

ЛЕКЦИЯ №1

Тема: Программное обеспечение ПК

1.1 Классификация ПО.....	1
1.2 Операционные системы.....	5
1.3 Прикладное ПО. Microsoft Office System.....	6
1.4 Подготовка текстовых документов в системе Microsoft Office Word	10

1.1 Классификация ПО

Все программное обеспечение (ПО, software) по назначению и использованию можно разделить на 2 большие группы - системное и прикладное.

Системное ПО - совокупность программ для обеспечения работы компьютера.

Прикладное ПО - комплекс программ для решения задач определенного класса конкретной предметной области.

В то же время, в состав основных системных средств - операционных систем (ОС) - входят и компоненты прикладного ПО, например, простейшие текстовые редакторы, программы для работы с мультимедиа, с Интернетом и другие. В составе прикладного ПО можно выделить особую группу - средства разработки ПО, назначение которых - разработка новых программ, в том числе и операционных систем. Так что эти 2 большие группы имеют некоторое пересечение по своим компонентам. Многие другие группы ПО также имеют общие компоненты, например система автоматизации управления предприятиями работает с системой управления базой данных (СУБД), имеет средства разработки, может обмениваться информацией с САПР, и т. п. На рисунке 3.1 показана классификация ПО,

которая учитывает наличие общих компонентов в различных видах ПО и взаимодействие различных видов ПО.



Рисунок. 3.1. Классификация программного обеспечения

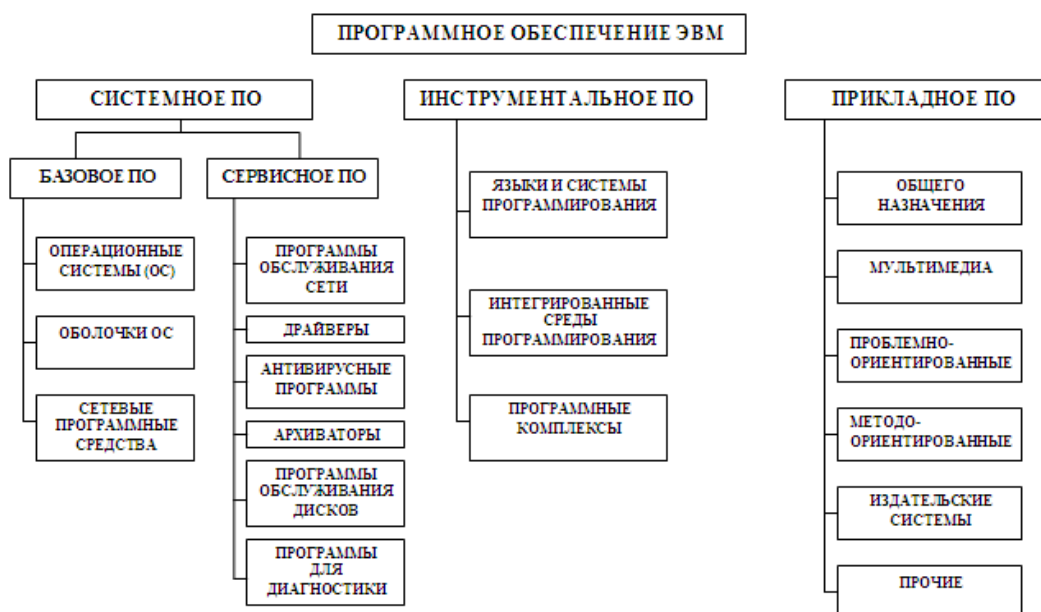


Рис. Структура программного обеспечения

Без системного ПО не может работать ни один компьютер. Наиболее распространенной операционной системой в настоящее время является **Microsoft Windows**. Драйверы (программы, обеспечивающие работу ОС со всеми компонентами ПК) могут присутствовать в составе ОС, но для самых новых материнских плат и внешних устройств поставляются отдельно. В сервисное ПО входят программы проверки дисков и файловой системы, антивирусные программы, архиваторы, программы записи оптических дисков, программы обслуживания компьютерных сетей и пр.

