемых технологией Microsoft COM (ComponentObjectModel). Она используется приложениями (называемыми COM-серверами) для предоставления доступа к их объектам, а также к свойствам и методам этих объектов другим приложениям (называемым COM-клиентами), каковыми могут быть и средства разработки. В частности, текстовый процессор MicrosoftOfficeWord, будучи COM-сервером, может предоставлять другим приложениям доступ к документу, абзацу, закладке с помощью соответствующих объектов. Для именования (и опознания) COM-серверов обычно используются специальные строковые обозначения — программные идентификаторы (ProgrammaticIdentifier, ProgID). Они нужны для того, чтобы операционная система могла с помощью этих идентификаторов определить, в каком именно каталоге (или на каком компьютере локальной сети, если используется тот или иной способ удаленного доступа к серверу) расположен исполняемый файл сервера автоматизации, и запустить его на выполнение.

Объекты автоматизации с точки зрения программирования мало чем отличаются от обычных объектов из теории и практики объектно-ориентированного программирования. Как и обычные объекты, они обладают свойствами и методами. Свойство— это характеристика объекта; например, свойством абзаца (объект Paragraph) может быть его стиль (Style). Методом называется действие, которое можно выполнить с объектом (например, можно сохранить документ с помощью метода SaveAs объекта Document).

Нередко серверы автоматизации содержат наборы однотипных объектов, называемых коллекциями. Например, текстовый процессор может содержать коллекцию документов, а каждый документ— коллекцию абзацев.

Объект исследования — прикладное программное обеспечение MicrosoftOfficeWord. Предмет исследования — разработка программных

приложений в среде MicrosoftOfficeWord.

Актуальность выбранной темы заключается в том, что существует необходимость автоматизации обработки текстовых документов.

Цель работы — рассмотрение теоретических основ разработки программных приложений в среде MicrosoftOfficeWord, анализ существующих решений в данной области и предложения по применению перспективных технологий в данной области.

Исходя из актуальности темы и поставленных целей, при написании выпускной квалификационной работы были поставлены следующие задачи:

- рассмотреть теоретические основы разработки программных приложений в среде MicrosoftOfficeWord;
- рассмотреть объектную модель MicrosoftOfficeWord;
- разработать программное приложение в среде MicrosoftOfficeWord.

Указанные цели и задачи определили логику изложения материала и структуру работы, состоящую из трёх глав.

Первая глава освещает предметную область разработки программных приложений в среде MicrosoftOfficeWord.

Во второй главе даётся описание объектной модели MicrosoftOfficeWord.

В третьей главе приводится описание процесса разработки программного приложения в среде MicrosoftOfficeWord.

Методологической основой написания работы являются:

- эмпирические методы: наблюдения и изучение процессов обработки информации в вычислительных системах и программных комплексах;
- теоретические методы: теоретический анализ учебной литературы (выделение и рассмотрение отдельных сторон, признаков, особенностей, свойств явлений);
- практический опыт разработки программных приложений.

Теоретической основой написания работы является литература в

данной области. Список использованных литературных источников приведён в конце работы.

При выполнении данной работы было использовано программное обеспечение класса текстовый процессор, а также система управления базами данных.

Глава 1 Основания для разработки и требования к системе

1.1 Описание предметной области

Требуется создать программное приложение для автоматизированного заполнения экзаменационных билетов аттестации сотрудников АО «Туапсинское АТП» на базе ТГМТ. Приложение должно предусматривать возможность создания шаблона экзаменационного билета аттестации (рис. 1.1) и базу данных вопросов к экзамену. Вопросы должны заполняться связанно и последовательно. Для решения данной задачи было решено использовать прикладное программное обеспечение MicrosoftOfficeWord, ввиду его наибольшего распространения, степени изученности и удобства использования. Базу данных было решено создавать при помощи системы управления базами данных MicrosoftOfficeAccess, также ввиду её наибольшего распространения, степени изученности и удобства использования.

Министерство образования и науки Краснодарского края
Государственное бюджетное образовательное учреждение
среднего профессионального образования
«ТУАПСИНСКИЙ ГИДРОМЕТЕОРОЛОГИЧЕСКИЙ ТЕХНИКУМ»
Краснодарского края

Специальность: 230103 «Автоматизированные системы
обработки информации и управления (по
отраслям)»

Дисциплина: «Название дисциплины»
Семестр: 6
Форма обучения: очная

Экзаменационный билет № 000000

СОСТАВИЛ: Преподователь информатики
→ /Филатов О. А./
(подпись)

УТВЕРЖДАЮ: Председатель ЦМК
→ /Обидейко А. Н./
(подпись)

« → » 2014 г.

Рисунок 1.1 Шаблон экзаменационного билета

Разработка программного приложения ведётся в среде VBA,

встроенной в приложения пакета прикладного программного обеспечения MicrosoftOffice. При разработке программного приложения была изучена среда программирования VBA, язык программирования VBA и объектная модель MicrosoftOfficeWord. Также были необходимы навыки создания текстовых документов и базы данных, а также знание основных приёмов работы с текстовыми документами и базами данных.

1.2 Обзор существующих информационных систем составления экзаменационных билетов

Аттестация сотрудников предприятия проводится в целях беспристрастной оценки их пригодности к предстоящей работе, повышения обязанности отвечать и исполнительской дисциплины, а также для принятия хороших решений по кадровым назначениям.

Результатом аттестации является решительное заявление Генерального управляющего руководителя (Директора филиала - при аттестации служащих аппарата управления филиала или компаний, входящих в смесь состава филиала), принятое с учетом рекомендаций Аттестационной комиссии и действующего законодательства РФ в положения взгляда касательно сотрудников, прошедших аттестацию.

Решения имеют все шансы быть надлежащими: о вознаграждении индивидуальных работников за достигнутые ими успехи; об изменении работнику величины габарита публично должностного оклада; об установлении, изменении либо отмене надбавок к должностным публично окладам; о направленности на с повышением повышение с с повышением увеличением квалифицированного мастерства; о повышении в poste службы ; о доказательстве должности, величины габарита публично должностного оклада и персональных надбавок; о понижении в должности.

Принятое решительное заявление обязано быть оформлено в облике Приказа Генерального управляющего руководителя (Директора филиала.

Подписанный Приказ посылается в соответствующее осознано и преднамеренно предназначенное дочернее отделение по работе с обслуживающим составом (Дирекцию по УП Исполнительной Дирекции, Отдел УП филиала или Группу работников предприятия) для уведомления (под роспись) с принятым решением работника, по которому оно принято.

В по ходу аттестации состава кадров в выполнении исполняется надлежащие сеансы мероприятий : подготовка Приказа о проведении аттестации; формирование аттестационных комиссий; подготовка Графика проведения аттестации; проведение разъяснительной трудовые нагрузки по процедуре аттестации; подготовка документальных подтверждений на аттестуемых служащих ; проведение аттестации; оформление итоговых продуктов аттестации; разрешение спорных бесед по действенным отчетам аттестации; принятие решения по итогам аттестации.

Кроме такого , применяя автоматические конструктивные порядки обретения навыков образования и контрольные сверки познаний , можно осуществить корпоративную трудовую эксплуатационную деятельность в учебном классе учащей организации либо корпоративной сети организации с делегированием возможностей структурным дочерним отделениям , что позволяет проведения экзамена либо аттестации территориально удаленных работников организации. Компьютерная обучающе-контролирующая конструктивный порядок «ОЛИМП:ОКС», с разработкой внедренная нашей группой , сотворена для организации хода формирования процесса дела обретения навыков образования и контрольные сверки знаний состава работников в учебных классах специализированных учебных центров или конкретно в организациях.

В реальный часть отрезка времени группой «Термика» вместе с задающими тон формированиями в области обретения квалификации и контрольные сверки познаний для работников формирований различных сфер деятельности создано больше 100 50-ти многофункциональных обучающе тренировочных курсов (тестов), позволяющих отводить провожая

контрольную сверку познаний в всевозможных областях познаний (промышленная, добывающая энергетическую деятельность, экологическая, пожарная, передвижная защищенность, охраняемая защита труда, безопасность гидротехнических сооружений). С полным списком обучающе-тренировочных курсов и тестов для контрольных сверки познаний можно ознакомиться в разделе Методическое обеспечение Интернет-сервера компании «Термика».

1.3 Анализ информационной структуры

Программирование в Office — это прежде ранее лишь уменьшение численности циклических событий (и ручной трудовой обязанности, которая для того чтобы достигнуть желанного результата требуется). Вот примеры неких обычных обстановок, когда применение программирования имеет подтекст логичный: с достоверно установленной периодичностью приходится изготавливать деловые бумаги, крайне подобные дружеский сторонник на приятеля по дружбе: приказы, приказного постановления в бухгалтерию, договорные обещания, доклады и т. п. Часто коммуникационные данные можно взять из базы уведомительно справочных сведений — тогда уже применение программирования может предоставить крайне огромный выгодный навар во отрезка времени продолжительности. Иногда ее приходится вводить вручную, но тогда уже и автоматизация даст выгодный навар и во времени, и в понижении численности с ошибками оговорок.

Разновидность той же обстановки: одни и другие же справочно уведомительные сведения необходимо применить пару раз. Например, при заключении договорного согласия с заказавшем заказчиком. Одни и другие же справочно уведомительные сведения (наименование, адрес, расчетный счет, номер договора, дата заключения, сумма и т. п.) имеют все шансы потребоваться во большом количестве документальных архивов: самом

договорном обещании , счете, счете-фактуре, акте сдачи выполненных трудовых использований и так дальше . Логично один раз ввести эти справочно уведомительные сведения (скорее всего, в основание данных), а позже автономно без помощи и роботизированно формировать (например, в Word) требуемые документы.

