

Профессор
Игорь Н. Бекман

КОМПЬЮТЕРНЫЕ НАУКИ

Курс лекций

Лекция 10. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Содержание

1. ПРИКЛАДНЫЕ ПРОГРАММЫ	1
2. ПАКЕТЫ ПРИКЛАДНЫХ ПРОГРАММ	3
3. ИНСТРУМЕНТАЛЬНАЯ СРЕДА ПОЛЬЗОВАТЕЛЯ	6
4. ПРИМЕРЫ ПРИКЛАДНЫХ КОМПЬЮТЕРНЫХ СИСТЕМ	12
4.1 Управление базами данных	12
4.2 Офисное приложение	13
4.3 Microsoft Office	15
4.4 Microsoft Word	20
4.5 Microsoft Excel	21
4.6 Microsoft PowerPoint	23
4.7 Microsoft Access	24
4.8 Microsoft FrontPage	24
4.9 Microsoft Outlook	24

Прикладная программа - в широком смысле - программа или пакет прикладных программ, реализующие обработку данных на компьютере в определенной области применения. Прикладная программа непосредственно выполняет функции, необходимые пользователю. Прикладная программа - в узком смысле - программа, решающая проблему конечного пользователя.

В настоящей лекции мы рассмотрим понятие прикладной программы, наиболее употребительные прикладные программы и инструментальную среду пользователя. В заключительной части лекции приведены примеры прикладных компьютерных систем (Office, Word, Excell, Power Point и др.).

1. ПРИКЛАДНЫЕ ПРОГРАММЫ

В соответствии с принципом программного управления любой компьютер можно рассматривать как совокупность аппаратной (или технической) и программной частей.

Прикладная программа (*application program*) - программа, описывающая процесс выполнения определенной задачи для нужд пользователей.

К прикладному программному обеспечению (*application software*) относятся программы, написанные для пользователей или самими пользователями, для задания компьютеру конкретной работы. Программы обработки заказов или создания списков рассылки - пример прикладного программного обеспечения. Программистов, которые пишут прикладное программное обеспечение, называют прикладными программистами. Прикладная программа - в широком смысле - программа или пакет прикладных программ, реализующие обработку данных в определенной области применения. Прикладная программа непосредственно выполняет функции, необходимые пользователю. Прикладная программа - в узком смысле - программа, решающая проблему конечного пользователя.

Прикладное программное обеспечение - программное обеспечение, состоящее из: отдельных прикладных программ и пакетов прикладных программ, предназначенных для решения различных задач пользователей; и автоматизированных систем, созданных на основе этих (пакетов) прикладных программ.

Прикладные программы являются главными компонентами информационной системы и сети. В соответствии с ними функционируют прикладные процессы.

Рассматриваемые программы делятся на две группы.

Первая группа - программы массового использования, именуемые также приложениями. Вторую группу образуют программы индивидуального применения. Они разрабатываются программистами, работающими совместно с соответствующими специалистами. Особое значение имеют экранные

программы (прикладная программа, имитирующая работу устройства, изображенного на экране). Функции прикладной программы может выполнять и пользователь.

Выполнение прикладных программ осуществляется операционной системой, имеющей определенный интерфейс. Чтобы сделать прикладную программу независимой от этого интерфейса, между ней и операционной системой размещают промежуточный слой программного обеспечения (комплекс программ, обеспечивающий взаимодействие пользователей и прикладных программ с различными платформами). Последний предоставляет прикладной программе стандартный интерфейс промежуточного слоя, который позволяет работать с разными операционными системами. Все более широкое значение приобретает распространение прикладных программ по глобальной сети. Среди них появились одноразовые программы. Заплатив небольшую цену за такую программу, ее можно загрузить в свою абонентскую систему и работать с ней одни либо несколько суток. После установленного срока программа перестает функционировать.

К настоящему моменту выделяют три вида программных продуктов: системное, или общее программное обеспечение (ПО); пакеты прикладных программ (ППП); инструментальной технологии программирования.

Это комплекс взаимосвязанных программ для решения задач определенного класса. Выделяются следующие виды ППП:

Проблемно-ориентированные. Используются для тех проблемных областей, в которых возможна типизация функций управления, структур данных и алгоритмов обработки. Например, это ППП автоматизации бухучета, финансовой деятельности, управления персоналом и т.д.;

Автоматизации проектирования (или САПР). Используются в работе конструкторов и технологов, связанных с разработкой чертежей, схем, диаграмм;

Общего назначения. Поддерживают компьютерные технологии конечных пользователей и включают текстовые и табличные процессоры, графические редакторы, системы управления базами данных (СУБД);

Офисные. Обеспечивают организационное управление деятельностью офиса. Включают органайзеры (записные и телефонные книжки, календари, презентации и т.д.), средства распознавания текста; Настольные издательские системы – более функционально мощные текстовые процессоры;

Системы искусственного интеллекта. Используют в работе некоторые принципы обработки информации, свойственные человеку. Включают информационные системы, поддерживающие диалог на естественном языке; экспертные системы, позволяющие давать рекомендации пользователю в различных ситуациях; интеллектуальные пакеты прикладных программ, позволяющие решать прикладные задачи без программирования.

Классификация прикладного программного обеспечения проводится по следующим направлениям.

Прикладное программное обеспечение предприятий и организаций. Например, финансовое управление, система отношений с потребителями, сеть поставок. К этому типу относится также ведомственное ПО предприятий малого бизнеса, а также ПО отдельных подразделений внутри большого предприятия. (Примеры: Управление транспортными расходами, Служба ИТ (информационных технологий) поддержки)

Программное обеспечение инфраструктуры предприятия. Обеспечивает общие возможности для поддержки ПО предприятий. Это базы данных, серверы электронной почты, управление сетью и безопасностью.

Программное обеспечение информационного работника. Обслуживает потребности индивидуальных пользователей в создании и управлении информацией. Это, как правило, управление временем, ресурсами, документацией, например, текстовые редакторы, электронные таблицы, программы-клиенты для электронной почты и блогов, персональные информационные системы и медиа редакторы.

Программное обеспечение для доступа к контенту. Используется для доступа к тем или иным программам или ресурсам без их редактирования (однако может и включать функцию редактирования). Предназначено для групп или индивидуальных пользователей цифрового контента. Это, например, медиаплееры, веб-браузеры, вспомогательные браузеры и др.

Образовательное программное обеспечение по содержанию близко к ПО для медиа и развлечений, однако в отличие от него имеет четкие требования по тестированию знаний пользователя и отслеживанию прогресса в изучении того или иного материала. Многие образовательные программы включают функции совместного пользования и многостороннего сотрудничества.

Имитационное программное обеспечение. Используется для симуляции физических или абстрактных систем в целях научных исследований, обучения или развлечения.

Инструментальные программные средства в области медиа. Обеспечивают потребности пользователей, которые производят печатные или электронные медиа ресурсы для других потребителей, на коммерческой

или образовательной основе. Это программы полиграфической обработки, верстки, обработки мультимедиа, редакторы *HTML*, редакторы цифровой анимации, цифрового звука и т. п.

Прикладные программы для проектирования и конструирования. Используются при разработке аппаратного («Железо») и программного обеспечения. Охватывают автоматизированный дизайн (*computer aided design - CAD*), автоматизированное проектирование (*computer aided engineering - CAE*), редактирование и компилирование языков программирования, программы интегрированной среды разработки (*Integrated Development Environments*), интерфейсы для прикладного программирования (*Application Programmer Interfaces*).

В данном курсе лекций мы уже обсуждали понятия алгоритма и программы, теперь немного поговорим об ещё одном компоненте программного обеспечения компьютера – утилите.

Утилита (*utility* или *tool*) – компьютерная программа, расширяющая стандартные возможности оборудования и операционных систем, выполняющая узкий круг специфических задач.

Утилиты предоставляют доступ к возможностям (параметрам, настройкам, установкам), недоступным без их применения, либо делают процесс изменения некоторых параметров проще (автоматизируют его). Утилиты зачастую входят в состав операционных систем или идут в комплекте со специализированным оборудованием.

Функции утилит:

Мониторинг показателей датчиков и производительности оборудования; мониторинг температур процессора, видеоадаптера; чтение *S.M.A.R.T.* жёстких дисков; бенчмарки.

Управление параметрами оборудования - ограничение максимальной скорости вращения CD-привода; изменение скорости вращения кулеров.

Контроль показателей - проверка ссылочной целостности; правильности записи данных.

Расширение возможностей - форматирование и/или переразметка диска с сохранением данных, удаление без возможности восстановления.

Типы утилит: дисковые утилиты; дефрагментаторы (проверка диска - поиск неправильно записанных либо повреждённых различным путём файлов и участков диска и их последующее удаление для эффективного использования дискового пространства, *CHKDSK, fsck, Scandisk*); очистка диска - удаление временных файлов, ненужных файлов, чистка «корзины» (*CCleaner*); разметка диска - деление диска на логические диски, которые могут иметь различные файловые системы и восприниматься операционной системой как несколько различных дисков (*PartionMagic, GParter, fdisk*); резервное копирование - создание резервных копий целых дисков и отдельных файлов, а также восстановление из этих копий; сжатие дисков - сжатие информации на дисках для увеличения вместимости жёстких дисков; утилиты работы с реестром (*CCleaner, Reg Organizer*), утилиты мониторинга оборудования и бенчмарки (*SpeedFan*); тесты оборудования; программные средства защиты

Бенчмарк, тест производительности (*benchmark*) - контрольная задача, необходимая для определения сравнительных характеристик производительности компьютерной системы. Иногда бенчмарками также называются программы, которые тестируют время автономной работы ноутбуков и КПК, радиус действия беспроводной сети, пропускную способность каналов передачи данных, АЧХ звукового тракта и другие доступные для измерения характеристики, напрямую не связанные с производительностью.

2. ПАКЕТЫ ПРИКЛАДНЫХ ПРОГРАММ

Пакеты общего назначения поддерживают компьютерные технологии конечных пользователей и включают текстовые и табличные процессоры (редакторы), графические редакторы, системы управления базами данных (СУБД), пакеты программ мультимедиа, пакеты демонстрационной графики.

Текстовые процессоры (редакторы) позволяют готовить текстовые документы, которые могут включать и таблицы, и рисунки, и диаграммы. Примером пакетов этого класса являются *MS Word*, Блокнот, *WordPad*. Перечень выполняемых функций, например *MS Word*, очень широк и изучается студентами в лабораторном практикуме по информатике.

Табличные процессоры (типичный пример - *MS Excel*) позволяют обрабатывать большие объёмы числовой информации (не исключая при этом обычную символьную), формируя из данных таблицы. Можно сказать, что это очень мощные калькуляторы, хранящие в своей памяти огромные числовые массивы и позволяющие выполнять над ними различные арифметические и логические операции, формировать диаграммы и делать множество других операций, полезных для решения различных задач пользователя. Аналогично пакету *MS Word*, табличный процессор *MS Excel* изучается в лабораторном практикуме по информатике.

Графические редакторы позволяют генерировать различные изобразительные объекты. Они делятся на 2 класса - растровой и векторной графики - в зависимости от того, какое внутреннее представление этих

объектов в них поддерживается. Редакторы растровой графики используются для работы с фотографиями. Они кодируют фотоизображения в цифровую форму и позволяют выполнять над ними различные редактирующие операции (выделение фрагментов, перемещение, вырезание, копирование и т.д.). Примерами редакторов этого класса являются: *Adobe Photoshop*, *Aldus Photo Styler*, *Picture Publisher*, *Photo Works Plus*. Редакторы векторной графики используются для профессиональной работы, связанной с технической и художественной иллюстрацией с последующей цветной печатью. Они занимают промежуточное место между САПР и настольными издательскими системами. Включают инструментарий для создания графического объекта; средства манипулирования объектами; средства обработки текста в части оформления и модификации параграфов, работы со шрифтами; средства вывода на печать и настройки цвета. Примерами графических редакторов этого класса являются *Corel Draw*, *Adobe Illustrator*, *Aldus Free Hand*, *Professional Draw*.

Системы управления базами данных (СУБД) используются для автоматизации процедур создания, хранения и извлечения электронных данных. Различаются способом организации данных, форматом, языком формирования запросов на операции с данными. Типичными примерами являются *MS Access*, *Oracle*, *Paradox*.