Среди прикладного ПО особое место по сложности разработки, внедрения и эксплуатации, по количеству компонентов системы, по объему обрабатываемой информации занимают системы комплексной автоматизации управления предприятиями, на верхнем уровне которых – ERP-системы (ERP – Enterprise Resource Planning). В составе ERP-системы – несколько десятков подсистем, работающих с информацией корпорации (**Планирование, Производство, Снабжение, Склады и сбыт, Финансы** и пр.), вся информация хранится на сервере баз данных **Oracle** или **Microsoft SQL Server**, в систему входят также средства разработки модулей для работы с базой данных, средства для взаимодействия с системами автоматизации проектирования (САПР) и другими системами.

Среди основной массы пользователей ПК обязательным компонентом ПО, установленного на компьютере, являются офисные системы, в состав которых входят текстовый редактор, табличный процессор, СУБД, органайзер и средства для работы с электронной почтой, средства разработки презентаций и пр. В нашей стране это обычно система **Microsoft Office**.

Сервисное ПО – это программы и программные комплексы, которые расширяют возможности базового программного обеспечения и организуют более удобную среду работы пользователя.

Это набор сервисных, дополнительно устанавливаемых программ, которые можно классифицировать по функциональному признаку следующим образом:

Драйверы – это специальные программы, обеспечивающие взаимодействие ОС с аппаратными устройствами (драйверы видеокарты, графического ускорителя, CD-ROM, клавиатуры, мыши, модема, сетевой карты и т.д.). При включении компьютера производится загрузка драйверов в оперативную память. Пользователь имеет возможность вручную установить или переустановить драйверы.

Программы диагностики работоспособности компьютера позволяют проверить конфигурацию ПК, выявить дефекты дисков и предотвратить потерю данных, хранящихся на дисках.

Антивирусные программы – это программы, для обнаружения, удаления и защиты от компьютерных вирусов.

Инструментальное ПО (Software tools) - программное обеспечение, используемое в ходе разработки, корректировки или развития других программ: редакторы, компиляторы, отладчики, вспомогательные системные программы, графические пакеты и др.

1. Системы программирования - это набор специализированных программных продуктов, которые являются инструментальными средствами разработчика. Программные продукты данного

класса поддерживают все этапы процесса программирования, отладки и тестирования создаваемых программ.

Множество различных приложений на компьютере создаётся с помощью языков и систем программирования.

Язык программирования - это формализованный язык описания алгоритмов, используемых для решения различных задач на компьютере.

Популярные системы программирования — Turbo Basic, Quick Basic, Turbo Pascal, Turbo C.

Современные системы программирования обычно предоставляют пользователям мощные и удобные средства разработки программ.

В них входят:

- 1.1 Компилятор или интерпретатор;
- 1.2 Интегрированная среда разработки;
- 1.3 Средства создания и редактирования текстов программ;
- 1.4 Библиотеки стандартных программ и функций;
- 1.5 Отладочные программы, т.е. программы, помогающие находить и устранять ошибки в программе;
- 1.6 Многооконный режим работы;
- 1.7 Графические библиотеки; утилиты для работы с библиотеками
- 1.8 Встроенный ассемблер;
- 1.9 Встроенная справочная служба;

Любая система программирования может работать только в соответствующей ей ОС, под которую она создана, однако при этом она может позволять разрабатывать программное обеспечение и под другие ОС.

Для того, чтобы компьютер мог понять программу, написанную на каком-то языке программирования, необходим переводчик (транслятор) такой программы в машинные коды.

Трансляторы языка программирования – это программа, предназначенная для преобразования программ, написанных на языках программирования, в машинный код.

Трансляторы делятся на два класса: **компиляторы** и **интерпретаторы**.

Компилятор преобразует (транслирует) всю программу в модуль на машинном языке, после этого программа записывается в память компьютера и лишь потом исполняется.

Интерпретатор - это транслятор, производящий покомандную обработку и выполнение исходной программы. Интерпретатор в отличие от транслятора не выдает результирующую программу или код.

Ассемблеры переводят программу, записанную на языке ассемблера (автокода), в программу на машинном языке.