Когда необходимо сделать так, для того чтобы вводимые пользователем справочно уведомительные сведения автономно без помощи и роботизированно проверялись. Вероятность ошибки при ручном вводе уведомительно справочных сведений зависит от многих разных факторов, но, согласно результатам неких исследований, она ориентировочно составляет около двух процентов. «Вылавливать» позже такие ошибки в уже введенных уведомительно справочных сведений — крайне тяжелый труд, поэтому сразу сделать так, для того чтобы они не возникали.

В общем, хоть какое влияние, которое приходится твердить более нескольких раз — это вероятно допустимый кандидат на автоматизацию. Например, занесение 100-н контактов в Outlook, либо подмена ресурса в 10-ках проектных замыслов Project, либо экспертный анализ коммуникационным данным из базы уведомительно справочных сведений за различные временные промежутки в таблице Excel — те обстановки , когда познание объектных модельных марок добавочных гаджетов Office выручит от часовых отрезков и временных промежутков дня и ночи скучноватого труда. Кроме того, применение программирования несет к тому же остальные выдающиеся качества для служащего работника ,который применяет его в работе.

Повышается его престиж в глазах руководящего состава и иных служащих, если программной разработки этого юзера интенсивно употребляются на производственном объединении (им самим либо другими сотрудниками), этим самым он охраняет себя от сокращений, снижения заработной платы и т. п. — так как поддерживать их и изменять если необходимости будет некому.

Входными документами являются:

1. Все регистрационные данные: логин, пароль, имя пользователя;
2. Данные об экзамене: Название, цельный список вопросов;
3. Ответы на вопросы;

Выходными сообщениями являются списки экзаменов, экзаменационных вопросов, а также ответов к ним. Помимо прочего в программе так же выводится информация о пользователе и его правах в конкретном экзамене.

Список экзаменов является главной страницей приложения. На нём выводятся все актуальные экзамены, количество вопросов в них, а так же автор и дата публикации. При нажатии на конкретный экзамен открывается список вопросов к нему.

В списке вопросов показаны все вопросы к экзамену, показано, дан ли ответ на конкретный вопрос (+) и даны ссылки на поисковые запросы по данному вопросу (Я, Г). Так же выводится название вопроса и стандартные кнопки — выхода и создания нового вопроса.

На данные, хранящиеся в БД, накладываются следующие ограничения:

- Логин и пароль пользователя должны быть длиннее пяти символов.
- Пустые строки в списке экзаменов недопустимы.
- Номера экзаменов уникальны
- Номера вопросов уникальны в каждом экзамене.
- Идентификационные номера вопросов уникальны для всей БД.
- Количество вопросов закреплено за экзаменом.

При анализе основных процессов и информационных объектов, их структуры, были определены следующие информационные объекты (табл. 1.1, рис.1.2).

Пользователь — физическое лицо, зарегистрированное в системе в соответствии с установленным порядком, и обладающее соответствующими

полномочиями.

Автор — пользователь, создавший экзамен считается автором экзамена. Автор имеет право просматривать и редактировать любой вопрос экзамена.

Участник — пользователь, ответивший хотя бы на один вопрос экзамена. Участник имеет право редактировать ответы на экзамен, где он является участником.

Таблица 1.1 - Условные обозначения

Условное обозначение	Значение
П	Пользователь
У	Участник
А	Автор
Э	Экзамен
В	Экзаменационный вопрос
О	Ответ

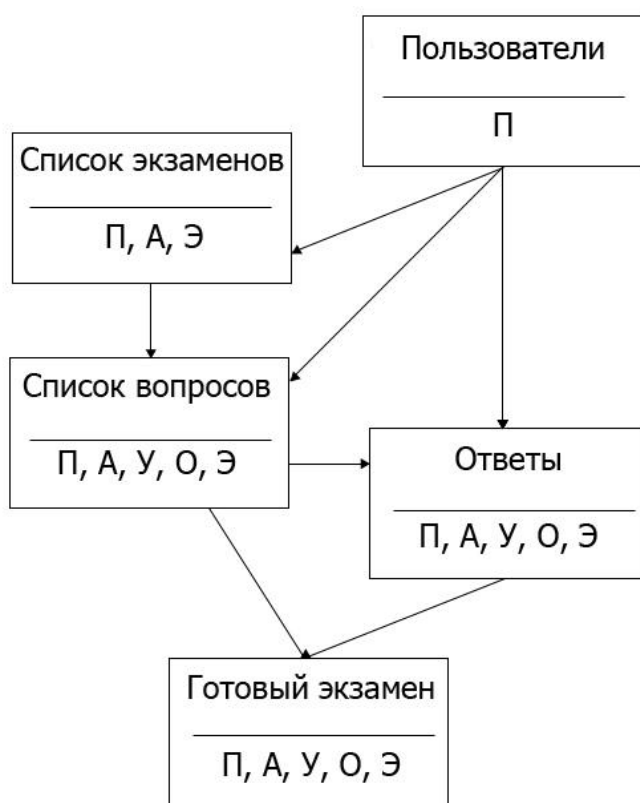


Рисунок 1.2 – Граф алгоритмической взаимосвязи показателей

ER-диаграмма процесса представлена на рисунке 1.3.

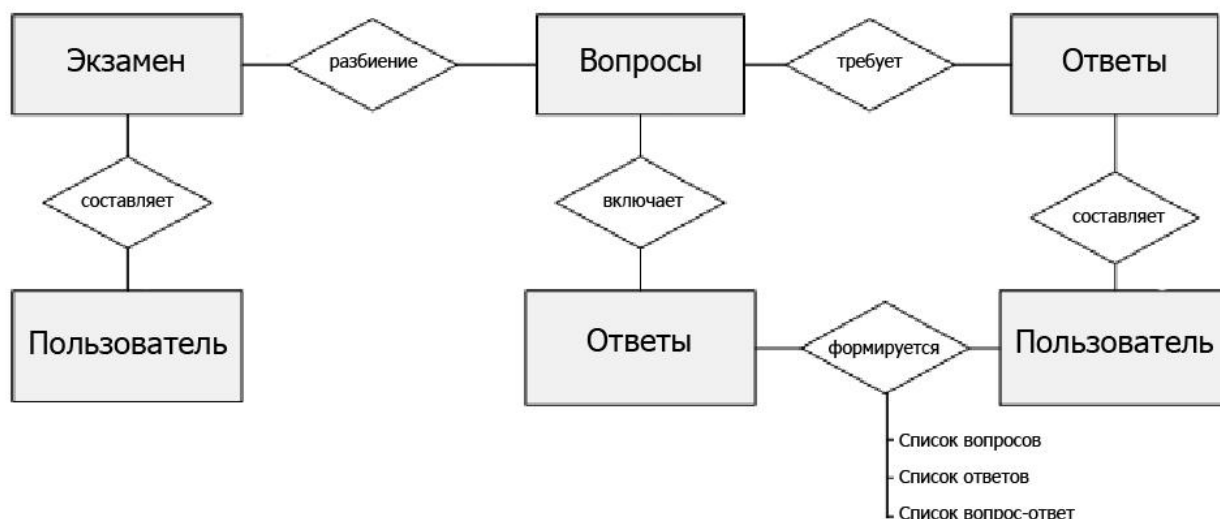


Рисунок 1.3 – ER – диаграмма

В результате анализа предметной области была построена инфологическая модель информационных объектов и выполнен функциональный анализ предметной области.

1.4 Функциональные требования к разрабатываемой информационной системе

В процессе работы с данным программным продуктом формируются и при необходимости распечатываются необходимые документы. Сначала сотрудник АО «Туапсинское АТП» заполняет базу исходных данных. Которая состоит из перечня возможных вопросов. Затем, при запуске, программа формирует два случайных вопроса. Требования к интерфейсам информационной системы

Цветовую палитру интерфейсов выдержать в стиле Windows. Разрешение экрана: 1024*768 — средний шрифт.

Интерфейс должен быть защищенным от неправильных действий пользователя.

Требования к техническому и программному обеспечению. Для ввода и

корректировки данных в интерактивном режиме необходимо использовать несколько компьютеров, объединенных в локальную сеть.

Требования к программному обеспечению — операционная система Windows XP и более поздних версий, MicrosoftOffice не ниже 2007.

Требования к техническому обеспечению — компьютер Pentium 4, 1 ГГц или более поздних моделей стандартной конфигурации, лазерный принтер LaserJet 1100.

Глава 2 Инструменты и технологии реализации

2.1 Выбор языка программирования

VBA есть комплект средств и сбережений программирования для сотворения своих личных программ и подгонки имеющихся добавочных гаджетов под требования юзера.

Приложение — это полномасштабная программная разработка , выполняющая конкретно выявленную практически подтвержденную трудовую обязанность (например, текстовый электрический микрочип , электрические таблицы либо приложение баз данных).

С поддержкою VBA можно изменять наружный облик либо метод внедрения имеющихся средств и сбережений приложения, а еще добавлять свои, безусловно новейшие возможной вероятности . Microsoft сотворила VBA и обеспечением удовлетворила спасительную помощь с поддержкою VBA во всех по главному главных приложениях Office: Word, Excel, Access и PowerPoint. Объектно нацеленное программирование Понимание объектов покоится в базе программирования в VBA, тем более когда дело касается сотворения пользовательских диалоговых окон и использования допустимых вероятностей задающего характер VBA приложения.

Объектом по названию зовется каждая именованная суть, имеющая: свойства, т. е. установки, которые можно испытать и поменять; методы, т. е. деяния , которые может исполнить предмет , когда программная разработка попросит об этом; события, т. е. обстановки , в каких объект как оказалось и на которые может ответить заблаговременно точно установленными для таких ситуаций деяниями.