Пакеты программ мультимедиа используются для отображения (воспроизведения) и обработки аудио- и видеoinформации. Включают, в частности, пакеты *Director for Windows*, *Multimedia Viewer Kit*, *NEC MultiSpin*.

Пакеты демонстрационной графики - это конструкторы графических образов деловой информации, призванные в наглядной и динамической форме представлять результаты некоторых аналитических исследований. последовательность работы с такими пакетами включает шаги: разработка общего плана представления, выбор шаблона для оформления элементов, формирование и импорт элементов (текст, графика, таблицы, диаграммы, звуковые эффекты, видеоклипы). Примеры таких пакетов: *Power Point*, *Harvard Graphics*, *WordPerfect Presentation*

Офисные пакеты обеспечивают организационное управление деятельностью офиса. Включают, например, органайзеры (используются для автоматизации процедур планирования использования различных ресурсов (времени, денег, материалов) как отдельного человека, так и всей фирмы или её подразделений; уществуют 2 вида пакетов этого класса: 1) органайзеры для управления проектами. используются для сетевого планирования и управления проектами. Позволяют спланировать проект любой величины и сложности, эффективно распределить людские, финансовые и материальные ресурсы, составить оптимальный график работ и проконтролировать его исполнение. К ним относятся *Time Line*, *MS Project*, *CA - Super Project*; 2) органайзеры для организации деятельности отдельного человека. Это электронные секретари для эффективного управления деловыми контактами. Включают, в частности, *Lotus Organizer*, *ACTI*.

Программы для распознавания символов используются для перевода графических изображений букв и цифр в ASCII-коды этих символов. Применяются в сканерах. Примерами таких пакетов являются *Fine Reader*, *Cunie Form*, *Tiger*, *Omni Page*.

Настольные издательские системы применяются для профессиональной издательской деятельности. Позволяют осуществлять электронную верстку широкого спектра основных типов документов типа информационного бюллетеня, цветной брошюры, каталога, справочника. Позволяют решать задачи: компоновать (верстать) текст; использовать всевозможные шрифты и выполнять полиграфические изображения; осуществлять редактирование текста на уровне лучших текстовых процессоров; обрабатывать графические изображения; выводить документы полиграфического качества; работать в сетях на разных платформах. Примерами таких пакетов являются: *Corel Ventura*, *Page Maker*, *QuarkXPress*, *Frame Maker*, *MS Publisher*, *Page Plus*, *Compu Work Publisher*.

Системы автоматизации проектирования (САПР) используются в работе конструкторов и технологов, связанных с разработкой чертежей, схем, диаграмм, т.е. с обработкой графических изображений. Реализуют функции: коллективная работа в сети; экспорт - импорт файлов различных форматов; масштабирование объектов; группировка объектов, передвижение, растяжка, поворот, разрезание, изменение размеров, работа со слоями; перерисовка; управление файлами; использование чертежных инструментов, позволяющих рисовать кривые, эллипсы, линии произвольной формы, многоугольники и т.п.; работа с цветом; автоматизация отдельных процедур с использованием встроенного макроязыка. Примерами пакетов этого класса являются: *AutoCAD (AutoDesk)*, *DesignCAD*, *Grafic CAD Professional*, *DrawBase*, *Microstation*, *TurboCAD*.

Проблемно-ориентированные пакеты используются в тех предметных областях, для которых возможна типизация функций управления, структур данных и алгоритмов обработки. Типичным примером является

серия программ 1С:, позволяющая автоматизировать решение задач управления предприятием, например, 1С:Бухгалтерия, 1С: Предприятие, 1С: Кадры и т.д. К пакетам этого класса относятся и программы, реализующие дистанционное обучение, например пакет *SunRav_BookOffice* для создания и работы с электронным учебником.

Инструментарий технологии программирования - совокупность программ, обеспечивающих технологию разработки, отладки и внедрения программных продуктов. Инструментарий технологии программирования делится на два больших класса инструментальных средств: для создания отдельных приложений (программ) и для создания информационных систем и технологий.

Средства для создания отдельных приложений включают локальные средства (языки программирования, системы программирования, инструментальные среды пользователя) и интегрированные среды разработки программ, основное назначение которых - повышение производительности труда программистов за счет автоматизации создания кодов программ, обеспечивающих интерфейс пользователя графического типа, а также автоматизации разработки запросов и отчетов (например, *Delphi*).

Экспертные системы впервые появились в области медицины. Возникла идея интеграции знаний экспертов в области медицины или ее отдельных разделов в некоторую электронную форму, которая позволила бы начинающему врачу иметь своеобразного электронного советника при принятии решений по тому или иному врачебному случаю. Выбор области медицины объясняется слишком большой ценой ошибок, которые касаются жизни и здоровья людей. Постепенно от области медицины эта технология распространилась и на другие сферы деятельности человека, например, производство. Технология использования экспертных систем предполагает первоначальное «обучение» системы, т.е. заполнение её конкретными знаниями из той или иной проблемной области, а потом уже эксплуатацию наполненной знаниями экспертной системы для решения прикладных задач. Эта идеология проявила себя в проекте ЭВМ пятого поколения в части привлечения конечного пользователя к решению своих задач и связана с проблемой автоформализации знаний.

Интеллектуальные пакеты прикладных программ позволяют, аналогично экспертным системам, предварительно создавать базу знаний, включающую совокупность знаний из той или иной области деятельности человека, а затем решать практические задачи с привлечением этих знаний. Различие этих видов пакетов состоит в том, что экспертные системы, в отличие от интеллектуальных ППП, позволяют интегрировать знания из так называемых слабо формализуемых предметных областей, в которых сложно определить входные и выходные параметры задачи, а также невозможно сформировать четкий алгоритм ее решения. Кроме того, экспертные системы не формируют алгоритм решения задачи как в случае интеллектуальных ППП, а лишь выдают «советы» пользователю на основании его запроса.

Языки программирования делятся на следующие виды:

Операторные. Используются для кодирования алгоритмов, а потому также называются алгоритмическими. Имеют в составе: машинно-зависимые (ассемблер). Применяются для написания программ, явно использующих специфику конкретной аппаратуры. Каждый компьютер имеет такую систему программирования, которая изготавливается и поставляется фирмой-изготовителем вместе с компьютером; машинно-ориентированные (язык *C*). Объединяет идеи ассемблера и алгоритмического языка. Программы компактны и работают очень быстро.

Универсальные. (Турбо-Паскаль, Бэйсик). Приближены максимально, насколько это возможно, к естественному английскому языку: название каждой команды – английское слово; функциональные. Применяются, как правило, для машинного моделирования той или иной проблематики.

Проблемно-ориентированные (GPSS). Моделируют систему с помощью последовательности событий. Применяются, в частности, при проектировании вычислительных комплексов.

Объектно-ориентированные (Форт). Имеют встроенные средства для моделирования новых объектов программирования;

Логико-ориентированные (Prolog). Отдельно описываются правила предметной области, по которым затем выводятся новые факты.

Системы программирования включают:

интегрированную среду разработчика программы, состоящую, в частности, из текстового редактора, позволяющего создавать и корректировать исходные тексты программ, средств поддержки интерфейса программиста с системными средствами для выполнения различных сервисных функций (например, сохранения или открытия файла);

транслятор – программу, переводящую исходный текст во внутреннее представление компьютера;

отладчик – программу для трассировки и анализа выполнения прикладных программ. Позволяет отслеживать выполнение программы в пооператорном режиме, идентифицировать место и вид ошибок в программе, наблюдают за изменением значений переменных, выражений и т.д.;

компоновщик – программа для подготовки прикладной программы к работе в конкретных адресах основной памяти компьютера;

справочные системы.

3. ИНСТРУМЕНТАЛЬНАЯ СРЕДА ПОЛЬЗОВАТЕЛЯ

Инструментальная среда пользователя – специальные программные средства, встроенные в ППП: библиотеки функций, процедур, объектов и методов обработки; макрокоманды; программные модули-вставки; конструкторы экранных форм и отчетов; языки запросов высокого уровня.

Средства для создания информационных систем и технологий поддерживают полный цикл проектирования сложной информационной системы или технологии от исследования объекта автоматизации до оформления проектной и прочей документации на информационную систему или технологию. Они позволяют вести коллективную работу над проектом за счёт возможности работы в локальной сети, экспорта – импорта любых фрагментов проекта, организации управления проектом.

Обращает на себя внимание обилие средств, относящихся к инструментарию технологии программирования. Это связано со сложностью разработки программного продукта.

Рассмотрим теперь традиционную схему проектирования программы.

При включении компьютера в тот или иной информационный процесс для решения некоторой прикладной задачи, как следует из предыдущего изложения, он должен быть снабжен соответствующим ПО. Для разработки программы в целях решения конкретной прикладной задачи сложилась традиционная технология, показанная на **Рис.1.**

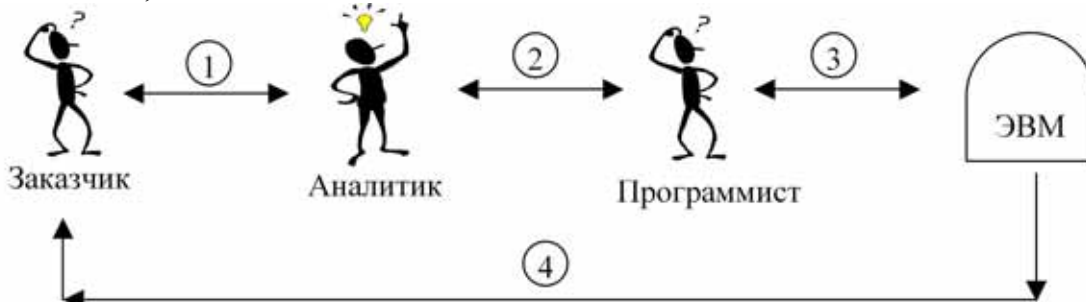


Рис. 1. Схема разработки программы.

Здесь связи означают:

1 – заказчик на естественном (например, русском) языке объясняет математику или системотехнику (аналитику), какую прикладную задачу он хочет решить. Эта задача из некоторой предметной области, например, задача бухгалтерского учета или зачисления абитуриентов в ВУЗ;

2 – математик (или системотехник - аналитик) формализует задачу, представляя её в виде математической модели или составляя строгое формальное описание процедуры её решения, входных и выходных данных. Этот этап называется формализацией задачи;

3 - программист на основании математической модели или другого формального описания поставленной задачи разрабатывает программу. Он выполняет также отладку программы, используя для этого ЭВМ. Этот этап называется программированием задачи;

4 – результаты решения задачи на ЭВМ сообщаются заказчику для определения того, удовлетворен ли он решением. Результаты решения задачи, будучи доведены до заказчика, могут его не удовлетворить в силу ряда причин. Это означает, что процесс, изображенный на рисунке, является циклическим: он завершается тогда, когда заказчик примет результаты решения задачи автоматизированным образом. В этом случае оформляется документация на эксплуатацию программного продукта. В ней описываются следующие характеристики программного продукта: основные характеристики программы, сведения об ее эксплуатации; сведения о назначении программы, области ее применения, используемых методах решения, ограничениях на применение, минимальной конфигурации технических средств; сведения для проверки работоспособности и корректности выполнения программы, для обеспечения функционирования и настройки программы на условия конкретного применения; сведения о необходимых запросах со стороны программы и форматах ответов пользователя; данные о нештатных ситуациях и поведении пользователя в них.

Этот этап называется сдачей программы в эксплуатацию.

Как видно, все связи на рисунке двунаправленны. Это означает, что в процессе проектирования программы идет диалог, в ходе которого уточняется и/или корректируется предмет общения.

Формализация задачи – непростой, для его выполнения нужно умение выделять в предметной области наиболее важные характеристики для решения задачи и выявлять способы решения задач, принятые на практике, а также требуются специальные знания и умения моделирования предметной области. Поэтому часто привлекаются математики (или системотехники - аналитики), если предметная область слишком сложна. В результате формализации создается математическая модель предметной области, которая записывается средствами различных видов математических моделей, определяются входные и выходные данные для задачи (или комплекса задач). Либо просто формируется строгое описание задачи, поскольку не всякая предметная область может быть описана средствами какой-либо математической модели (так называемые слабо формализуемые или неформализуемые области).