Прикладные программы (Application software) - это комплекс программ, предназначенных для решения определенного класса задач. Основное назначение – дать пользователю средство обработки информации, которое не требует знаний языков программирования.

В состав **Прикладного ПО** входят:

Пакеты прикладных программ различного назначения;

Рабочие программы пользователя и ИС в целом.

ППП является мощным инструментом автоматизации решаемых пользователем задач, практически полностью освобождая его от необходимости знать, как выполняет компьютер те или иные функции и процедуры по обработке информации.

Различают следующие типы ППП:

Общего назначения (универсальные) – это универсальные программные продукты, предназначенные для автоматизации разработки и эксплуатации функциональных задач пользователя и ИС в целом.

К этому классу относятся:

Программы, обрабатывающие тексты

Современные программы обработки текстов в зависимости от возможностей делятся на несколько категорий:

Текстовые редакторы. Это программы, предназначенные для ввода и редактирования текстов (**Norton Commander**, редактор **Блокнот**, **WordPad**).

Текстовые процессоры. Они позволяют не только вводить и редактировать текст, но и форматировать его, т.е. оформлять (**Лексикон, MultiEdit, MS Word**).

Настольные издательские системы. Назначение этого класса: автоматизация процесса верстки полиграфических изданий (**Wentura, Page Maker**).

Графические редакторы - это программы, позволяющие создавать и редактировать рисунки и другие изображения (**Paint**, включаемая в состав Windows).

Электронные таблицы - это особый вид прикладных программ, предназначенные для хранения и обработки информации, представленной в табличной форме (**MS Excel, Lotus 1-2-3**).

Особенность этих программ заключается в том, что с их помощью можно не только вводить данные в ячейки таблиц, редактировать и форматировать их, но и применять формулы для описания связи между значениями, хранящимися в разных ячейках. Изменение содержимого какой-либо ячейки приводит к перерасчету значений всех ячеек.

Применение электронных таблиц упрощает работу с данными и позволяет получать результаты без проведения расчетов или специального программирования.

Электронные таблицы нашли широкое применение в экономических и бухгалтерских расчетах. Их можно эффективно использовать для: автоматической обработки зависящих друг от друга данных, автоматизации итоговых вычислений, создания сводных таблиц, ведения простых баз данных, подготовки табличных документов, построения диаграмм.

Системы управления базами данных

База данных – это совокупность массивов и файлов данных, организованная по определенным правилам, предусматривающим стандартные принципы описания, хранения и обработки данных независимо от их вида.

Работа с БД требует решения различных задач: создание базы, запись базы данных в базу, корректировка данных, выборка данных из базы по запросам пользователя.

В ПЭВМ обычно используются реляционные базы данных. В таких БД является таблицей. Столбцы называются полями, строки – записями. Примером БД может служить расписание движения поездов и автобусов, телефонный справочник.

Для разработки программ, систем программ, работающих с базами данных, используются специальные средства – системы управления базами данных (СУБД). К ним относятся: **MS FoxPro, MS Access**.

Мультимедиа - одновременное использование в едином объекте различных форм представления информации и её обработки. Например, в одном объекте может содержаться текстовая, аудиальная, графическая и видео информация, а также, возможно, способ интерактивного взаимодействия с ней.

Методо-ориентированные ППП - включает программные продукты, обеспечивающие, независимо от предметной области и функции информационных систем, математические, статические и другие методы решения задач. Наиболее распространены методы математического программирования, решение дифференциальных уравнений, имитационного моделирования, исследования операций.

Проблемно-ориентированные ППП - это программные продукты, предназначенные для решения какой-либо задачи в конкретной функциональной области.

Выделяются группы ППП для комплексной автоматизации функций управления:

в промышленной сфере (Oracle, PRISM, Галактика (Россия))

в непромышленной сфере (банковские, финансовые, правовые ППП)

отдельных предметных областей (ППП бухгалтерского учета (1С.Бухгалтерия, Офис), ППП финансового менеджмента (ЭДИП, Инвестор), ППП правовых справочных систем (Консультант плюс, Гарант)

Помимо классификации ПО по-назначению существует еще одна классификация – рыночная.