Коллекция — это VBA предмет специально подготовленного предназначения . Коллекции сделаны для упрощения трудовые обязанности с набором объектов, когда этот набор объектов необходимо применить как одно единое . Как правило, всё объектов предметы в коллекции имеют один и тот же тип. Например, скопленная комплектация Pages состоит из объектов

Page. Однако в VBA выживает родовой предмет Collection, сделанный для хранения в нем объектов всех типов в любой комбинации.

Формой именуют хоть какое сделанное в VBA пользовательское окошко . Официально модели содержания в VBA описываются в определениях предмета UserForm. Каждый объект UserForm находится в собственности сразу двум коллекциям объектов: VBA программному замыслу , в каком хранится модель содержания , и коллекции UserForms, содержащей всё формы, загружаемые программой.

Свойства — это объективные оценки предмета . Каждое свойство собирает коммуникационные данные о некоем нюансе наружного облика , манер сдерживания , содержимого предмета . Главной с вопросами неувязкой характеристики является описание некоторой объективной оценки предмета .

Методы — это именованные воздействия, которые предмет может исполнить по команде. Ввиду такого , что каждый способ является необходимой отдельным кусочком предмета , предмет сам знает, что ему делать, когда вызывается метод. Таким образом видением, методы действий — не что другое , как сеансы мероприятий , привязанные к определенному предмету . Чтобы вызвать метод, слишком важно напечатать наименование имени предмета , точку, а позже наименование имени метода.

Событие есть что-то , случающееся с объектом, и то, на что объект может ответить заблаговременно задействованным с предвидением воздействием . К событиям относится следом идущее : физические деяния юзера программной разработки , к примеру щелчок кнопкой мыши, перемещение курсора и т. д.; ситуации, в которые попадает объект в процессе выполнения программы.

Язык VBA является объектно нацеленным. Это обозначает , что многие его команды имеют специфический формат. Типичная команда VBA имеет вид:

< Объект > . < Объект, входящий в первый объект > .

< ... > . < Тот объект, с которым нужно произвести действие > . < собственно действие >

Иными словесными фразами, буквально каждая команда пишется вроде бы с «конца»: сначала определяется то, над чем нужно изготовить воздействие, — предмет, а в последствии этого само действие — способ. Разделителями наборов элементов команды служат знаки «точка».

Тип уведомительно справочных сведений — это термин, относящийся к достоверно установленным обликам данных, которые VBA предохраняет и которыми может манипулировать. Любое определение типа задает: область абсолютно вероятно допустимых величин типа; структуру организации данных; — операции, определенные над данными этого типа.

VBA имеет 6 различных математических типов справочно уведомительных сведений: Byte, Integer, Long, Single, Double и Currency. Численные типы данных употребляются для сохранения (и манипулирования) чисел в различных величинах формата, зависимо от конкретно выявленного типа. Численные типы дают компактно изготовленный и действенный метод сохранения чисел. Численный тип, заполняющий огромную часть памяти (имеющий самый большущий спектр возможно реально действующих на данный момент значений), занимает не больше восьми байтов памяти для сохранения чисел, которые могут иметь до 300 цифр.

Числа с плавающей точкой (floatingpointnumbers) имеют все шансы иметь хоть какое число цифр до либо в последствии десятичной точки (в границах пределов определенного типа данных) и приобрели личное наименование на основании такого факта, что десятичная точка «плавает» из одной положения взгляда в другую, зависимо от того, сберегается в памяти большое либо небольшое величина логичного подтекста. Числа 24.09156, -1207.7, -0.00225 и 444.67779 — это числа с плавающей точкой. Числа с плавающей точкой время от времени именуют еще действительными (real) числами. Данные типы с плавающей точкой употребляются всегда, когда требуется сохранить число, имеющее дробную часть. VBA имеет 2 различных типа данных с плавающей точкой: Single и Double.

Хотя числа одинарной и попарно сдвоенной точности имеют огромные спектры , чем остальные численные типы справочно уведомительных сведений , у их есть в наличии 2 маленьких недостатка. Операции, выполняемые над числами с плавающей точкой, незначительно медленнее очень подобных процедур вмешательства над другими численными типами данных. Кроме такого , числа, хранимые как типы данных с плавающей точкой, имеют все шансы быть подвержены ошибкам округления. Если число с плавающей точкой очень огромное либо очень малое, VBA отображает его в экспоненциальном представлении.

VBA тип Currency — это число с фиксированной точкой (fixedpointnumber), т. е. десятичная точка буквально повсеместно располагается в одном и том же месте — справа от десятичной точки буквально повсеместно в наличии есть 4 единицы числа . Тип Currency употребляется для сохранения чисел, когда точность очень принципиальна , что случается при «денежных» вычислениях. Любые текстовые уведомительно справочные сведения , сохраняемые в программной разработке VBA, именуются строками (strings). Строки в VBA берегутся с в потребление использованием типа уведомительно справочных сведений String. Строки заполучили получая такое заглавие, так как текстовые данные стандартно и типично рассматриваются как строки знаков . Строка может содержать текстовые символы всех типов: алфавитные знаки алфавита, цифры, знаки пунктуации или различные символы. Строки в коде VBA буквально повсеместно заключаются в двойные кавычки ("").

Существуют две категории строк: строки переменной длины, величина габарита которых вырастает либо миниатюризируется , и строки фиксированной длины, величина габарита которых всякий раз остается одним и этим же . Все строки в VBA являются строками переменной длины, в неумышленном событии если исключительно в программной разработке не задается фиксированная длина. Типы String играют значимую по важности ролевая назначенная функция значения в практически всех

программных разработках VBA. Большинство уведомительно справочных сведений ввода обладателей (в диалоговых окошках , ячейках рабочих листов) — это строковые информационно справочные сведения . Кроме такого , так как на дисплее можно показывать только текст, все другие типы уведомительно справочных сведений обязаны быть преобразованы в строковые данные перед тем, как они будут выведены на экран. Многие интегрированные процедуры VBA используют строковые данные во всех либо в некоторых своих аргументах.

VBA пользуется тип Date для сохранения даты и длительности отрезка времени . VBA тип Date является типом поочередных дат (serialDates).(Последовательные даты охраняют дату как число интервал дня и ночи от данной стартовой даты.) Базовой числом даты для VBA типа Date является 30 1-го месяца зимы 1899. VBA пользуется отрицательно негативные числа для представления дат прежде 12/30/1899 и лестные — для дат после 12/30/1899. Число 0 с демонстрацией показывает саму дату 12/30/1899. По той самой схеме 1 второго зимнего месяца 1900 записывается числом 2 (01/01/1900 — это 2 промежутка дня и ночи после 12/30/1899), однако число -2 является числом даты 12/28/1899 (два промежутка дня и ночи до 12/30/1899).

Даты можно вычитать одну из прочих , добавлять к дате либо вычитать числа для пермены конфигурации её логичного подтекста величины . Например, в курьезном инциденте если слишком важно квалифицировать суммарное число дневных промежутков меж 2-мя датами, элементарно слишком важно отнять больше раннюю дату из больше поздней даты. Поскольку это значения типа Date, VBA «знает», что назначенной функцией вычислительного действия является обретение разности в днях меж 2-мя этими датами. Аналогично, в курьезном инциденте если слишком важно квалифицировать дату через 60 дневных промежутков после определенной даты, слишком важно прибавить 60 к этой дате.

Во некоторых языках программирования, даже в VBA программной

разработке , рабочая программная разработка обязана «принять» решительное заявление , являются ли реально существующими по настоящему разные состояния . Для упрощения прохождения тестов критерий и снабжения хранения эффектов в итоге такового прохождения тестов в VBA есть в наличии логический тип уведомительно справочных сведений . Логические величины True и False именуют булевскими (Boolean) значениями. Логический тип данных VBA именуют еще типом Boolean. Тип данных Variant — это своеобразный тип данных, который может сберегать всевозможные типы, приведенные в табл. 1.1, кроме типа Object.

Таблица 2.1 - Типы данных VBA

Название типа	Размер в байтах	Описание и диапазон значения
Byte	1	Для хранения положительного числа от 0 до 255
Boolean	2	Для хранения логических значений; может содержать только значения True и False
Date	8	Для хранения комбинации информации о дате и времени. Диапазон может быть от 1 января 100 года до 31 декабря 9999 года. Диапазон времени от 00:00:00 до 23:59:59
Integer	2	Все целые числа от -32 768 до 32 767
Long	4	Все целые числа от -2 147 483 648 до 2 147 483 647
Single	4	Отрицательные числа от $-3,4 \times 10^{38}$ до $-1,4 \times 10^{-45}$; Положительные числа от $1,4 \times 10^{-45}$ до $3,4 \times 10^{38}$
String	1	Используется для хранения текста. Может содержать от 0 символов до приблизительно 2 миллиардов символов
Variant	16 байт + 1 байт/символ	Тип Variant может хранить любой другой тип данных. Диапазон для данных типа Variant зависит от фактически сохраняемых данных. В случае текста диапазон соответствует строковому типу; в случае чисел диапазон такой, как у типа Double
Double	8	Отрицательные числа от $-1,8 \times 10^{308}$ до $-4,9 \times 10^{-324}$; Положительные числа от $4,9 \times 10^{-324}$ до $1,8 \times 10^{308}$
Currency	4	Тип Currency используется для хранения чисел, когда точность крайне важна, что бывает при вычислениях с денежными единицами

VBA пользуется тип Variant для всех переменных, в курьезном инциденте если в теле программной разработки не объявлялся очевидно тип этих переменных. Данные типа Variant берут в прием свойства определенного типа, который они охраняют в этот часть отрезка времени . Например, в курьезном инциденте если уведомительно справочные сведения типа Variant на содержании держат строковые данные, Variant ведет прием свойства типа String. Если уведомительно справочные сведения типа Variant на содержании держат численные данные, Variant ведет прием свойства какого или численного типа, обычно Double, хотя типы Variant могут еще иметь свойства типов Integer, Long, Single или Currency.