Пример 1. Пусть в качестве предметной области рассматривается процесс зачисления абитуриентов в ВУЗ, который хорошо знаком читателю, а потому не требует пространных комментариев. Попробуем формализовать его, определив исходные данные, требуемые для решения задачи, результаты решения, а также сам процесс решения (при этом возможны некоторые упрощения относительно реального положения дел, которые не должны смущать недавнего абитуриента).

Процесс зачисления происходит следующим образом: в приемную комиссию поступают сведения об абитуриенте, включающие данные о его оценках из аттестата о среднем образовании; по мере сдачи вступительных экзаменов оценки, если они положительны, добавляются к имеющимся сведениям об абитуриенте, либо абитуриент исключается из претендентов на зачисление, если экзамен не сдан; когда все вступительные экзамены сданы, список абитуриентов упорядочивается по мере убывания общей суммы оценок по сданным экзаменам с учетом среднего балла по аттестату; первые элементы из списка в количестве, соответствующем объему набора на первый курс, и есть студенты первого курса, фамилии которых вносятся в приказ на зачисление. Таким образом, для решения задачи требуются следующие исходные (входные) данные: фамилия, имя, отчество студента (для различия в списке), оценки из аттестата, оценки за вступительные экзамены, объем набора на первый курс; результат решения (выходные данные) – список студентов первого курса в виде списка фамилий, имен, отчеств; ход решения (последовательность требуемых действий) описан выше.

Формализация выполнена. Ее результат – описание решения в виде последовательности шагов, входные и выходные данные.

Пример 2. Пусть на первый курс по некоторой специальности зачислено N студентов. По результатам каждой сессии в течение первых четырех семестров часть студентов отчисляется за неуспеваемость в указанных среднестатистических количествах:

семестр, x	1	2	3	4
число отчисленных, y	20	5	2	1

Построить модель изменения численности студентов для определения их количества к дипломному проектированию.

Эта задача относится к классу задач экстраполяции функций: определить значение функции y в точке x ($x \notin x_0, x_n$), если известны $(n+1)$ значений функции y в точках $x_0, x_1, x_2, \dots, x_n$.

Решим задачу двумя способами.

Первый способ. Используем возможности табличного процессора EXCEL, который позволяет аппроксимировать функцию y , если известны значения функции и ее аргументов в определенном количестве точек. Для этого занесем в таблицу EXCEL исходные данные, построим диаграмму точечного типа и сформируем линию тренда, запросив вывод в диаграмму установленной табличным процессором функциональной зависимости:

Как видно из рисунка, сформированная функциональная зависимость имеет вид:

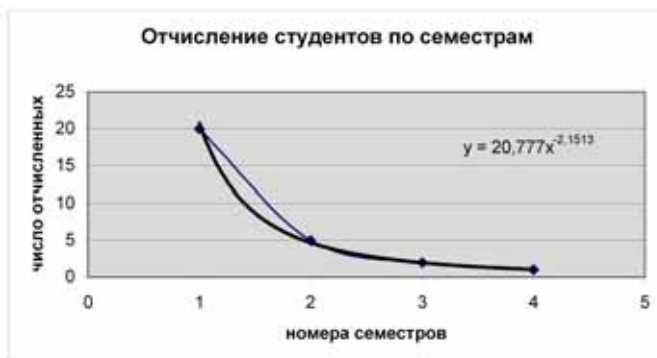
$$y(x) = 20,777x^{-2,1513},$$

где y – число отчисленных студентов; x – номер семестра.

Тогда для определения числа оставшихся студентов к пятому курсу (дипломному проектированию) может быть построена модель:

$$st = N - y(10),$$

где st – число оставшихся к дипломному проектированию студентов.



Формальная постановка задачи выполнена.

Второй способ. Используем интерполяционный многочлен Лагранжа для формализации задачи (несмотря на то, что этот многочлен решает задачу интерполяции, его можно использовать и для экстраполяции, если экстраполяционная точка не слишком отдалена от исходных точек).

В соответствии с принятыми выше обозначениями имеем:

$$st = N - \sum_{i=1}^4 p_i(10) y_i,$$

где y_i – число отчисленных студентов в i -м семестре;

$p_i(10)$ – коэффициент Лагранжа, который рассчитывается по формуле:

$$p_i(10) = \frac{(10-1) * \dots * (10-(i-1)) * (10-(i+1)) * \dots * (10-4)}{(i-1) * \dots * (i-(i-1)) * (i-(i+1)) * \dots * (i-4)}$$

и имеет значения из множества $\{p_1(10), p_2(10), p_3(10), p_4(10)\}$.

Задача решена.

Технология программирования задач различается для операторных и функциональных языков программирования. Ограничимся рассмотрением вопроса для операторных языков. Тогда этот этап разбивается на два последовательных шага – разработка алгоритма и отладку программы.

Алгоритм – точное предписание, определяющее вычислительный процесс, ведущий от исходных данных к искомому результату.

Алгоритм обладает следующими свойствами (они следуют из определения): определенность (детерминированность) – каждая команда (или предписание) понятна исполнителю (человеку или компьютеру) и исключает неоднозначность исполнения; результативность – реализация вычислительного процесса, предусмотренного алгоритмом, должна через определенное число шагов привести к результату или сообщению о невозможности его получения; массовость – если алгоритм разработан для решения определенной задачи, он должен быть применим для решения задач этого типа при всех допустимых значениях исходных данных; дискретность – пошаговый характер процесса получения результата, состоящий в последовательном выполнении конечного числа заданных алгоритмом действий.

Различают следующие простейшие виды алгоритмов: *линейный*, когда предписания алгоритма выполняются в той последовательности, в которой они представлены в алгоритме; *разветвляющийся*, когда ход исполнения предписаний может меняться относительно их нахождения в алгоритме в зависимости от значений исходных или промежуточных данных; *циклический*, когда предписания алгоритма выполняются многократно. В зависимости от характера повторений различают циклические алгоритмы с заданным и заданным числом повторений (в этом случае такие алгоритмы называют итерационными).

Существуют следующие способы описания алгоритма: словесный, структурно-стилизированный, язык графических символов, операторный язык. Не будем на этом подробно останавливаться.

Отладка программы - самый трудоемкий этап. Его цель – проверка синтаксической и логической правильности программы, а также определение того, что программа функционирует на всем диапазоне допустимых данных. В процессе отладки программы выделяются этапы: трансляция исходного текста программы; компоновка программы; выполнение программы с целью определения логических ошибок; тестирование программы. Рассмотрим их подробнее.

Трансляция. При трансляции выполняется перевод программы, понятной человеку, на язык, понятный компьютеру. Если цель трансляции – преобразование всего исходного текста на внутренний язык компьютера (т.е. получение некоторого нового кода) и только, то такая трансляция называется также компиляцией. Исходный текст называется также исходной программой или исходным модулем, а результат компиляции – объектным кодом или объектным модулем. Если же трансляции подвергаются отдельные операторы исходных текстов и при этом полученные коды сразу выполняются, такая трансляция называется интерпретацией. Поскольку трансляция выполняется специальными программными средствами, последние носят название компилятора или интерпретатора, соответственно.

В процессе компиляции последовательно выполняются лексический, синтаксический, семантический анализ, генерация промежуточного кода, генерация внутреннего представления.

При лексическом анализе выявляются отдельные составляющие текста программы, которые называются лексемами, и определяется их тип. К числу лексем относятся названия операторов, например, *read* или *write*; имена (переменных или констант), например, *NABOR* или *CHISLO*, различные разделители, такие как круглые скобки, знаки препинания и т.д. Одновременно типами указанных лексем являются названия операторов, имена переменных, разделители и т.д. Если программистом допущена ошибка и оператор ввода указан, например, *rread*, он будет распознан компилятором как имя. На этом этапе выявляется также использование недопустимых языком программирования символов, например, символа *@*. В результате лексического анализа исходная программа кодируется: каждая лексема заменяется кодом ее типа, что сокращает объем текста программы. Кроме того, из текста программы удаляются пробелы.

Пример 3. Лексический анализ фрагмента программы на Турбо-Паскале, содержащего только один правильно записанный оператор ввода:

`read (NABOR,CHISLO);`

дает цепочку

1 2 3 4 3 2 4,

где 1 – код названия оператора; 2 – код скобки; 3 – код имени; 4 – код знака препинания.

Пример 4. Лексический анализ фрагмента программы на Турбо-Паскале, содержащего только один неправильно записанный оператор ввода:

`rread (NABOR,CHISLO);`

даёт цепочку

3 2 3 4 3 2 4.

Синтаксический анализ. Определяется синтаксическая правильность закодированной в результате лексического анализа цепочки лексем. Например, определяется парность скобок, наличие в нужном месте требуемого знака препинания, соответствие исходного текста структурным правилам того или иного языка программирования, средствами которого была составлена программа. Для решения задачи в исходной (закодированной) цепочке выделяются подструктуры, соответствующие некоторым фрагментам оператора. Например, для оператора ввода в Турбо-Паскале можно определить структуру, представленную на **Рис. 2**:



Рис. 2. Структура оператора ввода в Турбо-Паскале.

В то же время структура «правильного» оператора ввода определяется схемой:



Рис. 3. Структура правильного оператора.

Поскольку представленные структуры идентичны, оператор `read (NABOR,CHISLO);` при синтаксическом анализе определяется как правильный. Именно на этом шаге выявляются ошибки в написании названий операторов: они ведут к некорректной структуре всего оператора. Так, примеру с "неправильным" оператором ввода соответствует структура, показанная на **Рис. 4**:



Рис. 4. Структура оператора ввода в Турбо-Паскале.

Поскольку структура этого рисунка не соответствует структуре **Рис. 2**, оператор `rread (NABOR,CHISLO);` расценивается как синтаксически некорректный: компилятор не найдет в анализируемом фрагменте требуемого действия, которое задается названием оператора, и сообщит об этом программисту.

Семантический анализ. На этом шаге выявляются ошибки, допущенные программистом в нарушение правил составления программ, например, следующего вида: все переменные и константы перед употреблением в операторах языка должны быть описаны; каждое имя (переменной или константы) должно быть описано только один раз; требуется согласование типов переменных с использующими их функциями и т.д. Так, если вся программа состоит только из оператора

```
read (NABOR,CHISLO);
```

она будет определена как семантически некорректная, поскольку в ней отсутствуют описания вводимых переменных.

Пример 5. Программа

```
var NABOR, CHISLO: integer;
```

```
read (NABOR, CHISLO);
```

расценивается семантическим анализатором как семантически корректная.

Пример 6. Программа

```
var CHISLO: integer;
```

```
CHISLO:=CHISLO + 1;
```

расценивается семантическим анализатором как семантически корректная.

Генерация промежуточного кода. Закодированная цепочка лексем преобразуется в некоторое промежуточное представление, принятое на том или ином компьютере, например, в программу на языке ассемблера (для простоты используем некоторый условный ассемблер).

Пример 7. Программа из примера 4 преобразуется в промежуточный код на условном ассемблере:

```
$1 DD ? операторы DD описывают переменные: каждый из них
```

```
$2 DD ?; отводит под переменную 2 байта памяти;
```

```
$3 DD ?; переменные $1 - $3 вводятся генератором кода как
```

```
CHISLO DD ?; вспомогательные переменные;
```

```
MOVE 1, $1; в операторах MOVE константы или содержимое
```

```
MOVE CHISLO, $2; переменных слева от запятой помещаются в переменные и
```

```
MOVE $1, AX; регистры процессора (AX и CX), указанные справа;
```

```
MOVE $2, CX;
```

```
ADD AX, CX; выполняется сложение содержимого регистров AX и CX,
```

```
; результат остаётся в регистре AX
```

```
MOVE AX, $3; выполняются перемещения значений, находящихся в
```

```
MOVE $3, CHISLO; регистре и переменной, в соответствующие переменные
```

Полученный код избыточен: в самом деле, выполняется бессмысленная пересылка константы 1 и значения переменной CHISLO сначала в дополнительные переменные \$1 и \$2, а затем уже в регистры AX и CX, которые и используются при сложении. Аналогичные бессмысленные действия выполняются и с результатом сложения. Однако уменьшение размеров кода осуществляется на следующем этапе.