Лицензионные или **коммерческие программы** (*commercial ware*) распространяется на коммерческой основе. Для получения права на использование таких программ необходимо купить экземпляр дистрибутива такой программы. Порядок использования программы оговаривается в лицензионном соглашении, подлинность ПО подтверждается сертификатом (**Certificate of Authenticity**), имеющим уникальный код каждой копии дистрибутива (product key).

Бесплатные программы (*freeware*) – распространяются авторами бескорыстно. Чаще всего это небольшие программы, написанные одним или несколькими программистами для себя, и предложенные ими для общего пользования. Такие программы можно найти в Интернете.

Условно-бесплатные программы (shareware) занимают промежуточное положение между коммерческими и бесплатными программами. Самая массовая группа программ, в которую входят практически все утилиты, а часто и весьма серьезные, нужные многим программные пакеты. Эти программы предоставляются бесплатно, однако по истечении определенного срока необходимо заплатить их автору или распространителю небольшую сумму. В противном случае программа либо не загружается, либо начинает надоедать просьбами о ее регистрации.

Пробные версии программ (trialware) – это полноценные версии коммерческих систем, которые можно использовать какое-то время бесплатно. По истечении этого времени программы прекращают работать.

Демо-версии (demoware) – демонстрационные версии коммерческих программ и игр с ограниченными возможностями. Например, с выключенной функцией сохранения документов в текстовом редакторе, с ограниченным количеством записей в базе данных информационной системы или с несколькими турами в играх.

Крупнейшим производителем коммерческого ПО является корпорация **Microsoft** (см. www.Microsoft.ru или www.Microsoft.com), которая производит следующие группы ПО:

- 1) операционные системы (**Windows**);
- 2) офисное ПО (**Microsoft Office System**);
- 3) средства для разработчиков-программистов (**Microsoft Visual Studio** и пр.);
- 4) сервер баз данных (**Microsoft SQL Server**);
- 5) системы автоматизации управления предприятиями (**Microsoft Dynamics**);
- 6) ПО для КПК, коммуникаторов;
- 7) игры.

1.2 Операционные системы

Операционная система - комплекс программ, обеспечивающих работу компьютера.

Функции современной ОС для ПК:

1. Управление конфигурацией компьютера.
2. Управление процессами, потоками и заданиями.
3. Управление памятью.
4. Обеспечение информационной безопасности.
5. Управление подсистемой ввода-вывода.
6. Управление внешней памятью.
7. Управление файловой системой.
8. Поддержка сетей.

В управлении конфигурацией компьютера (в ОС **Windows**) ключевую роль играют данные, сохраняемые в реестре этой системы (база данных общесистемных и пользовательских параметров) и инструментарий управления **Windows** (WMI).

Процесс в ОС - исполняемая программа и ее данные в закрытом адресном пространстве, данные защиты и прочие объекты, принадлежащие процессу.

Поток - некоторый компонент внутри процесса, который получает процессорное время для выполнения.

Задание (job) - совокупность процессов, управляемых как единая группа.

ОС управляет виртуальной памятью, которую проецирует на физическую память компьютера.

Информационная безопасность предусматривает защиту любых разделяемых системных объектов (файлов, каталогов, процессов, потоков и т. д.), аутентификацию паролей при регистрации, предотвращение несанкционированного доступа, аудит событий в системе (ведение журналов событий).

Диспетчер ввода-вывода ОС подключает приложения и системные компоненты к виртуальным, логическим и физическим устройствам, а также определяет инфраструктуру, поддерживающую драйверы устройств.

Управление внешней памятью включает организацию работы с жесткими и гибкими дисками, с оптическими дисками, магнитными лентами, флэш-памятью и другими устройствами.

Управление файловой системой, определяющей методы хранения информации на дисках, важная функция любой ОС. Многие ОС поддерживают работу с несколькими файловыми

системами, например, **Windows 2000, XP** и его более новые версии могут управлять файловыми системами **NTFS** (основная для магнитных дисков), **FAT12, FAT16, FAT32** (были основными для **DOS** и **Windows 3.1 - Me**), **CDFS** и **UDF** (для лазерных дисков). Для управления используется драйвер файловой системы, который через драйверы устройств внешней памяти работает с информацией логических томов. Основные функции работы с информацией файловой системы – **CreateFile** (создать файл), **ReadFile** (читать файл) и **WriteFile** (записать в файл).