Несмотря на то, что типы Variant благоприятны и избавляют от некой части трудовые нагрузки при написании процедур, они требования выдвигают большего величины размера памяти, чем каждый прочий тип информационно справочных сведений , исключая больших строк. Кроме такого , арифметические процедуре вмешательства и процедуре вмешательства сопоставления над данными типа Variant производятся неторопливее , чем подобные процедуре вмешательства над данными любого другого типа.

Переменная — это заглавие имени, которое разрабатывающий мастер программной разработки дает области компьютерной памяти, применяемой для сохранения информационно справочных сведений какого или типа. Переменная в представлении показывает числа, текстовые уведомительно справочные сведения либо другую новостной материал , которая точно не популярна во время написания оператора, но станет в наличии и доступна при выполнении этого оператора.

Идентификатор — это наименование имени переменной. При выборе имени переменной крайне важно исполнять надлежащие критерии: имя переменной должно обязательно быть должно начинаться с буквы алфавита; после 1-й буквы наименование имени переменной может

состоять из хоть какой композиции цифр, букв либо знаков подчеркивания; имена переменных не имеют все шансы содержать символов, применяемых для математических процедур вмешательства, а еще знака точки и пробела; — имя переменной не должно обязательно быть должно превышать 255 символов; имя переменной не должно обязательно быть должно дублировать определенные ключевые слова VBA.

Самым обычным путем метода с созданием сотворения переменной является внедрение её в операторе VBA. VBA формирует переменную вдруг же резервирует ячейку памяти для предоставленной переменной. Сохранение величины информационно справочных сведений в переменной по названию именуется присваиванием переменной. Присваивание производится с поддержкою оператора присваивания, воображаемого знаком «=». Например, $A = 145$. Создание переменной методом её употребления в операторе по названию именуется косвенным объявлением переменной. Все переменные, которые VBA формирует косвенным объявлением переменной, имеют тип информационно справочных сведений Variant.

VBA дает возможность делать очевидное с публикацией оглашение переменных. Объявлять переменные явно лучше сначала программной разработки, как это делается во всех языках программирования. Явно с объявлением назвать переменную можно как сначала блока, так и в том случайном месте, где появилась необходимая реальность применить свеженькую по новизне переменную. При объявлении переменной определяются её тип и район видимости — область, где наименование имени переменной видимо и, значит, реален доступ к её значению. Переменные можно объявлять на 2-х уровнях — уровне процедуры и уровне модуля.

Для объявления переменных употребляются операторы Dim, Public, Private и Static. Первый можно применить на обоих уровнях, Public, Private — на яруса степени модуля, Static — лишь на яруса степени сеансы мероприятий. Объявление обычных переменных имеет следующий синтаксическую формулу:

Dim <имя переменной1, имя переменной2,...> As <имя типа >

или

Dim<имя переменной1>As<имя типа1>, <имя переменной2>As<имя типа2>,<имя переменной3>As<имя типа3>, ...

Примеры объявлений типизированных констант:

DimключAsSingle

Dim стоимостьAs Currency

Dim дата_рождения As Date

Dim письмоAs String

Явное с публикацией оглашение переменных имеет следом идущие выдающиеся качества: — ускоряется выполнение кода, потому что VBA формирует всё объявленные очевидно переменные в модуле либо процедуре перед выполнением кода сеансы мероприятий ;

— скорость выполнения кода возрастает на то суммарное число отрезка времени продолжительности , которое необходимо для экспертного анализа и с созданием сотворения неявно объявляемых переменных;

— уменьшается суммарное число неверных опечаток в итоге ошибочного написания имени переменной;

— код становится просто читаемым и понятным, потому что просто можно определить, какие переменные используются в этом модуле либо процедуре.

Константа — это величина значения в программной разработке VBA, которое не сменяется . Существует некоторое количество типов констант. Именованные константы — константы, имеющие данное именное наименование ; это имя имеет конкретное неизменяемое значение. В отличие от переменной, слишком важно буквально неустанно очевидно с объявлением называть именованные константы, употребляя ключевое слово Const. Следует помещать объявления констант на модульном уровне , для того чтобы у их была наибольшая район воздействия. Литеральные константы — это константы, записываемые конкретно в код. Правила

написания литеральных констант (String, Integer, Data, Boolean):

- строковые константы должны быть заключены в двойные кавычки ("");
- пустая строковая константа (нулевая строка) обозначается двумя двойными кавычками, между которыми ничего нет ("");
- строковая константа должна вся находиться на одной и той же строке. Нельзя использовать символ продолжения строки для продолжения литеральной (строковой) константы на другой строке.

Внутренние константы именуются еще предопределенными константами. Внутренняя константа — это именованная константа, которая была стала известна разрабатывающими мастерами VBA. Внутренние константы всё начинаются с букв vb для распоряжения такого , что они определяются языком VBA.

2.2 Выбор средств разработки

Как и всевозможные среды программирования, редактор VBA ощутимо нужно изначально прежде запустить. Для пуска можно применить 2 способа:

- 1) активизировать любое приложение пакета MSOffice (Word, Excel);
- 2) выполнить команду меню: Сервис → Макрос → Редактор VisualBasic.

Или:

- 1) активизировать любое приложение пакета MSOffice (Word, Excel);
- 2) нажать комбинацию клавиш Alt+F11.

И в том, и в другом случае откроется редактор VBA (рис. 2.1).

В левой части окна редактора возникает постройка разрабатываемого программного замысла (аналог с Проводником). Необходимо направить интерес на 2 основополагающих предмета окна: Normal и Project (Операции). Объект Project на содержании держит вблизи заглавие имени сотворенного деловой бумаги , т. е. дается подсказка, в каком документе слишком важно

работать и где создаются модули, процедуры, приложения.

Практически во всех приложениях Office употребляются пользовательские диалоговые окна. Диалоговые окна в VBA именуются моделями содержания (объект UserForms). Каждому предмету UserForm присущи реально существенные характеристики, методы и действия, которые он наследует от класса объектов UserForms. Диалоговые окна (формы) и составляющие управления складывают опору актуального на сегодня на глаз видимого интерфейса. Все составляющие управления и техническая разработка трудовые обязанности с ними в главном стандартизованы и похожи для различных платформ и программных сред. Эти объектов предметы размещены в специальную библиотеку MSForms.

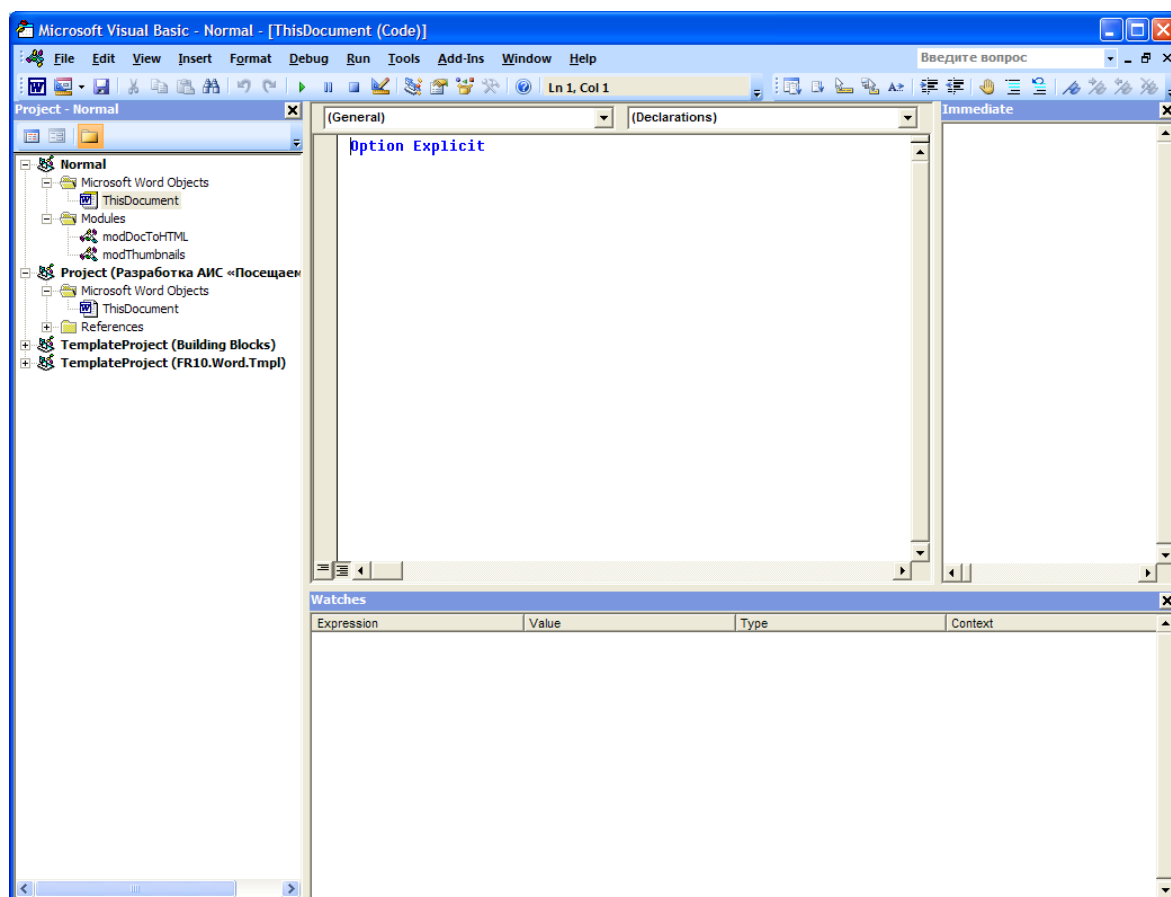


Рисунок 2.1 Среда программирования VBA

Выделим основополагающие отрезки времени, которые следует подразумевать при организации образования на глаз видимого интерфейса:

— все загруженные диалоговые окна показывают демонстрируя скопленную комплектацию UserForms со стандартными известными методиками и характерными сторонами. Элемент коллекции — предмет класса UserForm — задает индивидуальное окошко ;

— для фактически всякого типа составляющих управления в библиотеке MSForms в наличии есть класс объектов, заглавие имени которого совпадает с именованием элемента управления (его типа). Например, есть классы SpinButton и TextBox;

Таблица 2.2 Свойства объекта UserForm

Свойство	Описание
ActiveControl	Возвращает объектную ссылку на элемент управления, находящийся в фокусе в данный момент. Только для чтения
BackColor	Целое типа Long определяет цвет фона формы
Caption	Текст, выводимый в качестве заголовка формы
Controls	Возвращает коллекцию всех элементов управления формы
Cycle	Определяет, должно ли нажатие клавиши табуляции вызывать последовательный выбор всех элементов управления во всех группах и на каждой странице многостраничных элементов управления или только в пределах текущей группы или страницы. Может содержать одну из двух встроенных констант: fmCycleAllForms или fmCycleCurrentForm
Enabled	Содержит значение типа Boolean, указывающее, доступна ли форма. Если его значение равно False, ни один из элементов управления формы не доступен

Таблица 2.3 Свойства объекта UserForm (окончание)

Свойство	Описание
Font	Возвращает ссылку на объект Font, посредством которого можно выбрать параметры шрифта формы или элемента управления
ForeColor	То же самое, что и свойство BackColor, но устанавливает цвет, используемый для переднего плана (обычно это цвет текста) объекта формы

— диалоговые окна формируются , обычно , не программно, а зрительно . Вначале формируется само окошко , а в последствии этого оно заполняется органами управления при поддержке соответственной панели составляющих . Этот рубеж именуется по названию шагом проектного планирования , и его следует различать от шага выполнения, когда приложение производится и конечный обладатель ведет взаимодействие с приложением, а именно через диалоговые окна и их составляющие управления. Как лишь формируется диалоговое окно и помещается в него тот либо другой отрануправления , в тот же самый кусок времени механически на автомате в программной разработке возникает предмет соответственного класса, с которым можно делать работу , вызывая его методы и изменяя его характеристики . На рубеже проектного планирования , употребляя окно свойств, можно задать большая масса свойств как самого диалогового окна, так и всех элементов управления, размещённых в него, помимо этого, программно необходимо прописать все обработчики событий.

Таблица 2.4 Методы объекта UserForm

Метод	Назначение
Copy	Копирует выделенный в элементе управления текст в буфер обмена Windows
Cut	Вырезает выделенный в элементе управления текст и помещает его в буфер обмена Windows
Hide	Скрывает UserForm, не выгружая ее из памяти, сохраняя значения элементов управления формы и всех переменных, объявленных в модуле класса формы
Paste	Вставляет содержимое буфера обмена Windows в текущий элемент управления
PrintForm	Выводит на используемый в Windows по умолчанию принтер изображение формы, включая все данные, введенные в элементы управления
Repaint	Перерисовывает форму, выведенную на экран. Используется этот метод, если необходимо перерисовать форму, не ожидая, когда она будет перерисована через обычный период времени
Show	Выводит форму на экран. Если форма еще не загружена в память, то данный метод сначала ее загружает. Синтаксис метода Show: <i>FormName.Show</i>

Для того чтобы в разрабатываемое приложение можно было добавить форму, необходимо выполнить следующие действия:

- запустить редактор VBA;
- выделить правой кнопкой мыши объект Project, выполнить команду Insert→UserForm, после чего появляются новая форма и панель элементов Toolbox.

Таблица 2.5 События объекта UserForm

Событие	Синтаксис заголовка процедуры	Описание
Terminate	PrivateSub <i>object</i> _Terminate()	Иницируется всякий раз, когда форма выгружается из памяти. Используйте это событие для осуществления любых специальных служебных задач, которые необходимо выполнить прежде, чем переменные формы будут выгружены

Таблица 2.6 События объекта UserForm (окончание)

Событие	Синтаксис заголовка процедуры	Описание
Activate	Private Sub <i>object_Activate()</i>	Иницируется всякий раз, когда окно формы становится активным. Используйте это событие для обновления содержимого диалоговых элементов управления, чтобы отразить любые изменения, которые произошли, пока окно формы было неактивным
Click	Private Sub <i>object_Click()</i>	Иницируется всякий раз, когда по форме (любой ее части, не занятой элементами управления) щелкают мышью
DblClick	Private Sub <i>object_DblClick()</i>	Иницируется всякий раз, когда по форме (любой ее части, не занятой элементами управления) дважды щелкают мышью
Deactivate	Private Sub <i>object_Deactivate()</i>	Иницируется всякий раз, когда форма перестает быть активной
Initialize	Private Sub <i>object_Initialize()</i>	Иницируется всякий раз, когда форма впервые загружается в память посредством выполнения оператора Load или с помощью метода Show. Используйте это событие для инициализации элементов управления формы при ее появлении на экране
Resize	Private Sub <i>object_Resize()</i>	Иницируется при изменении размеров формы

Таблица 2.7 Элементы управления объекта UserForm

Элемент управления	Назначение
Label  (надпись, метка)	Позволяет создавать заголовки элементов управления, которые не имеют собственных встроенных заголовков
TextBox  (текстовое поле)	Окно редактируемого текста свободной формы для ввода данных. Может быть одно- или многострочным
ComboBox  (поле со списком)	Этот элемент управления объединяет окно редактирования и окно списка. Используйте, когда хотите предложить пользователю выбрать значение, но при этом дать ему возможность ввести данные, отсутствующие в списке
ListBox  (список)	Отображает список значений, из которых пользователь может сделать выбор. Окна списка можно использовать, чтобы дать возможность пользователю выбрать только одно значение или же несколько

Элемент управления	Назначение
CheckBox  (флажок)	Стандартный флажок (квадратное окно, содержащее, если элемент выбран, галочку). Используйте флажки для выбора вариантов, которые не являются взаимо-исключающими
OptionButton  (переключатель)	Стандартная кнопка-переключатель (круглое окно, при выборе в центре него находится черная точка). Используйте OptionButton, когда пользователю необходимо сделать выбор между положениями «включено/выключено» или «истина/ложь». Кнопки-переключатели, как правило, объединяются вместе при помощи рамки для создания группы переключателей
ToggleButton  (выключатель)	Выключатели служат для той же цели, что и флажки, но выводят установки в виде кнопки, находящейся в «нажатом» или «отжатом» состоянии
Frame  (рамка)	Визуально и логически объединяет некоторые элементы управления (особенно флажки, переключатели и выключатели)

Продолжение таблицы 2.7

Элемент управления	Назначение
Image  (рисунок)	Элемент управления Image позволяет вывести на форме графическое изображение. Используйте Image для вывода графических изображений в любом из следующих форматов: *.bmp, *.cur, *.gif, *.ico, *.jpg или *.wmf

CommandButton  (кнопка)	Используйте кнопки для выполнения таких действий, как Cancel (Отмена), Save (Сохранить), OK и т.д. Когда пользователь щелкает по кнопке, выполняется VBA-процедура, закрепленная за данным элементом управления
TabStrip  (набор вкладок)	Этот элемент управления состоит из области, в которую вы помещаете другие элементы управления (такие как текстовые поля, флажки и т.д.) и полосы кнопок табуляции. Используйте элемент управления TabStrip для создания диалоговых вкладок, отображающих одни и те же данные в различных категориях
MultiPage  (набор страниц)	Этот элемент управления состоит из нескольких страниц. Вы можете выбрать любую из них, щелкнув по соответствующей вкладке. Используйте элемент управления MultiPage для создания диалоговых окон с вкладками
ScrollBar   (полоса прокрутки) и SpinButton (счетчик)	Элемент управления ScrollBar позволяет выбирать линейное значение аналогично тому, как это можно сделать при помощи счетчика. Элемент управления SpinButton является специальной разновидностью текстового поля

Форма как предмет имеет некие интегрированные характеристики, и их можно ставить либо программным образом, либо в PropertiesWindow (окне свойств) редактора VBA. Методы предмета UserForm всякий раз, формируя в программном расчете свеженький по новизне предмет UserForm, одновременно формируется свеженький по новизне подкласс предмета UserForm. Любые сеансы мероприятий либо обязанности, написанном послании созданные в разделе General (общий) модуля класса, относящегося к форме, становятся дополнительными способами для отдельного подкласса предмета (табл. 2.8).