Оптимизация промежуточного кода. Из программы, полученной на предыдущем шаге, устраняются «лишние» операторы, переменные и константы, использование которых не влияет на корректность выполняемых действий.

Пример 8. Оптимизация промежуточного кода для программы из примера 5 приводит её к виду:

```
CHISLO DD ?
```

```
MOVE 1, AX
```

```
MOVE CHISLO, CX
```

```
ADD AX, CX
```

```
MOVE AX, CHISLO
```

Такой результат связан с тем, что ряд операторов выполняли лишние перемещения данных из одного места хранения в другое. Удаление этих операторов повлекло удаление вспомогательных переменных.

Генерация объектного кода. Программа, полученная в результате оптимизации, преобразуется в машинный код (так называемый объектный модуль), в котором использованы относительные, а не абсолютные адреса основной памяти. Как правило, для получения объектного модуля применяется условная память с начальным адресом 1. Каждый оператор программы преобразуется в машинную команду и размещается, начиная с начального адреса, в той последовательности, в которой он следует в программе. При этом учитывается размер каждого оператора.

Пусть командам программы из примера 6 соответствуют машинные команды:

код	объем	действие операции
124	1 б	сложить содержимое регистров AX и CX, результат – в регистре AX,
125	3 б	поместить содержимое регистра AX по адресу,
126	2 б	поместить константу в регистр AX,

127 3 б поместить содержимое адреса в регистр CX.

Оператор DD выделяет объем памяти в 2 байта для описываемой переменной. Тогда схема размещения объектного кода имеет вид, представленный в таблице:

Адрес (байт)	Команда		Комментарий
	внутреннее представление	объем (байт)	
1	-	2	Память для переменной CHISLO – результат действия команды DD
3	126 1	2	В регистр AX помещается 1
5	127 1	3	Содержимое адреса 1 помещается в регистр CX
8	124	1	Соответствует действию CHISLO + 1
9	125 1	3	Содержимое регистра AX помещается по адресу 1, т.е. выполняется действие CHISLO = CHISLO+1

Этот код еще не готов к выполнению: для этого требуется его «ориентация» на те адреса основной памяти, где он будет выполняться.

При интерпретации анализу подвергается не вся программа, а отдельные операторы. При этом из названных шагов выполняются лишь первые четыре.

Компоновка. Компоновка программы создает готовую для работы программу, которая называется также исполняемой программой или загрузочным модулем. При этом решаются две основные задачи: если в программе используются функции, например, *sin*, *exp* и т.д., соответствующие им программные модули выбираются из библиотеки подпрограмм соответствующей системы программирования и вставляются в объектный модуль; объектный модуль преобразуется в соответствии с реальными адресами основной памяти, куда будет размещаться программа для выполнения.

Например, пусть для выполнения программы из таблицы раздела Трансляция отводится область основной памяти, начиная с абсолютного адреса 40. Тогда, с учетом сегментированной схемы адресации указанные в таблице адреса (с учетом абсолютного адреса 40) преобразуются в свои сегментированные эквиваленты:

Исходные адреса	Сегментированные адреса
1	0002:0008
3	0002:000A
5	0002:000C
8	0002:000F
9	0003:0000

В полученных сегментированных адресах номера сегментов хранятся в специальных регистрах процессора и в коды команд не включаются, а включаются лишь смещения. Тогда объектный код из таблицы преобразуется в следующий вид:

Адрес размещения команды		Содержимое адреса
Абсолютный (десятичная система счисления)	Сегментированный (шестнадцатеричная система счисления)	
40	0002:0008	-
42	0002:000A	126 1
44	0002:000C	127 0008
47	0002:000F	124
48	0003:0000	125 0008

Выполнение и тестирование программы. Выполнение программы с целью определения логических ошибок осуществляется после успешной компоновки или по ходу интерпретации каждого оператора. При этом выявляются такие ошибки, как, например, деление на ноль или вычисление логарифма отрицательного числа. Здесь же апробируется корректность описания используемых переменных, а также законченность циклов.

Пример 9. Для программы
`var CHISLO: integer;`

```
CHISLO:=CHISLO + 1;
```

и (ее кода) будут выявлены две логические ошибки: неизвестность результата ввиду отсутствия в программе оператора вывода, например, *write (CHISLO)*; после добавления оператора вывода в программу и выполнения последней, полученное потребителем значение переменной *CHISLO* имеет произвольный характер, так как не определено начальное значение этой переменной.

Тогда правильная программа вместо той, которая приведена в примере 1, могла бы иметь вид:

```
var CHISLO: integer;  
input (CHISLO);  
CHISLO:=CHISLO+1;  
write (CHISLO);.
```

Тестирование программы имеет целью определение работоспособности программы на всем требуемом диапазоне исходных данных. Программистом составляется представительная выборка исходной информации, которая позволит определить корректность программы при любых входных параметрах. Для программы из примера 1 задаются набором переменной *CHISLO*. Так, если среди этих значений будет такое, которое превышает допустимый диапазон типа *integer*, при выполнении программы в процессе тестирования возникает переполнение разрядной сетки (*overflow*), что обсуждалось ранее. Это потребует от программиста переделки исходной программы в части описания переменной *CHISLO*.

4. ПРИМЕРЫ ПРИКЛАДНЫХ КОМПЬЮТЕРНЫХ СИСТЕМ

4.1 Управление базами данных

Система управления базами данных (СУБД) - специализированная программа (чаще комплекс программ), предназначенная для организации и ведения базы данных.

Базой данных является представленная в объективной форме совокупность самостоятельных материалов (статей, расчетов, нормативных актов, судебных решений и иных подобных материалов), систематизированных таким образом, чтобы эти материалы могли быть найдены и обработаны с помощью электронной вычислительной машины (Гражданский кодекс РФ, ст. 1260).

Существует множество других определений понятия «база данных», так или иначе сводящихся к понятию «совокупность хранимых данных». Однако большинство из этих определений не позволяет отличить базу данных от объектов, которые базой данных заведомо не являются, например, от архивов документов, карточек, библиотек и т.п. Таким образом, база данных есть не просто совокупность хранимых данных (записей, документов, фактов и т.п.), но такая совокупность, которая обладает, по меньшей мере, тремя важными свойствами (признаками): База данных хранится и обрабатывается в вычислительной системе. Таким образом, любые внекомпьютерные хранилища информации (архивы, библиотеки и т. п.) базами данных не являются. Данные в базе данных хорошо структурированы (систематизированы). Под структурированностью в данном случае понимается явное выделение составных частей (элементов), связей между ними, а также типизация элементов и связей, при которой с каждым типом элемента или связи соотносится определённая семантика и допустимые операции. Структура базы данных обеспечивает эффективный поиск и обработку данных. Эффективность здесь главным образом определяется тем, как соотносятся гибкость и мощность возможностей (поиска и обработки) с затратами усилий и ресурсов. Из трёх перечисленных признаков только первый является строгим, а два других допускают различные трактовки и различные степени оценки. Не существует возможности строго формально определить, является ли некоторая совокупность данных на компьютере базой данных или нет. Можно лишь установить некоторую степень соответствия требованиям к БД. В такой ситуации не последнюю роль играет общепринятая практика. В соответствии с ней, например, не называют базами данных файловые архивы или электронные таблицы, несмотря на то, что они в некоторой степени обладают признаками БД. Принято считать, что эта степень в большинстве случаев недостаточна (хотя могут быть исключения). Базой данных часто ошибочно называют систему управления базами данных. Необходимо различать хранимые данные (собственно БД) и программное обеспечение, предназначенное для организации и ведения базы данных (СУБД).

Для создания и управления информационной системой СУБД необходима в той же степени, как для разработки программы на алгоритмическом языке необходим транслятор. Основные функции СУБД: управление данными во внешней памяти (на дисках); управление данными в оперативной памяти с использованием дискового кэша; журнализация изменений, резервное копирование и восстановление базы данных после сбоев; поддержка языков БД (язык определения данных, язык манипулирования данными). Обычно современная СУБД содержит следующие компоненты: ядро, которое отвечает за управление данными во внешней и оперативной памяти и журнализацию, процессор языка базы данных,

обеспечивающий оптимизацию запросов на извлечение и изменение данных и создание, как правило, машинно-независимого исполняемого внутреннего кода, подсистему поддержки времени исполнения, которая интерпретирует программы манипуляции данными, создающие пользовательский интерфейс с СУБД, а также сервисные программы (внешние утилиты), обеспечивающие ряд дополнительных возможностей по обслуживанию информационной системы.

По типу управляемой базы данных СУБД разделяются на: иерархические, сетевые, реляционные, объектно-реляционные и объектно-ориентированные. По архитектуре организации хранения данных: локальные СУБД (все части локальной СУБД размещаются на одном компьютере); распределенные СУБД (части СУБД могут размещаться на двух и более компьютерах).

По способу доступа к БД

Файл-серверные. В файл-серверных СУБД файлы данных располагаются централизованно на файл-сервере. Ядро СУБД располагается на каждом клиентском компьютере. Доступ к данным осуществляется через локальную сеть. Синхронизация чтений и обновлений осуществляется посредством файловых блокировок. Преимуществом этой архитектуры является низкая нагрузка на ЦП сервера, а недостатком - высокая нагрузка локальной сети. На данный момент файл-серверные СУБД считаются устаревшими. Примеры: Microsoft Access, Paradox, dBase.

Клиент-серверные. Такие СУБД состоят из клиентской части (которая входит в состав прикладной программы) и сервера. Клиент-серверные СУБД, в отличие от файл-серверных, обеспечивают разграничение доступа между пользователями и мало загружают сеть и клиентские машины. Сервер является внешней по отношению к клиенту программой, и по надобности его можно заменить другим. Недостаток клиент-серверных СУБД в самом факте существования сервера (что плохо для локальных программ - в них удобнее встраиваемые СУБД) и больших вычислительных ресурсов, потребляемых сервером. Примеры: Oracle, Firebird, Interbase, IBM, MS SQL Server, Sybase, PostgreSQL, MySQL, ЛИНТЕР, MDBS.

Встраиваемые. Встраиваемая СУБД - библиотека, которая позволяет унифицированным образом хранить большие объемы данных на локальной машине. Доступ к данным может происходить через SQL либо через особые функции СУБД. Встраиваемые СУБД быстрее обычных клиент-серверных и не требуют установки сервера, поэтому востребованы в локальном ПО, которое имеет дело с большими объемами данных (например, геоинформационные системы). Примеры: OpenEdge, SQLite, BerkeleyDB, один из вариантов Firebird, один из вариантов MySQL, Sav Zigzag, Microsoft AQL Server Compact, ЛИНТЕР.

4.2 Офисное приложение

Текстовый редактор – компьютерная программа, предназначенная для создания и изменения текстовых файлов, а также их просмотра на экране, вывода на печать, поиска фрагментов текста и т. п.

Условно выделяют два типа редакторов, а также отдельно рабочие среды, содержащие редактор. Первый тип ориентирован на работу с последовательностью символов в текстовых файлах. Такие редакторы обеспечивают расширенную функциональность – подсветку синтаксиса, сортировку строк, шаблоны, конвертацию кодировок, показ кодов символов и т. п. Иногда их называют редакторы кода, так как основное их предназначение - написание исходных кодов компьютерных программ.

Второй тип текстовых редакторов имеет расширенные функции форматирования текста, внедрения в него графики и формул, таблиц и объектов. Такие редакторы часто называют текстовыми процессорами и предназначены они для создания различного рода документов, от личных писем до официальных бумаг. Классический пример – Microsoft Word.

Также выделяют более общий класс программ – текстовые рабочие среды. По сути, такие среды представляют собой полноценную рабочую среду, в которой можно решать самые разнообразные задачи: с помощью надстроек они позволяют писать и читать письма, веб-каналы, работать в вики и Вебе, вести дневник, управлять списками адресов и задач. Представители этого класса - Emacs, Archy, Vim и Acme из операционной системы Plan 9. Такие программы могут служить средами разработки программного обеспечения. В любом случае, последние всегда содержат текстовый редактор как необходимый инструмент программирования.