Поддержка сетей присутствует во всех современных ОС. В ОС **Windows** появилась с версии 3.11. К четырем базовым типам сетевого программного обеспечения относятся сетевые службы, программные интерфейсы сетевого программного обеспечения (сетевые API), протоколы и драйверы устройств сетевых адаптеров. Многоуровневая архитектура поддержки сетей обеспечивает ее легкую расширяемость при появлении новых сетевых протоколов и нового оборудования.

1.3 Прикладное ПО. Microsoft Office System

Система **Microsoft Office System** включает в себя серверы, службы, решения и приложения. Эти четыре компонента созданы для совместной работы при решении широкого круга бизнес-задач предприятий.

В системе представлены структурные блоки для создания решений, которые позволяют выполнять следующие действия:

- ♦ предоставляют сотрудникам подразделений удобный доступ к данным предприятия;
- ♦ обеспечивают эффективную и гибкую совместную работу групп и организаций;
- ♦ дают возможность большому количеству сотрудников подразделений организации участвовать в создании бизнес-сред растущей сложности.

Система **Microsoft Office System** имеет в своем составе следующие приложения:

Microsoft Office Access – СУБД для работы с собственным форматом реляционной базы данных и с сервером баз данных **Microsoft SQL Server**;

Microsoft Office Communicator – позволяет пользователям легко общаться и работать с другими пользователями в различных географических пунктах с помощью обмена мгновенными сообщениями, общего доступа к рабочему столу, голосовой и видеосвязи;

Microsoft Office Excel – табличный процессор для выполнения математических вычислений и оформления результатов;

Microsoft Office FrontPage – создание веб-узлов и управление ими, разработка веб-страниц;

Microsoft Office Groove – обеспечение совместной работы пользователей за счет создания общих рабочих областей групп; позволяет синхронизировать процесс обработки данных участниками таких групп при работе в Инtranете и Интернете, либо при подключении к рабочей области после оффлайн-работы;

Microsoft Office InfoPath – создание многофункциональных динамических бизнес-форм для сбора данных и обмена ими внутри широкого круга бизнес-процессов различной сложности на основе языка [XML](#) и программирования на Jscript или VBScript. Источниками данных могут быть база данных системы **Microsoft SQL Server**, **Microsoft Access**, файл XML-данных или веб-служба;

Microsoft Office OneNote – обеспечивает возможность записи данных в различной форме на переносном, настольном или планшетном компьютере, включая вводимые с помощью клавиатуры и рукописные заметки, нарисованные от руки схемы, звукозаписи, фотографии и рисунки из Интернета, а также данные из других приложений.

Microsoft Office Outlook - диспетчер связей и органайзер: работа с электронной почтой, календарь для планирования встреч и совещаний, база адресов, список задач и заметок;

Microsoft Office PowerPoint - дизайнер презентаций;

Microsoft Office Project - управление корпоративными проектами; использование эффективных средств анализа данных и создания отчетов для оптимизации использования ресурсов, определения приоритетных направлений работы и обеспечения соответствия проектов общим бизнес-целям;

Microsoft Office Publisher - создание публикаций для печати, веб-узлов и электронной почты на основе шаблонов;

Microsoft Office SharePoint Designer - создание и настройка веб-узлов **Microsoft SharePoint** и разработки интерактивных решений по организации рабочих процессов с помощью технологий

SharePoint - создание настраиваемых представлений данных, отчетов, средств, в том числе отображающих данные из нескольких источников;

Microsoft Office Visio - работа с бизнес-диаграммами и техническими диаграммами;

Microsoft Office Word - текстовый процессор; может также использоваться для редактирования HTML и XML файлов.

Дополнительные приложения:

■ **Microsoft Office Picture Manager** - просмотр, редактирование и печать файлов изображений форматов JPEG, GIF, BMP, PNG, TIFF, WMF, EMF;

■ **Microsoft Office Document Imaging** - сканирование и просмотр изображений. Для настройки и выполнения сканирования использует компонент **Microsoft Office Document Scanning**.