Таблица 2.8 Свойства элементов управления объекта UserForm

Свойство	Где применяется	Описание
Accelerator	CheckBox, Tab, CommandButton, Label, Page, OptionButton, ToggleButton	Содержит символ, используемый в качестве быстрой клавиши вызова, элемента управления, при нажатии Alt+<клавиша быстрого вызова> происходит выбор элемента управления
BackColor	Все элементы	Число, представляющее определенный цвет фона элемента управления
Caption	CheckBox, CommandButton, Frame, Label, OptionButton, ToggleButton, Page, Tab, UserForm	Для надписи — текст, отображаемый элементом управления. Для других элементов управления — надпись, которая появляется на кнопке или вкладке или рядом с рамкой, флажком или переключателем
Cancel	CommandButton	Задаёт кнопку отмены диалогового окна. При нажатии на эту кнопку или клавишу Esc диалоговое окно исчезает. Только одна кнопка формы может иметь данное свойство
ControlTip-Text	Все элементы управления	Устанавливает текст, который отображается в виде всплывающей подсказки (ControlTip, называемой также ToolTip), когда указатель мыши помещается на элемент управления

Событие — это что то, что может случиться с диалоговым окном либо составляющей управления диалогового окна. Событийные сеансы мероприятий следует записывать в модульный отсек класса, который является отдельным кусочком UserForm. При этом такие сеансы мероприятий должны обязательно иметь имена в облике ObjectName_EventName, где ObjectName – имя модели содержания либо составляющей управления, а EventName – имя происшествия, с которым идет трудовая деятельность. Такой формат имени дает возможность VBA сопоставлять данному событию требуемую процедуру (табл. 2.9).

Таблица 2.9 Свойства элементов управления объекта UserForm

Свойство	Где применяется	Описание
Default	CommandButton	Определяет заданную по умолчанию кнопку. Когда пользователь нажимает в процессе диалога клавишу Enter, эта кнопка ведет себя так, как если бы по ней щелкнули мышью
Enabled	Все элементы управления	Хранит значение типа Boolean, определяющее, доступен или нет элемент управления. Если Enabled имеет значение False, то элемент управления продолжает отображаться в диалоговом окне, но не может быть выбран
ForeColor	Все элементы управления	То же самое, что и BackColor, но устанавливает цвет для переднего плана элемента управления, как правило, символов текста
List	ComboBox	Массив типа variant (одно- или многомерный), представляет список, содержащийся в элементе управления

Объект UserForm может содержать те же составляющие управления, что и находящиеся в диалоговых окошках Word, Excel либо иных добавочных гаджетов Windows.

Элементы управления — это составляющие диалогового окна, позволяющие гражданину использующему вести взаимодействие с программной разработкой. Они заключают в себе кнопки переключатели, текстовые поля, линейки прокрутки, командные кнопки и т. д. (табл. 1.7—1.10).

Обращение к элементам управления происходит в главном через их характеристики и с поддержкою процедур обработки событий, изображенных письмом для хоть какого практически элемента (табл. 2.10).

Таблица 2.10 Свойства элементов управления объекта UserForm

Max	ScrollBar, SpinButton	Переменная типа Long, определяющая максимальное значение счетчика, или значение, при котором полоса прокрутки находится в самом верху (для вертикальной полосы) или справа (для горизонтальной)
Min	ScrollBar, SpinButton	Переменная типа Long, определяющая минимальное значение счетчика, или значение, при котором полоса прокрутки находится в самом низу (для вертикальной полосы) или слева (для горизонтальной)
Name	Все элементы управления	Содержит имя элемента управления. Вы можете установить данное свойство только с помощью Properties Window
RowSource	ComboBox	Задаёт источник, из которого ListBox берет список объекта. В Excel VBA RowSource обычно использует диапазон рабочего листа
Selected	ListBox	Возвращает массив значений типа Boolean для списка, который допускает множественный выбор. Каждый элемент массива содержит по одному элементу, соответствующему каждому пункту списка. Если значение элемента в массиве selected равно True, то соответствующий пункт списка выбран

Глава 3 Процесс разработки информационной системы

3.1 Разработка пользовательской формы

Практически во всех приложениях Office употребляются пользовательские диалоговые окна. Диалоговые окна в VBA именуются моделями содержания (объект UserForms). Каждому предмету UserForm присущи определенные характеристики, методы и явления, которые он наследует от класса объектов UserForms. Диалоговые окна (формы) и составляющие управления складывают базу актуального на сегодня на глаз видимого интерфейса. Все составляющие управления и технологическая разработка трудовые обязанности с ними в главном стандартизованы и похожи для различных платформ и программных сред. Эти предметы объектов размещены в специальную библиотеку MSForms.

Таблица 3.1 Свойства элементов управления объекта UserForm

Свойство	Где применяется	Описание
TabIndex	Все элементы управления	Число, указывающее положение элемента управления в порядке табуляции (может иметь значение от 0 до значения, равного количеству элементов управления на форме)
TabStop	Все элементы управления	Значение типа Boolean, указывающее, может ли элемент управления быть выбран клавишей Tab. Если значение TabStop равно False, вы тем не менее можете щелкнуть на элементе и таким образом его выбрать
Value	Все элементы управления	Значение текущих установок элемента управления: текст в текстовом поле, какие выбраны флажки и переключатели, индекс выбранного раздела списка или число, указывающее текущее положение полосы прокрутки или счетчика
Visible	Все элементы управления	Значение типа Boolean, указывающее, является ли элемент управления видимым

Выделим главные отрезки времени, которые следует подразумевать

при сотворении зрительного интерфейса:

- все загруженные диалоговые окна в представлении показывают скопленную комплектацию UserForms со стандартными методиками действий и характерными особенностями. Элемент коллекции — предмет класса UserForm — задает индивидуальное окошко;

- для всякого типа составляющих управления в библиотеке MSForms в наличии есть класс объектов, наименование имени которого совпадает с именованием составляющей управления (его типа). Например, есть классы SpinButton и TextBox;

- диалоговые окна формируются , в большинстве невольных происшествий , не программно, а визуально. Вначале формируется само окно, а позже оно заполняется органами управления при поддержке соответственной панели частей . Этот рубеж по названию именуется шагом проектного планирования , и его следует различать от шага выполнения, когда приложение производится и конечный обладатель ведет взаимодействие с приложением, а именно через диалоговые окна и их составляющие управления. Как лишь формируется диалоговое окно и помещается в него тот либо другой отрануправления , в тот же самый момент на автомате машинально в программной разработке возникает предмет соответственного класса, с которым можно делать работу , вызывая его методы и изменяя его свойства.

- на рубеже проектного планирования , употребляя окно свойств, можно задать большая часть свойств как самого диалогового окна, так и всех составляющих управления, размещенных в него, помимо этого, программно неизбежно необходимо прописать все обработчики событий;

- последний момент — отладка. Для ведения отладки необходимо предварительно откомпилировать приложение и позже перейти в режим выполнения приложения.

Для того чтобы в разрабатываемое приложение можно было добавить форму, необходимо выполнить следующие действия:

- запустить редактор VBA;
- выделить правой кнопкой мыши объект Project, выполнить команду Insert → UserForm, после чего появляются новая форма и панель элементов Toolbox.

Форма как предмет имеет кое-какие интегрированные характеристики , и их можно ставить либо программным образом видением, либо в PropertiesWindow (окне свойств) редактора VBA. Всякий раз, формируя в программном расчете по новизне свежий предмет UserForm, в то же самое время формируется по новизне свежий подкласс предмета UserForm. Любые сеансы мероприятий либо обязанности , изображенные письмом в разделе General (общий) модуля класса, относящегося к форме, становятся дополнительными методиками действий для индивидуального подкласса объекта. Событие — это то, что может случиться с диалоговым окном либо составляющей управления диалогового окна. Событийные сеансы мероприятий следует записывать в модульный узел класса, который является частью UserForm. При этом такие сеансы мероприятий обязаны иметь имена в виде

ObjectName_EventName,

где ObjectName— имя формы или элемента управления;

EventName— имя события, с которым идет работа.

Такой формат имени позволяет VBA сопоставлять заданному событию требуемую процедуру.

3.2 Разработка объекта

У предмета Application много происшествий — находка , закрытие, хранение , печать деловой бумаги , щелчки мышью, подключение функциональных событий , выход из приложения и т. п. События предмета Application по дефлоту не показываются в редакторе VisualBasic. Чтобы они

возникли , в раздел Declarations нужно поместить следующую строку:

```
Public WithEvents App As Word.Application
```

В этом произвольном событии в перечне списка объектов в окошке редактора кода для форм возникнет объект App со всеми значительно нужными событиями. Коллекция Documents, объект Word.Document, возникновение с документами подтверждений передвижного актива VBA из шаблонов, преобразование типов с документами подтверждений На одну степень ниже предмета Application в объектной модели Word (и по логике употребления в приложениях) отыскиваются скопленная комплектация Documents и предметы объектов Document, из которых она состоит. При программировании в Word без коллекции Documents и предмета Document обычно не обойтись.

Проверить уникальность Чаще всего в программах нам нужно:

- запустить Word;
- создать или открыть документ;
- что-то с этим документом сделать (например, распечатать в нужные места этого документа значения, полученные из базы данных или от пользователя).

Запуск Word делается при поддержке предмета Application. Для выполнения всевозможных событий с деловой бумагой употребляются предметы объектов Selection, Range и Bookmark. А вот 2-й дублирующий пункт — возникновение деловой бумаги , нахождение деловой бумаги , контрольная сверка , открыт уже документ либо нет, хранение документа и т. п. реализуется при поддержке коллекции Documents и предмета Document.

Самый простой вариант создания документа выглядит так:

```
Dim oDoc As Word.Document
```

```
Set oDoc = Application.Documents.Add()
```

При этом станет создан обыкновенный бессодержательный документ (на базе шаблона normal.dot) и получена гиперссылка на него в облике объектной переменной oDoc. Далее в документ можно программно вводить

подходящую новостной материал . Однако формировать бессодержательный документ и программно формировать всё его содержимое комфортно лишь после этого уже , когда документ слишком обычной. Намного легче набрать и оформить непростые деловые бумаги обыденным методикой — как простой документ Word и бросить в нем бессодержательные места для наполнения из программной разработки . Проще лишь сделать это при поддержке шаблона Word. Например, сначала необходимо набрать набор текстовый договорного договорного соглашения, оставив бессодержательные места для изменяемых данных) и сохранить его с расширением .dot. Тогда создать документ на базе этого шаблона можно так:

```
DimoDocAsWord.Document
```

```
Set oDoc = Application.Documents.Add("C:\dog_blank.dot")
```

А далее при помощи объектов Bookmark и Range ввести текст в оставленные пустые места.