Список текстовых редакторов приведен в Табл. 1. В таблицу включаются проекты согласно следующим критериям: Редактор достаточно известен, по крайней мере представлен в значительном количестве каталогов ПО; Разработка проекта продолжается; Интерфейс редактора — оконный, а не текстовый, как nano. Текстовые неинтересны широкому кругу пользователей по причине непривычности интерфейса, хотя при желании можно сделать по ним отдельную таблицу. Распространенность ОС:

Операционная система - *Windows*, либо кроссплатформенная и имеется порт под *Windows*. Чисто линуксовые можно выделить в отдельную таблицу, объединив её с предыдущим пунктом.

Табл. 1. Текстовые редакторы.

Текстовый редактор	Ссылка	Цена	Лицензия	Открытость кода	ОС помимо Windows	Возможности				
						Работа без установки	Открытие нескольких документов в одном окне	HEX-редактор	Нумерация строк	Смена конца строки (CR LF)
Стандартный блокнот Windows	стандартно входит в состав системы	б/п	часть ОС	код закрыт	нет	есть в ОС	нет	нет	нет	нет
Notepad++	notepad-plus.sourceforge.net	б/п	GNU GPL	код открыт	Linux		да, вкладки либо панели	нет	Да	
EditPlus	editplus.com	35\$	Shareware, проприет.	код закрыт	нет					
EmEditor	emeditor.com	40\$	Shareware, проприет.	код закрыт	нет	да	да	нет		возможность выбора окне сохранения
UltraEdit	ultraedit.com	50\$	Shareware, проприет.	код закрыт	нет		да	Да		
Crimson Editor[см 2]	crimsoneditor.com	б/п	GNU GPL	код открыт	нет		да			
TextPad	textpad.com	32\$	Shareware, проприет.	код закрыт	нет		да	Да		
RJ TextEd	rj-texted.se	б/п	Freeware	код закрыт	нет		да	Да		
AkelPad	akelpad.sourceforge.net	б/п	BSD	код открыт	нет	да	да	нет	да (плагин)	да
Boxer Editor	Text boxersoftware.com		проприет.	код закрыт	нет		да	да		

Перечислим теперь текстовые редакторы

Emacs - Один из самых мощных по возможностям многоцелевой, свободный редактор. История развития *Emacs* превышает 35 лет. Ядро реализовано на Си, остальная часть на диалекте Лисп - *Elisp*. Это позволяет модифицировать поведение редактора без его перекомпиляции. Имеет большое число режимов работы, при использовании редактора для программирования - не уступает возможностям *IDE*.

jEdit - Свободный редактор на *Java*.

Kate - Мощный расширяемый свободный текстовый редактор с подсветкой синтаксиса для массы языков программирования и разметки (модули подсветки можно автоматически обновлять по сети). Гибкий настраиваемый интерфейс. Входит в состав *KDE*.

Notepad - входит в состав *Microsoft Windows*.

Vim. Свободный. Один из самых мощных по возможностям редактор для программистов. Модальный, основные режимы работы: режим ввода текста и командный режим. Обладает широкими возможностями настройки и автоматизации. Целевая аудитория - администраторы и программисты. Возможно расширение функционала за счёт поддержки скриптовых языков.

GNU nano - Свободный редактор для текстового режима.

EditPlus - текстовый редактор для *Windows*, предназначенный для программирования и веб-разработки.

EmEditor - платный редактор для *Windows*-систем. Обеспечивает подсветку текста для разных форматов, модулей, однако интерфейс требует изучения.

SciTE - свободный кроссплатформенный редактор с подсветкой синтаксиса для многих языков программирования, фолдингом. Широкие возможности настройки и автоматизации.

NEdit - популярный текстовый редактор и редактор исходного кода для среды *X Window System*.

Notepad++ (GNU GPL), основан на том же движке, что и *SciTE*. Имеет сходные возможности и удобный интерфейс.

Notepad gnu - текстовый редактор с открытым исходным кодом базирующийся на компоненте *SynEdit*, представляющий альтернативу стандартному Блокноту *Windows*, данную программу могут использовать как программисты (редактирование кода), так и все те, кому необходим хороший текстовый редактор. Редактор имеет подсветку синтаксиса большого количества языков программирования (55 штук) и распространяется под лицензией *GNU GPL (General Public License)*.

PSPad - Текстовый редактор с подсветкой синтаксиса, поддержкой скриптов и инструментами для работы с *HTML*-кодом.

TEA - Редактор с сотнями функций обработки текста и разметки в *HTML, LaTeX, Docbook*.

Texter - текстовый редактор для *Windows*

metapad - текстовый редактор для *Windows*

Crimson Editor - текстовый редактор с подсветкой синтаксиса для *Microsoft Windows*, распространяемый под лицензией *GNU GPL*

AkelPad - текстовый редактор с открытым исходным кодом, созданный, чтобы быть маленьким и быстрым.

Другие редакторы: *UltraEdit, RulNote, EditPad, HippoEdit, MiBEditor, TextMate*

TextEdit - текстовый редактор-процессор от *Apple Inc.*

VEdit - один из самых быстрых универсальных файловых редакторов доступных на рынке для программистов. Написан на ассемблере.

Популярные текстовые процессоры - второй тип текстовых редакторов - имеет расширенные функции формирования текста, внедрения в него графики и формул, таблиц и объектов. Такие редакторы часто называют текстовыми процессорами и предназначены они для создания различного рода документов, от личных писем до официальных бумаг. Классический пример - *Microsoft Word*. Кроме того: *AbiWord, Lotus WordPro, Microsoft Works, WordPad* - входит в дистрибутив *MS Windows, OpenOffice.org Writer, PolyEdit, WordPerfect*

Текстовый процессор - вид прикладной компьютерной программы, предназначенной для производства (включая набор, редактирование, форматирование, иногда печать) любого вида печатной информации. Иногда текстовый процессор называют текстовым редактором второго рода. Текстовыми процессорами в 1970-е - 1980-е годы называли предназначенные для набора и печати текстов машины индивидуального и офисного использования, состоящие из клавиатуры, встроенного компьютера для простейшего редактирования текста, а также электрического печатного устройства. Позднее наименование «текстовый процессор» стало использоваться для компьютерных программ, предназначенных для аналогичного использования.

Текстовые процессоры, в отличие от текстовых редакторов, имеют больше возможностей для форматирования текста, внедрения в него графики, формул, таблиц и других объектов. Поэтому они могут быть использованы не только для набора текстов, но и для создания различного рода документов, в том числе официальных. Классическим примером текстового процессора является *Microsoft Word*.

Программы для работы с текстами можно разделить на простые текстовые процессоры, мощные текстовые процессоры и издательские системы.

Известные текстовые процессоры: *AbiWord, Adobe InCopy, JWPce* - текстовый процессор для японского языка, *LaTeX* - наиболее популярный набор макрорасширений (или макропакет) системы компьютерной вёрстки *TeX, Lotus WordPro, Microsoft Word, Microsoft Works, OpenOffice.org Writer, PolyEdit, WordPad* - входит в дистрибутив *MS Windows, WordPerfect*

4.3 Microsoft Office

Microsoft Office - пакет приложений, созданных корпорацией *Microsoft* для операционных систем *Microsoft Windows* и *Apple Mac OS X*. В состав этого пакета входит программное обеспечение для работы с различными типами документов: текстами, электронными таблицами, базами данных и др. *Microsoft Office* является сервером *OLE* объектов и его функции могут использоваться другими приложениями, а так же самими приложениями *Microsoft Office*. Поддерживает скрипты и макросы, написанные на *VBA*.

Microsoft Office поставляется в нескольких редакциях. Отличия редакций в составе пакета и цене. Наиболее полная из них содержит:

Microsoft Office Word – текстовый процессор. Доступен под *Windows* и *Apple Mac OS X*. Позволяет подготавливать документы различной сложности. Поддерживает *OLE*, подключаемые модули сторонних разработчиков, шаблоны и многое другое. Основным форматом в последней версии является позиционируемый как открытый *Microsoft Office Open XML*, который представляет собой *ZIP*-архив, содержащий текст в виде *XML*, а также всю необходимую графику. Наиболее распространенным остается двоичный формат файлов *Microsoft Word 97-2000* с расширением *.doc*. Продукт занимает ведущее положение на рынке текстовых процессоров, и его форматы используются как стандарт де-факто в

документообороте большинства предприятий. *Word* также доступен в некоторых редакциях *Microsoft Works*. Главные конкуренты - *OpenOffice.org Writer*, *StarOffice Writer*, *Corel WordPerfect* и *Apple Pages* (только на платформе *Mac OS*), а также *AbiWord* (в тех случаях, когда его возможностей достаточно, а малый объем и скорость работы при невысоких требованиях к ресурсам более важны).

Microsoft Office Excel – табличный процессор. Поддерживает все необходимые функции для создания электронных таблиц любой сложности. Занимает ведущее положение на рынке. Последняя версия использует формат *OOXML* с расширением «*.xlsx*», более ранние версии использовали двоичный формат с расширением «*.xls*». Доступен под *Windows* и *Apple Mac OS X*. Главные конкуренты - *OpenOffice.org Calc*, *StarOffice*, *Gnumeric*, *Corel Quattro Pro* и *Apple Numbers* (только на платформе *Mac OS*).

Microsoft Office Outlook (не путать с *Outlook Express*) - персональный коммуникатор. В состав *Outlook* входят: календарь, планировщик задач, записки, менеджер электронной почты, адресная книга. Поддерживается совместная сетевая работа. Главные конкуренты почтового клиента - *Mozilla Thunderbird/SeaMonkey*, *Eudora Mail*, *The Bat!*. Главные конкуренты диспетчера персональных данных - *Mozilla*, *Lotus Organizer* и *Novell Evolution*. Доступен под *Windows*. Эквивалент для *Apple Mac OS X* - *Microsoft Entourage*.

Microsoft Office PowerPoint - приложение для подготовки презентаций под *Microsoft Windows* и *Apple Mac OS X*. Главные конкуренты - *OpenOffice.org Impress*, *Corel WordPerfect* и *Apple Keynote*.

Microsoft Office Access - приложение для управления базами данных.

Microsoft Office InfoPath - приложение сбора данных и управления ими - упрощает процесс сбора сведений.

Microsoft Office Communicator - предназначен для организации всестороннего общения между людьми.

Microsoft Office Communicator 2007 обеспечивает возможность общения посредством простого обмена мгновенными сообщениями, а также проведения голосовой и видеобеседы. Данное приложение является частью программного пакета *Microsoft Office* и тесно с ним интегрировано, что позволяет ему работать совместно с любой программой семейства *Microsoft Office*.

Microsoft Office Publisher - приложение для подготовки публикаций.

Microsoft Office Visio - приложение для работы с бизнес-диаграммами и техническими диаграммами - позволяет преобразовывать концепции и обычные бизнес-данные в диаграммы.

Microsoft Office Project - управление проектами.

Microsoft Query - просмотр и отбор информации из баз данных.

Microsoft Office OneNote - приложение для записи заметок и управления ими.

Microsoft Office Groove 2007 - приложение для поддержки совместной работы.

Microsoft Office SharePoint Designer - инструмент для построения приложений на платформе *Microsoft SharePoint* и адаптации узлов *SharePoint*.

Microsoft Office Picture Manager - работа с рисунками.

Microsoft Office Diagnostics - диагностика и восстановление поврежденных приложений *Microsoft Office*.

Ранее в *Microsoft Office* входило приложение *Microsoft FrontPage*, однако *Microsoft* приняла решение исключить это приложение из *Office* и прекратить его разработку. В *Microsoft Office 2007* программа *FrontPage* была заменена на *Microsoft SharePoint Designer*.

Перечислим теперь версии продукта.

Версии выпущенные до *Microsoft Office 97* (включая *Outlook 97*) более не поддерживаются.

Office 97 (включая *Outlook 98*) расширенная поддержка закончилась 16.01.2004. Выпуск исправлений в рамках основной поддержки прекращен 31.08.2001.

Office 2000 - Базовая поддержка продукта прекращена 30.07.2004. Расширенная поддержка закончилась 14.07.2009.

Office XP - Базовая поддержка продукта прекращена 11.07.2006. Расширенная поддержка доступна до 12.07.2011.