Система **Microsoft Office System** имеет в своем составе следующие серверы:

♦ **Microsoft Office Forms** - расширяет взаимодействие с клиентами, партнерами и поставщиками с помощью простых в использовании электронных форм;

♦ **Microsoft Office Groove Server** - предоставляет возможности для определения присутствия и немедленного обмена сообщениями, передачи файлов, организации видео- и аудиоконференций, ведения журналов, архивирования.

♦ **Microsoft Office Project Server** - платформа для поддержки функций управления корпоративными проектами и ресурсами, а также совместной работы в рамках решений для корпоративного управления проектами. Решение для корпоративного управления проектами **-Microsoft Office Enterprise Project Management (EPM)** включает дополнительный компонент **Microsoft Office Project Web Access**.

♦ **Microsoft Office SharePoint Server** - позволяет организовать централизованный доступ ко всей ключевой деловой информации и приложениям, а также обеспечить распределенное использование информации файловых серверов, баз данных, общих папок, интернет-сайтов, а также веб-сайтов на базе Microsoft SharePoint Team Services.

Корпорация Microsoft поставляет следующие основные дистрибутивные наборы приложений Office System 2007 (<http://offwe.microsoft.com/en-us/products/FX101635841033.aspx>)

Приложения Microsoft Office System 2007	Дистрибутивные наборы, их цена и состав					
	Home and Student	Standard	Small Business	Profes- sional	Ulti- mate	Enter- prise
	\$149.95	\$399.95	\$449.95	\$499.95	\$679.95	-
Word	●	●	●	●	●	●
PowerPoint	●	●	●	●	●	●
Excel	●	●	●	●	●	●
OneNote	●				●	●
Outlook		●				
Outlook with Business Contact Manager			●	●	●	●
Access				●	●	●
Accounting Express			●	●	●	
Publisher			●	●	●	●
Groove					●	●
InfoPath					●	●
Communicator						●

Система **Microsoft Office System** имеет в своем составе следующие сервисы:

♦ **Microsoft Office Live Meeting** - сервис для проведения веб-конференций. Позволяет в реальном времени общаться с партнерами, коллегами и заказчиками, в группах до 2000 участников.

♦ **Microsoft Office Online** - веб-ресурс для продуктов **Microsoft Office System**. Содержит статьи, материалы справочной системы, обновления и исправления для приложений, описание интегрированных Интернет-служб партнеров, наборы картинок, шаблонов, бланков деловых документов, пошаговые обучающие курсы и многое другое.

Пользовательский интерфейс приложений **Microsoft Office** до версии 2007 показан на рисунке 3.32.

Во время работы использовались системы меню, панели инструментов, области задач и диалоговые окна. При этом существовала избыточность интерфейса: некоторые команды могли выполняться с помощью панели инструментов и меню системы, а также контекстного меню (все они были настраиваемыми).

Версия **Microsoft Office 2007** кардинальным образом отличается от предыдущих версий по пользовательскому интерфейсу. В новой версии дублирование команд убрано. Традиционные меню и панели инструментов заменены лентой (см. рисунок 3.33).

Лента - новый вид интерфейса, в виде набора вкладок, каждая из которых содержит набор сгруппированных элементов для различных команд и управления настройками при работе.