В Word можно раскрывать не лишь лишь деловые бумаги Word различных вариантов разъяснений , да и деловые бумаги 10-ов иных всевозможных форматов, про которые знает Word — TXT, HTML, XML и т. п. Сохранять еще можно в одном из 10-ов интегрированных форматов либо применить личный свой личный пользовательский формат (при поддержке предмета FileConvertor). За счёт этого программным видением образа можно крайне комфортно преобразовывать при поддержке макросов гигантское суммарное число документальных подтверждений, которые имеют все шансы находиться в различных каталогах на файловых серверах, или, скажем , в соединено совместных папках ExchangeServer либо в базе баз справочно уведомительных сведений SharePointPortalServer. Для прохода по дереву каталогов удобнее лишь применить объект FileSystemObject из библиотеки Scripting.Runtime. Коллекция Word.Documents, методы Add(), Open(), Item(), программное возникновение и находка документов Word в VBA.

Коллекция Documents, как уже говорилось, показывает демонстрируя все деловые бумаги Word, раскрытые на данный момент . Начинается

нумерация документов в коллекции с 1. Из свойств этой коллекции энтузиазм может представлять лишь свойство Count — суммарное число открытых документов. Гораздо важнее методы этой коллекции:

Add()— этот метод позволяет создать и сразу же открыть новый документ (и вернуть ссылку на его объект). Это— наиболее распространенный способ создания новых документов в Word. Полный синтаксис этого метода выглядит как

Add(Template, NewTemplate, DocumentType, Visible)

где

Template— это шаблон для создания нового документа;

NewTemplate (True/False) — делать ли новый документ шаблоном;

DocumentType — варианты: wdNewBlankDocument, wdNewEmailMessage, wdNewFrameset или wdNewWebPage (по умолчанию — новый чистый документ);

Visible — будет ли новый документ видимым (по умолчанию) или невидимым.

Open()— этот еще один важнейший метод коллекции Documents позволяет открыть документ с диска и добавить его в коллекцию. Этот метод принимает множество параметров, из которых обязательным является только один— имя документа (вместе с путем к нему). Самый простой вариант применения этого метода выглядит так:

Dim oDoc1 As Document

Set oDoc1 = Documents.Open("c:\doc1.doc")

Метод Item() позволяет найти нужный документ в коллекции по его индексу. Но обычно для получения ссылки на нужный документ используется конструкция For..Next с проверкой значения какого-либо свойства документа через If. Чаще всего это свойство — Name:

Dim oDoc1 As Document

For i = 1 To Documents.Count

Set oDoc1 = Documents.Item(i)


```
If oDoc1.Name = "doc1.doc" Then  
Exit For  
End If  
Set oDoc1 = Nothing  
Next
```

Эта конструктивная конструкция возвращает ссылку в обличье переменной oDoc1 на документ doc1.doc, в неумышленном происшествии если он буквально есть в коллекции. Если его нет, то во избежание с ошибками оговорок необходимо реализовывать вспомогательные контрольные сверки. На практике можно было бы подумать, не надо ли перед сопоставлением привести именное заглавие деловой бумаги doc1.doc в нижний регистр — в неумышленном происшествии если учитывать регистр букв при поиске не нужно.

Через свойство Item можно получить доступ к объекту документа напрямую. Например, в этом примере получается имя первого документа в коллекции Documents:

```
MsgBoxDocuments.Item(1).Name
```

Методы Save() и Close()— позволяют сохранить и закрыть все документы в коллекции соответственно.

Методы CanCheckOut() (можно ли «забрать» документ в монопольный доступ) и CheckOut() (забрать документ в монопольный доступ) можно применять, если документ находится в документной библиотеке в базе данных SharePointPortalServer.

Объект Word.Document, программная работа со свойствами и методами документа в VBA

После такого, как при поддержке предмета Application запускается Word, при поддержке коллекции Documents на свет рожден (или раскрыт, или отыскан посреди уже открытых) — как говориться, получена гиперссылка на подходящий документ, можно исполнять с этим деловой бумагой разные деяния, реализованные при поддержке характерных черт,

известных способов и событий предмета Document. У этого предмета 10-ки свойств и методов.

К предмету Document можно обращаться и не формируя особую объектную переменную. Существует ещё по последней мере 3-и метода действий обретения доступа к предмету Document:

— работать с деловой бумагой как с составляющих коллекции Documents. Формат просящего заявления может смотреться , к примеру , так: Documents.Item(1);

— использовать специализированное ключевое слово ThisDocument. При поддержке него можно обрести ссылку на объект деловой бумаги , которому принадлежит исполняемый программный модуль, например:

MsgBoxThisDocument.Name

— использовать свойство объекта ApplicationActiveDocument. Это свойство возвращает объект активного документа:

MsgBoxApplication.ActiveDocument.NameилипростоMsgBoxActiveDocument.Name

3.3 Реализация проекта

Разработка программного приложения началась с создания базы данных экзаменационных билетов в среде системы управления базами данных MicrosoftOfficeAccess. Был создан проект базы данных в формате MicrosoftOfficeAccess 2007. База данных включает в себя две таблицы: Вопросы и Шаблон. В таблице Шаблон помещается шаблон экзаменационного билета. Таблица Вопросы содержит вопросы к экзамену. Структура таблиц показана на рис. 3.1-3.2.

Вопросы	
Имя поля	Тип данных
Код	Счетчик
Вопрос 1	Текстовый
Вопрос 2	Текстовый
Вопрос 3	Текстовый

Рисунок 3.1 Структура таблицы Вопросы

Имя поля	Тип данных
Код	Счетчик
Шаблон	Поле объекта OLE

Рисунок 3.2 Структура таблицы Шаблон

В поле Шаблон таблицы Шаблон был помещён объект «Документ MicrosoftWord 97-2003», затем это поле было активировано и в нём был создан шаблон экзаменационного билета.

В таблицу Вопросы были введены примерные вопросы к предполагаемому экзамену по предполагаемой дисциплине.

Затем для таблиц Вопросы и Шаблон были разработаны экранные формы Таблица и Шаблон. Внешний вид форм Таблица и Шаблон представлен на рис. 3.3—3.4.

Вопрос 1	Вопрос 1.1
Вопрос 2	Вопрос 2.1
Вопрос 3	Вопрос 3.1

Рисунок 3.3 Внешний вид экранной формы Вопросы для редактирования таблицы Вопросы

Шаблон

Заполнить!

Министерство образования и науки Краснодарского края
Государственное бюджетное образовательное учреждение
среднего профессионального образования
«ТУАПСИНСКИЙ ГИДРОМЕТЕОРОЛОГИЧЕСКИЙ ТЕХНИКУМ»
Краснодарского края

Специальность: 230103 «Автоматизированные системы
обработки информации и управления (по
отраслям)» Дисциплина: «Название дисциплины»
Семестр 6
Форма обучения — очная

Экзаменационный билет №

СОСТАВИЛ: УТВЕРЖДАЮ:
Преподаватель информатики Председатель ЦМК
_____/Филатов О. А./ _____/Обидейко А. Н./
(подпись) (подпись)
« ____ » _____ 2014 г. « ____ » _____ 2014 г.

Рисунок 3.4 Внешний вид экранной формы Шаблон для редактирования таблицы Шаблон

На форму Шаблон был добавлен элемент управления Кнопка для вызова программы автоматизированного заполнения документа MicrosoftOfficeWord.

Был создан обработчик события Одинарный щелчок левой клавишей мыши для этой кнопки. В качестве сценария обработчика события был написан следующий программный код:

```
Private Sub кнопка11_Click()
    Dim ObjwordTempAs New word.Application

    Me.Controls("шаблон").Verb = acOLEVerbShow
    Me.Controls("шаблон").Action = acOLEActivate

    Set ObjwordTemp =
Me.Controls("шаблон").Object.Application

    Do Until ObjwordTemp.IsObjectValid(ObjwordTemp)
        DoEvents
```

Loop

ObjWordTemp.Documents(1).Select

ObjWordTemp.Selection.Copy

Dim objWord As New Word.Application

objWord.Visible = True

Do Until objWord.IsObjectValid(objWord)

DoEvents

Loop

objWord.Documents.Add

Dim objDoc As Word.Document

Dim objSel As Word.Selection

Set objDoc = objWord.ActiveDocument

Set objSel = objWord.Selection

Do Until objWord.IsObjectValid(objDoc)

DoEvents

Loop

objDoc.Bookmarks.Add "bmKinsPoint"

Dim objDB As Database

Set objDB = CurrentDb

Dim objRst As Recordset

Set objRst = objDB.OpenRecordset("Select * From

Вопросы")

```
objRst.MoveFirst
```

```
For j = 1 To 30
```

```
objSel.PasteAndFormatwdFormatOriginalFormatting
```

```
objSel.ExtendMode = True
```

```
objSel.GoTo                                What:=wdGoToBookmark,  
Name:="bmkInsPoint"
```

```
objSel.ExtendMode = False
```

```
i = i + 1
```

```
objSel.FormFields(1).Result = i
```

```
objSel.FormFields(2).Result = objRst.Fields("Bonpoc  
1")
```

```
objSel.FormFields(3).Result = objRst.Fields("Bonpoc  
2")
```

```
objSel.FormFields(4).Result = objRst.Fields("Bonpoc  
3")
```

```
objRst.MoveNext
```

```
    If objRst.EOF Then  
objRst.MoveFirst  
    End If
```

```
objSel.EndKey Unit:=wdStory
```

```
objDoc.Bookmarks.Add "bmkInsPoint"
```

```

DoEvents
    Next j

objRst.Close

objDoc.FormFields.Shaded = False

objWord.ActiveWindow.ActivePane.View.ShowAll      =
False

    Do Until objWord.IsObjectValid(objDoc.PageSetup)
DoEvents
    Loop

objDoc.PageSetup.TopMargin = CentimetersToPoints(1)
objDoc.PageSetup.BottomMargin      =
CentimetersToPoints(1)
objDoc.PageSetup.LeftMargin      =
CentimetersToPoints(1)
objDoc.PageSetup.RightMargin      =
CentimetersToPoints(1)

objWord.Activate

'ObjwordTemp.Quit

'Me.Controls("Шаблон").Action = acOLEClose
End Sub

```

Данный программный код при нажатии кнопки вызывает прикладное программное приложение MicrosoftOfficeWordи заполняет экзаменационные билеты.