Office 2003 - Базовая поддержка продукта прекращена 13.01.2009. Расширенная поддержка будет доступна до 14.01.2014.

Для текущей (2007) и будущих версий *MS Office* - Продолжительность базовой поддержки 5 лет после выпуска продукта или 2 года после выпуска следующей версии продукта (что наступит позднее), расширенная поддержка будет доступна в течение 5 лет после окончания срока основной поддержки.

MS Office включает программы - помощники: Программа *Graph*, позволяет ввести несколько чисел и быстро превратить их в график. Программа *Organization Chart* поможет создать штатное расписание фирмы. Программа *Equation Editor* предназначена для создания математических формул в редакторе *Word*. Программа *WordArt* поможет изменить почти до неузнаваемости форму букв или цифр, чтобы использовать

их при создании привлекательных логотипов или заголовков. Программа *ClipArt Gallery* может просмотреть несколько сотен рисунков в поисках наилучшей иллюстрации для информационного бюллетеня или презентации.

Система оперативной помощи: Пакет *Office* и его приложения изобилуют полезными подсказками и пошаговыми указаниями того, как выполнять ту или иную операцию. Талантливые Мастера (*Wizards*) проведут шаг за шагом к решению сложных задач. Карточки - шпаргалки (*Cue Cards*) предложат подробные указания, которые постоянно будут перед глазами во время работы. Функция Просмотр (*Preview*) и Примеры (*Demos*) продемонстрируют выполнение сложных задач.

Перечислим дополнительные возможности *Office XP*.

Смарт-теги - набор кнопок, работающих в различных приложениях, которые появляются, когда в них возникает необходимость (например, при вставке данных или возникновении ошибки в формуле *Microsoft Excel*), и позволяют выбрать наиболее быстрый способ выполнения задачи.

Области задач. С помощью областей задач осуществляется централизованный доступ к средствам выполнения основных операций. Поиск, открытие документов, просмотр буфера обмена, форматирование документов и презентаций, загрузка шаблонов из Веба и многое другое

Смарт-теги параметров вставки позволяют изменить форматирование копируемых данных перед их вставкой в новый документ. Можно даже назначать определенные свойства вставляемым данным в зависимости от того, что в них содержится.

Смарт-теги параметров автозамены позволяют управлять функциями автозамены в приложениях *Office*. Можно отменить или отключить автозамену, а также вызвать диалоговое окно «Параметры автозамены» без использования меню «Сервис».

Создание файла (область задач) Теперь можно загрузить новый документ, работая в другом документе, создать документ на основе шаблона, хранящегося в Вебе и посмотреть список недавно использованных шаблонов и документов.

Буфер обмена *Office* (область задач) В буфере обмена теперь 24 ячейки для хранения информации – это вдвое больше, чем было предусмотрено в пакете *Office 2000*. Пользуясь эскизами в области задач стало проще находить нужную ячейку.

Office Document Imaging позволяет осуществить сканирование документов для последующего просмотра и повторного использования текста в приложениях *Office*. Возможности поиска текста, имеющиеся в пакете *Office*, теперь используются для отсканированных документов

Распознавание речи Повышение производительности труда за счёт диктовки текста и подачи голосом команд для изменения форматирования и перехода по меню.

Рукописный ввод Заметки на карманном устройстве, могут быть загружены в приложения *Office* в виде текста. Можно также осуществлять рукописный ввод непосредственно в приложении *Microsoft Word* (используя, кроме английского языка, японский, китайский и корейский языки) и сохранять работу в виде рукописного документа или печатного текста.

Сжатие рисунков Чтобы сжать изображение, достаточно просто указать, как оно будет использоваться (например, для Веба). С помощью сжатия *Office XP* уменьшает размер файла без ущерба для качества изображения.

Предисловие к электронным сообщениям *Office* К сообщениям, отправляемым из приложений *Office* по электронной почте, добавляется краткое предисловие

Задание вопросов Справку в *Office XP* можно получить, введя вопрос в поле, расположенное в меню *Office*, без загрузки мастера ответов или помощника

Помощник В *Office XP* помощник по умолчанию скрыт и появляется только при обращении к справке

Усовершенствования справки *Window* Упрощен доступ к справке и полезным веб-узлам, включая узел технической поддержки и узел *Office Update*. Каталог «Что нового» позволяет узнать о новых возможностях.

Доступ к информации:

Контекстно-чувствительные смарт-теги. С помощью смарт-тегов *Office XP* автоматически распознает данные различного типа, включая имена, даты, адреса, телефоны, географические названия и обозначения акций (распознавание данных любого вида)

Область задач поиска. Эта область задач упрощает поиск текста в документе и определение местоположения файлов и папок во время работы над документом. Можно также для ускорения поиска проиндексировать файлы на своем компьютере.

Веб-запросы Доступ к данным реального времени из Веба и их анализ осуществляется методом копирования и вставки веб-страниц в *Excel*. Смарт-теги автоматически предоставляют возможность создания обновляемых веб-запросов

Библиотека шаблонов Office. Область задач создания документа предоставляет возможность непосредственного доступа к сотням профессионально разработанных шаблонов, находящихся в Интернете в библиотеке шаблонов *Office*.

Потоковая библиотека макетов Microsoft Design Gallery Live Доступ через Интернет к тысячам изображений, звуков и анимаций на веб-узле *Office Media Gallery*. Выполняя ежемесячное обновление, вы всегда имеете доступ к самым новым ресурсам.

Мои источники данных. В этой новой папке, вложенной в папку «Мои документы», отслеживается, к каким данным вы ранее обращались. Благодаря этому обеспечивается быстрое подключение к базам данных и другим источникам информации

Печать из обозревателя Печать *HTML*-файлов теперь выполняется непосредственно из тех приложений *Office*, в которых они были созданы, благодаря чему значительно повышается качество печатных документов.

Возможность задания веб-параметров Новое раскрывающееся меню «Веб-параметры» позволяет с лёгкостью создавать документы, совместимые с определенными веб-обозревателями

Диалоговое окно «Добавить гиперссылку» Усовершенствованное диалоговое окно «Добавить гиперссылку» позволяет без труда задать ссылки на файлы при создании и изменении веб-страниц.

Надежность, восстановление данных и безопасность:

Восстановление документов Приложения *Microsoft PowerPointR, Word, Excel* и *Access* при возникновении ошибки в программе позволяют сохранить файл, над которым вы в это время работали.

Автосохранение Функция автосохранения, которой теперь можно воспользоваться в приложениях *Word, Excel, PowerPoint* и *Publisher*, позволяет выбрать частоту сохранения файлов. Если происходит ошибка, можно восстановленный файл сохранить вместо исходного или отдельно.

Отчеты об ошибках в приложениях Приложения *Office XP* автоматически передают отчеты об ошибках непосредственно в корпорацию *Microsoft* или в отдел информационных технологий организации. Эти сведения могут быть использованы для исправления ошибок в программе или выработке соответствующих рекомендаций для пользователей

Восстановление приложений и документов. Средство восстановления приложений и документов обеспечивает безопасное завершение программ, не отвечающих на запросы системы. Оно позволяет начать восстановление документов уже при завершении работы приложения.

Восстановление и извлечение данных. Это средство восстановления документов автоматически вызывается в приложениях *Word* и *Excel* каждый раз, когда возникает ошибка или происходит сбой при загрузке документа. Оно также может быть вызвано из диалогового окна открытия файла.

Безопасный режим Office XP позволяет сократить простои в работе за счет автоматического выполнения действий по устранению типичных неполадок, встречающихся при загрузке приложений

Цифровые подписи Позволяет подписать документ *Office XP*. Благодаря цифровой подписи можно узнать, заслуживает ли доверия источник, из которого поступил документ, и не были ли внесены в этот документ изменения.

Шифрование пароля документа В приложениях *Word and Excel* усовершенствовано применение пароля за счет использования стандарта *CryptoAPI*. Впервые применяется шифрование пароля в приложении *PowerPoint*.

Единая вкладка «Безопасность» В приложении *Office XP* удобный доступ к средствам защиты осуществляется за счёт размещения на одной вкладке настроек параметров безопасности всех приложений

Обзор средств совместной работы

Отправка на проверку Новые средства позволяют назначить соответствующие роли всем, кто вовлечен в цикл проверки документа. При этом каждому рецензенту автоматически предоставляются нужные средства. В ходе проверки изменения могут быть без труда внесены в текст исходного документа

Сравнение и объединение исправлений В пакете *Office XP* легко объединить примечания и исправления нескольких рецензентов в одной версии документа, принимая или отклоняя предложенные исправления

Панель «Рецензирование» Новая панель «Рецензирование» обеспечивает возможность просмотра изменений определенного рецензента или всех рецензентов, а также возможность принятия или отклонения изменений по одному или всех сразу.

Совместная работа: интеграция со службами *Microsoft Sharepoint Team Services*

Microsoft SharePoint Team Services Удобное средство создания веб-сайта для централизованного места хранения всей информации о проекте, служба *Microsoft SharePoint Team Services* (в приложении *FrontPageR* версия 2002) обеспечивает проведение дискуссий, формирование списков контактной информации и отслеживаемых задач, ведение и создание календаря событий и библиотек документов, проведение опросов – все из окна веб-обозревателя

Библиотеки документов Библиотеки документов упрощают совместную работу, обеспечивая для веб-пользователей централизованное место хранения документов.

Диалоговые окна сохранения и открытия документа Новый *HTML*-интерфейс позволяет переходить в библиотеку документов непосредственно в диалоговых окнах сохранения и открытия документа.

Списки Структурированные списки просмотра и исправления используются в веб-интерфейсе при построении веб-сайтов *Microsoft SharePoint Team*. Предусмотрены шаблоны списков для оповещений, контактов группы, событий группы и оповещений. Списки могут изменяться или создаваться заново на основе существующих шаблонов.

События Централизованное место хранения информации о событиях группы, это новое средство позволяет всем членам группы добавлять информацию о событиях, а затем осуществлять ее экспорт непосредственно в календарь *Microsoft OutlookR*

Контакты группы Управление контактами, изменение отображаемых колонок и представления информации о контактах, экспорт и импорт этой информации, соответственно, в папку или из папки «Контакты» приложения *Outlook*.

Обсуждения документов Используя *Office XP*, можно организовывать обсуждения на базе документа *Office* или веб-узла .

Средства редактирования в приложении FrontPage Используя *FrontPage*, можно редактировать веб-сайты *Microsoft SharePoint Team*. Предусмотрено изменение темы, добавление компонентов *FrontPage* и вставка статических *HTML*-элементов.

Подключение и координация совместной работы

Совместимость форматов файлов Обратная совместимость между *Office XP*, *Office 2000* и *Office 97* (за исключением *Access 97*) обеспечивает пользователям всех версий возможность совместной работы с документами

Расширяемые смарт-теги Технология смарт-тегов обеспечивает независимым разработчикам удобную платформу для построения бизнес-решений, предназначенных пользователям *Office XP*. Пример. Когда вы вводите в приложении *Excel* номер заказа, при помощи смарт-тега с ним связываются заказчик, заказ и цена

Поддержка XML в приложениях Excel и Access Развитая поддержка технологии *XML* в пакете *Office XP* позволяет непосредственно в приложении *Excel* загружать и сохранять документы в формате *XML* и анализировать находящиеся в сети данные, запрашивая электронные таблицы в формате *XML* из Веба. В приложении *Access* можно импортировать и экспортировать схемы *XML*, данные и таблицы как применяя настройки, установленные по умолчанию, так и задавая дополнительные параметры, обеспечивающие расширенные возможности управления.

Веб-компоненты Office Веб-компоненты *Excel* поддерживают файлы, созданные в приложении *Excel*, и именованные диапазоны, позволяют использовать несколько листов, выравнивание по словам, обеспечивают публикацию в Вебе интерактивных электронных таблиц. Компонент «Сводные таблицы» предоставляют возможность условной фильтрации в диапазоне выражений. *Office XP* расширяет модель объектов, позволяя разработчикам создавать собственные технические решения для задачи распространения информации средствами веб-обозревателей.

Развертывание, управление и администрирование

Автоматизированная установка Усовершенствованная установка упрощает обновление до уровня *Office XP*, анализируя систему и устанавливая компоненты пакета *Office* в соответствии с существующей конфигурацией

Мастер настройки установки Появилась возможность удаления файлов (например, устаревших надстроек) во время установки; можно установить для каждой программы свой уровень безопасности и задать параметры настройки приложения *Outlook*

Мастер настройки сопровождения Это новое средство упрощает сопровождение и настройку после установки. Оно применяется для добавления и удаления файлов, программ и компонентов; изменения конфигурации и выполнения многих других административных задач

Мастер сохранения настройки Упрощается копирование настроек с одного компьютера на другой с помощью сохранения их в файле или передачи через Веб (используя *Microsoft Passport* для проверки подлинности) с последующим применением на другом компьютере

Поддержка HTTP программой установки Поддержка программой установки протоколов *HTTP*, *HTTPS* и *FTP* означает, что *OfficeXP* может быть установлен непосредственно с веб-сервера или общей папки корпоративной сети.

Поддержка политик Office XP обеспечивает дополнительную поддержку политик для большей части новых функций *Office XP*, позволяя администраторам устанавливать политики, управляющие настройками безопасности и областями задач

Поддержка национальных стандартов Усовершенствования поддержки национальных стандартов охватывают независимую программу установки для каждого языка, новый мастер установки пакета *Multi-Language Pack* и автоматическое определение языка операционной системы. Приложение *FrontPage*, теперь имеющееся на 26 языках (приложение *FrontPage 2000* выпускалось на 15 языках), поддерживает наборы знаков для двунаправленных текстов, Юникод, позволяет создавать веб-страницы на любом из установленных языков

Поддержка сервера терминалов Windows Благодаря обновлению средств взаимодействия между *Office* и сервером терминалов *WindowsR*, теперь сервер терминалов *Windows* может быть установлен вместе с *Office XP* без применения выборочной установки. Кроме того, при подключении к удаленному компьютеру не применяются сложные средства отображения, благодаря чему повышается скорость работы приложений

4.4 Microsoft Word

Microsoft Word (часто - *MS Word*, *WinWord* или просто *Word*) - текстовый процессор, предназначенный для создания, просмотра и редактирования текстовых документов, с локальным применением простейших форм таблично-матричных алгоритмов. Выпускается корпорацией Microsoft в составе пакета *Microsoft Office*. Первая версия была написана Ричардом Броди для *IBM PC*, использующих *DOS*, в 1983. Позднее выпускались версии для *Apple Macintosh* (1984), *SCO UNIX* и *Microsoft Windows* (1989).

WORD - мощный текстовый редактор, позволяющий быстро создать документ любой сложности из разрозненных заметок и довести до совершенства информационный бюллетень или брошюру. Редактор *Word* фирмы Microsoft является сегодня самой популярной в мире программой. *Word* начинен «быстрыми» командами и самыми современными средствами, такими как встроенная программа проверки правописания и словарь синонимов, которые помогают грамотно составлять документы, и готовыми шаблонами, позволяющими сводить воедино заметки, письма, счета и брошюры без больших усилий.

Microsoft Word многим обязан *Bravo* - текстовому процессору с оригинальным графическим интерфейсом, разработанному в центре «*Xerox PARC*». Создатель *Bravo*, Ч. Симони) покинул *PARC* в 1981. Тем же летом он переманил Броди, с которым вместе работал над *Bravo*. Первый выпуск *Word* для *MS-DOS* состоялся в 1983 и был плохо принят рынком из-за конкурирующего продукта - *WordPerfect*. Но «*Word 3.01* (1985) для *Macintosh*» усилил позиции. *Word* был полностью *WYSIWYG*-редактором (принцип «*What You See Is What You Get*» - «получаю то, что вижу»). Хотя *MS-DOS* и являлась текстовой операционной системой, лишённой графической оболочки, *Word* для *DOS* был первым текстовым процессором для *IBM PC*, который был способен отображать разметку текста, например, полужирный или курсивный текст в процессе редактирования. Однако он всё же не являлся в полном смысле *WYSIWYG*-редактором. Другие же текстовые процессоры, такие как *WordStar* и *WordPerfect*, использовали простой текстовый экран с кодами разметки, иногда текст был цветным. Однако, поскольку в большинстве программного обеспечения под *DOS* применялись труднозапоминаемые комбинации «горячих клавиш» для каждой команды (например, в *Word'e* для *DOS* сохранение файла выполнялось по комбинации *ESC-T-S*) и большинство секретарей умели пользоваться только *WordPerfect*, компании неохотно переходили на конкурирующие с ним продукты, имеющие сравнительно небольшие преимущества

Первая версия *Word* для *Windows*, выпущенная в 1989 году, продавалась по цене 500 долларов США. Она демонстрировала выбранный компанией Майкрософт путь развития: как и сама *Windows*, она многое взяла от *Macintosh*, и использовала стандартные клавиатурные сокращения (например, *CTRL-S* для сохранения файла). После выпуска в 1990 *Windows 3.0* продажи поползли вверх (*Word 1.0* гораздо лучше работал с *Windows 3.0*, чем с более старыми версиями *Windows/386* и *Windows/286*), главный конкурент – *WordPerfect* - не смог выпустить рабочую версию под *Windows*, что оказалось для него смертельной ошибкой. Версия 2.0 утвердила *WinWord* на позиции лидера рынка.

У *Word* для *Macintosh* никогда не было серьёзных конкурентов, даже несмотря на наличие программ вроде *Nisus*, предоставлявшую возможность выделения нескольких несвязных кусков текста (появившуюся только в *Word 2002* из *Office XP*) и несмотря на мнение многих пользователей об отсутствии кардинальных различий между версиями 3.01, выпущенной в 1987 и версией 5.0, выпущенной в 1991. Однако, версия 6.0 для *Macintosh* (1994) была многими воспринята скептически. Версия под Windows, следовавшая за 2.0, была пронумерована как 6.0 для координации названия версий под разные платформы.

Последовавшие версии добавляли возможности, выходящие за рамки простого текстового процессора. Инструменты рисования позволяли выполнять примитивные операции вёрстки, такие как добавление графики в документ, хотя, естественно, специализированные программы для вёрстки лучше справляются с этими задачами. Внедрение объектов, сравнение версий документа, мультязычная поддержка и многие другие возможности были добавлены за последовавшие несколько лет.

В настоящее время *Microsoft Word* является наиболее популярным из используемых в данный момент текстовых процессоров, что сделало его бинарный формат документа стандартом де-факто, и многие конкурирующие программы имеют поддержку совместимости с данным форматом. Расширение «.doc» на платформе *IBM PC* стало синонимом двоичного формата *Word 97-2000*. Фильтры экспорта и импорта в данный формат присутствуют в большинстве текстовых процессоров. Формат документа разных версий *Word* меняется, различия бывают довольно тонкими. Форматирование, нормально выглядящее в последней версии, может не отображаться в старых версиях программы, однако есть ограниченная возможность сохранения документа с потерей части форматирования для открытия в старых версиях продукта. Последняя версия *MS Word 2007* «использует по умолчанию» формат основанный на XML - *Microsoft Office Open XML*. Спецификация форматов файлов *Word 97-2007* были опубликованы *Microsoft* в 2008. Ранее, большая часть информации, нужной для работы с данным форматом, добывалась посредством обратного инжиниринга, поскольку основная её часть отсутствовала в открытом доступе или была доступна лишь ограниченному числу партнеров и контролирурующих организаций.

Как и прочие приложения из *Microsoft Office*, *Word* может расширять свои возможности посредством использования встроенного макроязыка (сначала использовался *WordBasic*, с версии *Word 97* применяется *VBA - Visual Basic* для приложений). Однако это предоставляет широкие возможности для написания встраиваемых в документы вирусов (так называемые «макровирусы»). Наиболее ярким примером была эпидемия червя *Melissa*. В связи с этим, многие считают разумной рекомендацию всегда выставлять наивысший уровень настроек безопасности при использовании *Word* (Меню: «*Tools>Macro>Security*», «*Сервис>Макрос>Безопасность...*» в локализованных русских версиях). Также нелишним будет использовать антивирусное программное обеспечение. Первым вирусом, заражавшим документы *Microsoft Word*, был *DMV*, созданный в 1994 Дж. МакНамарой для демонстрации возможности создания макровирусов. Первым же вирусом, попавшим в «дикую природу» и вызвавшим первую в мире эпидемию макровирусов (1995), был *Concept*. 12.08.2009 суд штата Техас запретил продажу программы *Word* на территории США, в связи с тем, что *Microsoft* незаконно использует метод чтения XML-файлов, патент на который принадлежит канадской компании *i4i*.

Курьёзные ошибки в модуле проверки русской орфографии Набор фразы «*Правоспособность – способность иметь права и нести гражданские обязанности*» вызывает (в случае, если включена автоматическая проверка орфографии) незамедлительное закрытие приложения; при этом все открытые документы закрываются без сохранения изменений. В *MS Word* версии 2007 данная ошибка исправлена. Модуль проверки орфографии в *MS Word* при наборе слова «мультиканальный» предлагает автозамену на фразу «мультик анальный» (как и модуль проверки орфографии в *Firefox 3*). Аналогичные курьёзы происходят с некоторыми другими словами: «*культурологии*» - «*культ урологии*»; «*стриптизёрша*» - «*стриптиз ерша*»; «*отмелькала*» - «*отмель кала*»; «*спецназовец*» - «*спецназ овец*»; «*завязочками*» - «*завяз очками*» и др. В *Word 2002* и выше пример не работает. При наборе в *MS Word 2000* фразы «*Хочу избежать службу в армии*» модуль проверки орфографии в качестве одного из вариантов исправлений предлагает текст «*Ошибка в управлении. Глагол „избежать“ требует дополнения в родительном падеже. Например: „Никому не удалось избежать службы в армии.“*»

4.5 Microsoft Excel

Microsoft Excel (также иногда называется *Microsoft Office Excel* - программа для работы с электронными таблицами, созданная корпорацией *Microsoft* для *Microsoft Windows*, *Windows NT* и *Mac OS*. Она предоставляет возможности экономико-статистических расчетов, графические инструменты и, за исключением *Excel 2008* под *Mac OS X*, язык

макро-программирования *VBA (Visual Basic для приложений)*. *Microsoft Excel* входит в состав *Microsoft Office* и на сегодняшний день *Excel* является одним из наиболее популярных приложений в мире.

EXCEL - производит с числами то же, что *Word* с существительными и глаголами. Каждый, кто работает с цифрами, почувствует себя в среде *Excel* как рыба в воде. Программой *Excel* можно пользоваться для составления бюджетов и финансовых отчетов, превращения сухих цифр в наглядные диаграммы и графики, проведения анализа типа «А что будет, если?» практически по любому вопросу, а также для сортировки длиннейших списков в считанные секунды. С помощью электронных таблиц *Excel* можно впечатывать числа строку за строкой и столбец за столбцом, будучи вполне уверенным в том, что *Excel* правильно сложит, вычтет, умножит, разделит и вообще обойдется с ними как следует.

В 1982 *Microsoft* запустила на рынок первый электронный табличный процессор *Multiplan*, который был очень популярен на *CP/M* системах, но на *MS-DOS* системах он уступал *Lotus 1-2-3*. Первая версия *Excel* предназначалась для *Mac* и была выпущена в 1985, а первая версия для *Windows* была выпущена в 1987. *Lotus* не торопилась выпускать 1-2-3 под *Windows*, и *Excel* с 1988 года начала обходить по продажам 1-2-3, что в конечном итоге помогло *Microsoft* достичь позиций ведущего разработчика программного обеспечения. *Microsoft* укрепляла свое преимущество с выпуском каждой новой версии, что имело место примерно каждые два года. Текущая версия для платформы *Windows* - *Excel 12*, также известная как *Microsoft Office Excel 2007*. Текущая версия для платформы *Mac OS X* - *Microsoft Excel 2008*. В начале своего пути *Excel* стал причиной иска о товарном знаке от другой компании, уже продававшей пакет программ под названием «*Excel*». В результате спора *Microsoft* была обязана использовать название «*Microsoft Excel*» во всех своих официальных пресс-релизах и юридических документах. Однако со временем эта практика была позабыта, и *Microsoft* окончательно устранила проблему, приобретя товарный знак другой программы. *Microsoft* также решила использовать буквы *XL* как сокращенное название программы: иконка *Windows*-программы состоит из стилизованного изображения этих двух букв, а расширение файлов по умолчанию в *Excel* - *.xls*. В сравнении с первыми табличными процессорами *Excel* представляет множество новых функций пользовательского интерфейса, но суть остается прежней: как и в программе-родоначальнике, *VisiCalc*, ячейки расставляются в строки и столбцы и могут содержать данные или формулы с относительными или абсолютными ссылками на другие клетки.