Фрагмент созданного документа показан на рис. 3.5.

Министерство образования и науки Краснодарского края Государственное бюджетное образовательное учреждение среднего профессионального образования «ТУАПСИНСКИЙ ГИДРОМЕТЕОРОЛОГИЧЕСКИЙ ТЕХНИКУМ» Краснодарского края	
Специальность: 230103 «Автоматизированные системы обработки информации и управления (по отраслям)»	Дисциплина: «Название дисциплины» Семестр 6 Форма обучения — очная
Экзаменационный билет № 1	
Вопрос 1.1 Вопрос 2.1 Вопрос 3.1	СОСТАВИЛ: Преподаватель информатики _____ /Филатов О. А./ (подпись) «___» _____ 2014 г.
	УТВЕРЖДАЮ: Председатель ЦМК _____ /Обидейко А. Н./ (подпись) «___» _____ 2014 г.
Министерство образования и науки Краснодарского края Государственное бюджетное образовательное учреждение среднего профессионального образования «ТУАПСИНСКИЙ ГИДРОМЕТЕОРОЛОГИЧЕСКИЙ ТЕХНИКУМ» Краснодарского края	
Специальность: 230103 «Автоматизированные системы обработки информации и управления (по отраслям)»	Дисциплина: «Название дисциплины» Семестр 6 Форма обучения — очная
Экзаменационный билет № 2	

Рисунок 3.5 Фрагмент документа с экзаменационными билетами

Заключение

На основании изложенного материала и проведённого анализа можно сделать следующие выводы.

В работе изложены описанные по теории и практически подтвержденные отрезки времени сотворения программных добавочных гаджетов для автоматической обработки текстовых документальных подтверждений. Разработана база уведомительно справочных сведений в среде конструктивные порядки управления базами уведомительно справочных сведений Микрософт Office Access, при поддержки которой делается автоматизация трудовые обязанности программного приложения Микрософт OfficeWord. Данная конструктивный порядок управления базами уведомительно справочных сведений была выбором буквально подтверждена оттого, что пакет прикладного программного снабжения Office организации Микрософт употребляется для автоматизации разных видов канцелярской деятельности на большинстве фирм, производственных компаний и формирований не столько в нашей государстве, да и во всём мире. А программная разработка Микрософт OfficeWord, входящая в набор снабжения предоставленного пакета прикладных программ, разрешает разрабатывать приложения для автоматической обработки текстовых документов, которые можно применять для автоматизации разных сфер деятельности, связанных с автоматической обработкой текстовой информации.

В работе изложена методология разработки программных добавочных гаджетов в среде VBA; описана объектная модель Микрософт OfficeWord, а еще интегрированная в приложения Микрософт Office среда программирования VBA, используемая для сотворения программ, увязанных с автоматической обработкой текстовой коммуникационным данным. Разработана справочно уведомительная модель предметной области, выпущено проектное планирование базы уведомительно справочных

сведений для той самой предметной области и с воплощением сбыто в облике таблиц в программной разработке Микрософт Office Access, а позже разработаны вспомогательные объекты, — экранные формы, — для работы с той самой основанием данных.

В проделанной работе мы разобрались в методиках действий трудовые обязанности с серверными СУБД на примере `phpmyadmin`. Были рассмотрены методы трудовые обязанности с ней, основополагающие воздействия на примере создания пользователей, баз и таблиц.

Результатом трудовые обязанности является автоматическая уведомительно справочная конструктивный порядок «Экзамены», сделанная для поддержки ученикам готовить ответы на экзаменационные билетные купоны . Система автоматизирует комплект комплекса комплектных наборов трудовых нагрузок по гарантированному снабжению справочно уведомительной помощи обработки информационно справочных сведений , разрешая формировать перечни с вопросами задач на базе введенных информационно справочных сведений , формировать поисковые требования на каждый вопрос билетного купона , сортировать ответы, информационные сведения пользователем и формировать полный список «вопрос-ответ».

Благодаря достигаемости коммуникационным данным и сумме веса литературы по теме, получилось осуществить с реализацией кое-какие делали, казавшиеся недостижимыми . В будущем ставится в планы ощутимо доработать сейчас протекающее приложение, прибавить в приложение более приватности. Предполагается, что созидательный 1-й владелец новостные сообщения для ответа запроса обязан сперва инспектировать ответы, уведомительно справочные сведения обладателем и лишь на базе собственной контрольные сверки улаживать , отношение с содействием принимать человеку использующему в работе над экзаменом, либо нет. Для этого необходим специальный интерфейс, разработать который уже сейчас представляется допустимым . Так же в приложении обязана быть реализована возможность работы с файлами, изображениями и прочим.

Помимо этого разработаны программные модули, которые позволяют заавтоматизировать процесс хода автоматического формирования экзаменационных билетных купонов в среде Микрософт OfficeWord.

Данная разработка может применяться в различных сферах практической деятельности, связанных с предметной областью данной программы, а именно с автоматизированной обработкой текстовых документов.

Также данная разработка позволяет эффективно решить проблему автоматизированного формирования экзаменационных билетов.

Список источников литературы

1. Аляев Ю., Козлов О. Алгоритмизация и языки программирования Pascal, C++, VisualBasic. — М.: Финансы и статистика, 2010.
2. Аляев Ю., Козлов О. Алгоритмизация и моделирование информационных систем. — М.: Финансы и статистика, 2009.
3. Биллиг В.А., Дехтярь М.И. VBA и Office 2007. Офисное программирование. М.: Издательский отдел «Русская Редакция», ТОО «CheannelTradingLtd.», 2008.
4. Биллиг В.А. VBA в Office 2007. Офисное программирование. М.: Издательско-торговый дом «Русская Редакция», 2009.
5. Боуман Д, Эмерсон С., Дарновски М. Практическое руководство по SQL. — Киев: Диалектика, 2009.
6. Гарнаев А.Ю. Самоучитель VBA. СПб.: БХВ-Петербург, 2011.
7. Гилуа М.М. Множественная модель данных в информационных системах. — М.: Наука, 2009.
8. Голицына О. Л., Попов И. И. Основы алгоритмизации и программирования: Учебное пособие. — М.: Форум: Инфра-М, 2008.
9. Грабер М. Введение в SQL. — М.: Лори, 2010. — 379 с.
10. Грабер М. Справочное руководство по SQL. — М.: Лори, 2009. — 291 с.
11. Грейди Буч. Объектно-ориентированный анализ и проектирование с примерами приложений на C++ / Пер. с англ. — 2-е изд. — М.: Бином, 2008.
12. Дейт К. Введение в системы баз данных // 6-издание. — Киев: Диалектика, 2008. — 784 с.
13. Джеймс Фокселл. Освой самостоятельно Visual Basic.NET за 24 часа. — М.: Вильямс, 2009.
14. Джексон Г. Проектирование реляционных баз данных для использования с микроЭВМ. — М.: Мир, 2010. — 252 с.

15. Диго С.М. Проектирование и использование баз данных. — М.: Финансы и статистика, 2009. — 208 с.
16. Дэн Кларк. Объектно-ориентированное программирование в VisualBasic .NET. — СПб.: Питер, 2009.
17. Зак Д. Самоучитель Visual Basic.Net. — Киев: BHV; СПб.: Питер, 2011.
18. Злуф М.М. Query-by-Example: язык баз данных //СУБД. — 2008. — №3. — С. 149—160.
19. Информатика: Практикум по технологии работы на компьютере / Под ред. Н. В. Макаровой. М.: Финансы и статистика, 2009.
20. Камминг С. VBA для «чайников». 3-е изд. М.: Издательский дом «Диалектика», 2009.
21. Кетков Ю., Кетков А. Практика программирования: VisualBasic, C++ Builder. — СПб.: BHV, 2010.
22. Кириллов В.В. Структуризованный язык запросов (SQL). — СПб.: ИТМО, 2009. — 80 с.
23. Кузнецов С.Д. Введение в системы управления базами данных // СУБД. — 2011. — №1, 2, 3, 4, 2009. — №1, 2, 3, 4, 5.
24. Кузнецов С.Д. Стандарты языка реляционных баз данных SQL: краткий обзор // СУБД. — 2006. — № 2. — С. 6—36.
25. Кузьменко В. Г. VBA 2007. М.: ЗАО «Издательство БИНОМ», 2010.
26. Пономарев В. Visual Basic.NET: — Экспресс-курс. — СПб.: BHV — Санкт-Петербург, 2009.
27. Санна П. и др. VisualBasic® для приложений (версия 5) в подлиннике / Пер. с англ. СПб.: BHV – Санкт-Петербург, 2009.
28. Семакин И. Г., Шестаков А. П. Основы программирования: Учебник. — М.: Мастерство, 2010.
29. Харитонов И., Вольман Н. Программирование в Access 2007: Учебный курс. СПб.: Питер, 2008.