Excel был первым табличным процессором, позволявшим пользователю менять внешний вид таблицы (шрифты, символы и внешний вид ячеек). Он также первым представил метод умного пересчета ячеек, когда обновляются только те ячейки, которые зависят от измененных ячеек (раньше табличные процессоры либо постоянно пересчитывали все ячейки или ждали команды пользователя). Будучи впервые объединенными в *Microsoft Office* в 1993, *Microsoft Word* и *Microsoft PowerPoint* получили новый графический интерфейс для соответствия *Excel*, главного стимула модернизации ПК в то время.

Начиная с 1993, в состав *Excel* входит *Visual Basic для приложений (VBA)*, язык программирования, основанный на *Visual Basic*, позволяющий автоматизировать задачи *Excel*. *VBA* является мощным дополнением к приложению и в более поздних версиях *Excel* доступна полнофункциональная интегрированная среда разработки. Можно создать *VBA*-код, повторяющий действия пользователя и таким образом автоматизировать простые задачи. *VBA* позволяет создавать формы для общения с пользователем. Язык поддерживает использование (но не создание) *DLL* от *ActiveX*; более поздние версии позволяют использовать элементы объекто-ориентированного программирования.

Функциональность *VBA* делала *Excel* легкой мишенью для макровирусов. И это было серьезной проблемой до тех пор, пока антивирусные продукты не научились обнаруживать их. Фирма *Microsoft*, с опозданием приняв меры для уменьшения риска, добавила возможность выбора режима безопасности: полностью отключить макросы; включить макросы при открытии документа; доверять всем макросам, подписанным с использованием надежных сертификатов.

Макрос (*macros*, мн.ч. от *macro*) - программный объект, при обработке «развёртывающийся» в последовательность действий или команд. Во многих «офисных» продуктах (*Microsoft Office*, *OpenOffice.org* и др.), в текстовых редакторах (например, *Vim*), а также в графических программах (например, *CorelDRAW*) при обработке макроса автоматически выполняется заданная для каждого макроса последовательность действий - нажатия на клавиши, выбор пунктов меню и т. д. Предоставляется интерфейс для записи новых и перезаписи существующих макросов. Корректный перевод термина с - «**макрокоманда**», слово же «макрос» получило распространение благодаря использованию в локализованных продуктах американской корпорации *Microsoft*. К универсальным программам для создания макросов относятся линейки "*Comfort Keys*", "*Robotask*" и "*Hot Keyboard*". "*AutoHook*" используется для упрощения работы в «*AutoCAD*», причём наиболее функциональной является версия программы для "*AutoCAD 2006*", а не для более поздних версий.

Несмотря на то, что *Excel 2007* направлен на использование новых *XML*-форматов в качестве основных, он совместим с традиционными бинарными форматами. Большинство версий *Microsoft Excel* могут читать *CSV*, *DBF*, *SYLK*, *DIF* и другие форматы. *Microsoft Excel 2007*, наряду с другими продуктами *Microsoft Office 2007*, вводит множество новых форматов файлов. Они являются частью *Office Open XML* спецификации.

Табл. 2. Новые форматы *Excel 2007*

Формат	Расширение	Примечания
Рабочая книга <i>Excel</i>	<i>.xlsx</i>	Стандартный формат рабочих книг <i>Excel 2007</i> . В действительности это сжатый <i>ZIP</i> -архив каталога <i>XML</i> документов. Является заменой бывшего бинарного формата <i>.xls</i> , хотя и не поддерживает макросы по соображениям безопасности.
Рабочая книга <i>Excel</i> с макросами	<i>.xlsm</i>	Та же рабочая книга <i>Excel</i> , но с поддержкой макросов.
Бинарная рабочая книга <i>Excel</i>	<i>.xlsb</i>	Та же рабочая книга <i>Excel</i> с макросами, но хранящая информацию в бинарном формате, открывая документы быстрее чем <i>XML</i> . В особенности часто используется для очень больших документов с десятками тысяч строк и/или сотнями колонок.
Шаблон <i>Excel</i> с макросами	<i>.xltm</i>	Шаблон, созданный как основа для рабочих книг, включена поддержка макросов. Заменяет старый формат <i>.xlt</i> .
Надстройка <i>Excel</i>	<i>.xlam</i>	Надстройка <i>Excel</i> , направленная на добавление дополнительных функциональных возможностей и инструментов.

API позволяет открывать таблицы *Excel* в ряде других приложений. Это включает в себя открытие документов *Excel* на веб-страницах с помощью *ActiveX* или таких плагинов, как *Adobe Flash Player*. Проект *Apache POI* представляет *Java*-библиотеки для чтения и записи электронных таблиц *Excel*. Также предпринимались попытки копировать таблицы *Excel* в веб-приложения с использованием разделенных запятыми значений. Ценной возможностью *Excel* является возможность писать код на основе *Visual Basic* для приложений (*VBA*). Этот код пишется с использованием отдельного от таблиц редактора. Управление электронной таблицей осуществляется посредством объектно-ориентированной модели кода и данных. С помощью этого кода данные входных таблиц будут мгновенно обрабатываться и отображаться в таблицах и диаграммах (графиках). Таблица становится интерфейсом кода, позволяя легко работать изменять его и управлять расчётами.

Общая критика табличных процессоров применима и к *Excel*. Специфичные предметы критики в *Excel* - это точность, проблемы с датами и ошибки отображения *Excel 2007*. Из-за того, что *Excel* работает на основе расчетов с плавающей запятой, статистическая точность *Excel* подвергается критике. Сторонники *Excel* заявляют в ответ, эти ошибки проявляются лишь в особых условиях специально подобранных исходных данных, которые затрагивают относительно небольшое число пользователей, и с низкой вероятностью могут встретиться на практике.

4.6 Microsoft PowerPoint

PowerPoint – предназначен для профессиональной подготовке презентаций: пользователь может превратить документ, подготовленный в редакторе *Word*, в презентацию одним щелчком мыши.

Идея *PowerPoint* появилась у Б.Гаскинса, студента университета Беркли, который решил, что наступает век графических интерактивных материалов. В 1984 Гаскинс присоединился к провальной компании *Forethought* и нанял разработчика Д.Остина. Боб и Деннис объединили усилия и создали программу *Presenter*. Деннис создал оригинальную версию программы с Т. Рудкиным. Боб сменил имя на *PowerPoint*, которое и стало названием конечного продукта. В 1987 вышел *PowerPoint 1.0* для *Apple Macintosh*. Он работал в чёрно-белом цвете. Вскоре появились цветные *Macintosh* и новая версия *PowerPoint*. Программное руководство с первой редакцией было уникально. Это была книга синего цвета в твердом переплете. Компания *Forethought* считала, что это лучше чем куча исполняемых файлов помощи на компьютере и инструкций в электронном виде. Но обновление инструкции было необоснованно дорого. Идея с подобными книгами была вскоре заброшена. В 1987 *Forethought* и её продукт были куплены *Microsoft*. В 1990 вышла версия для *Windows*. С 1990 *PowerPoint* - стандарт в наборе программ *Microsoft Office*. В 2002 вышла версия *PowerPoint*, которая была включена в пакет *Microsoft Office XP*, и распространялась как отдельный продукт. В нем появились такие функции, как сравнение и смешение изменений в презентации, возможность задавать пути анимации для индивидуальных форм, создание пирамидальных, радиальных и целевых диаграмм, а также кругов Эйлера, панель задач для просмотра и

выбора объектов буфера обмена, защита презентации паролем, автоматическая генерация фотоальбома, а также «умные теги» для быстрого выбора формата текста, скопированного в презентацию.

Microsoft PowerPoint 2003 не сильно отличается от предшественника. Он увеличивает эффективность работы в группе и теперь имеет возможность создания «Пакета для CD», который легко позволяет скопировать презентацию с мультимедийным содержанием и просмотрщик на компакт диск. Текущая версия *PowerPoint 2007* привносит основные изменения в интерфейс программы и увеличивает графические возможности. *PowerPoint* является частью *Microsoft Office*. Это позволило *PowerPoint* стать наиболее распространенной во всем мире программой для создания презентаций. Файлы презентаций *PowerPoint* часто пересылаются пользователями программы на другие компьютеры, что означает необходимую совместимость с ними программ конкурентов. Однако, поскольку *PowerPoint* имеет возможность подключения элементов других приложений через *OLE*, некоторые презентации становятся сильно привязанными к платформе *Windows*, что делает невозможным открытие данных файлов, например, в версии для *Mac OS*. Это привело к переходу на открытые стандарты, такие как *PDF* и *OASIS OpenDocument*.

4.7 Microsoft Access

ACCESS - представляет собой мощную программу управления данными, предназначенную главным образом для программистов. *Access* поставляется только в составе профессионального издания пакета *Office*. *Microsoft Access* - реляционная СУБД корпорации *Microsoft*. Имеет широкий спектр функций, включая связанные запросы, сортировку по разным полям, связь с внешними таблицами и базами данных. Благодаря встроенному языку *VBA*, в самом *Access* можно писать приложения, работающие с базами данных.

Основные компоненты *MS Access*: построитель таблиц; построитель экранных форм; построитель *SQL*-запросов (язык *SQL* в *MS Access* не соответствует стандарту *ANSI*); построитель отчетов, выводимых на печать. Они могут вызывать скрипты на языке *VBA*, поэтому *MS Access* позволяет разрабатывать приложения и БД практически «с нуля» или написать оболочку для внешней БД. *MS Access* является файл-серверной СУБД и потому применима лишь к маленьким приложениям. Отсутствует ряд механизмов необходимых в многопользовательских БД, таких, например, как транзакции. Опыт показывает, что даже для проектов на 5-20 пользователей предпочтительно использовать клиент-серверные решения.

4.8 Microsoft FrontPage

FrontPage - *WYSIWYG*-редактор *HTML*, входящий в состав пакета приложений *Microsoft Office*.

Данное приложение при разработке страниц веб-узла пользуется движком *Internet Explorer (Trident)*. В браузерах, использующих другие движки (например, *Gecko*), страницы, созданные с помощью *FrontPage* могут отображаться по-другому. Тем не менее, при правильном использовании режимов совместимости, *FrontPage* позволяет создавать сайты, отображающиеся одинаково в любом браузере. Программа обладает широким спектром возможностей, в частности, может автоматически отправлять изменения, внесённые разработчиком сайта в исходные тексты, в режиме реального времени. В *Microsoft Office 2007* программа *FrontPage* была заменена на *SharePoint Designer* и *Microsoft Expression Web*.

4.9 Microsoft Outlook

Microsoft Outlook – персональный информационный менеджер с функциями почтового клиента и Groupware компании Майкросот, входящий в пакет офисных программ *Microsoft Office*. Помимо функций почтового клиента для работы с электронной почтой, *Microsoft Outlook* является полноценным органайзером, предоставляющим функции календаря, планировщика задач, записной книжки и менеджера контактов. Кроме того, *Outlook* позволяет отслеживать работу с документами пакета *Microsoft Office* для автоматического составления дневника работы. *Outlook* может использоваться как отдельное приложение, так и выступать в роли клиента для почтового сервера *Microsoft Exchange Server*, что предоставляет дополнительные функции для совместной работы пользователей одной организации: общие почтовые ящики, папки задач, календари, конференции, планирование и резервирование времени общих встреч, согласование документов. *Microsoft Outlook* и *Microsoft Exchange Server* являются платформой для организации документооборота, так как они обеспечены системой разработки пользовательских плагинов и скриптов, с помощью которых возможно программирование дополнительных функций документооборота (и не только документооборота), не предусмотренных в стандартной поставке.