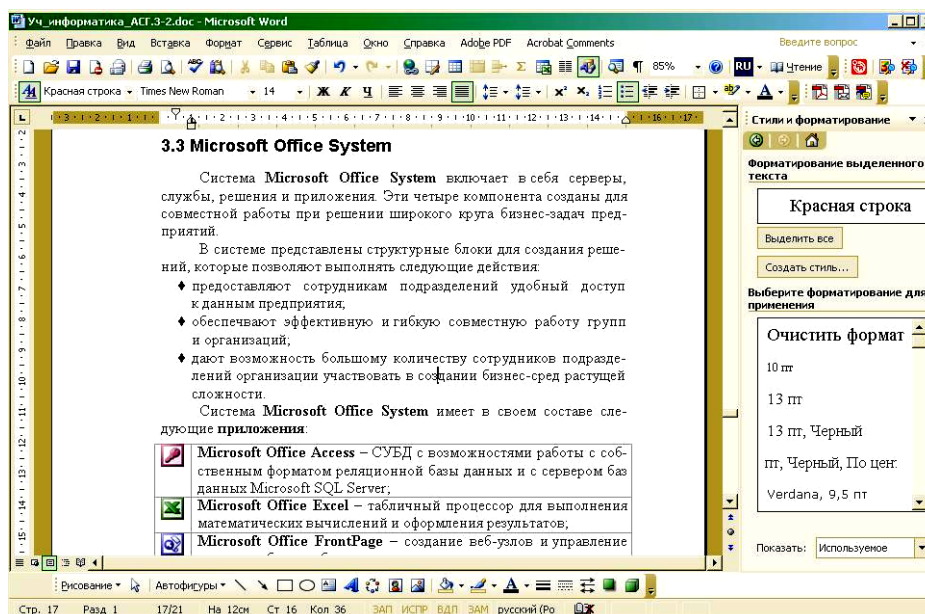


Рисунок 3.32. Интерфейс Microsoft Office Word 2003

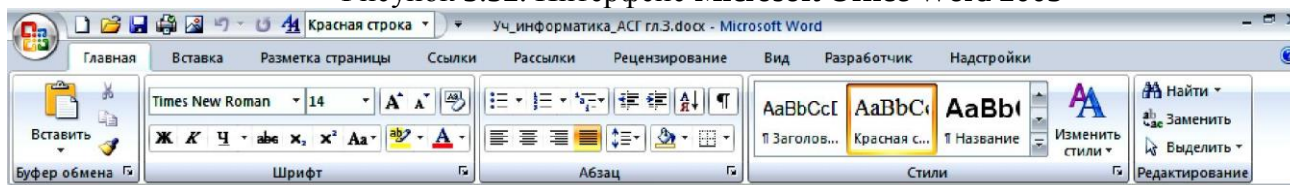


Рисунок 3.33. Вкладка ленты Главная Microsoft Office Word 2007

В **Word 2007** имеются постоянные вкладки ленты **Главная**, **Вставка**, **Разметка страницы**, **Ссылки**, **Рассылки**, **Рецензирование**, **Вид**, **Разработчик** для набора текста, вставки, разметки страницы, работы со ссылками, почтовых рассылок, рецензирования документов и работы с формами, макросами и скриптами. В **Excel 2007** имеет похожий набор вкладок ленты с командами для работы с электронными таблицами: создание новых листов, добавление таких объектов, как диаграммы и рисунки, разметка страницы, работа с формулами, управление данными и рецензирование. По замыслу разработчиков системы, новый интерфейс должен увеличить простоту и скорость доступа к элементам управления работой и оформлением документов. Однако, переход со старых версий, если пользователь длительное время пользовался ими, к новой, достаточно сложен, т. к. привычное расположение команд настройки полностью изменилось. Для освоение нового интерфейса требуется достаточно много времени, после чего его преимущества действительно становятся реальны.

Приблизиться к некоторым привычным элементам в интерфейсе позволяет настраиваемая панель быстрого доступа, которая может располагаться в заголовке окна справа от главной круглой кнопки окна (которая содержит все же меню с важными командами!) или под лентой.

В настройках панели быстрого доступа, показанных на рисунке 3.34, добавлены кнопки для часто используемых команд **Создать**, **Быстрая печать**, **Вставить рисунок из файла**, **Стили**, а также **список Стиль**, отображающий текущий стиль и позволяющий быстро задать новый.

К привычным элементам управления и настройки относятся контекстные меню, появляющиеся при щелчке правой кнопкой мыши на элементах в окне системы.

К новым элементам интерфейса относится мини-панель инструментов, которая автоматически появляется полупрозрачной над выделенным фрагментом текста и становится более четкой при наведении на нее курсора. Ее можно использовать для форматирования выделенного фрагмента: применения полужирного начертания или курсива, изменения размера и цвета шрифта и т. д. При удалении от нее курсора мини-панель плавно исчезает.

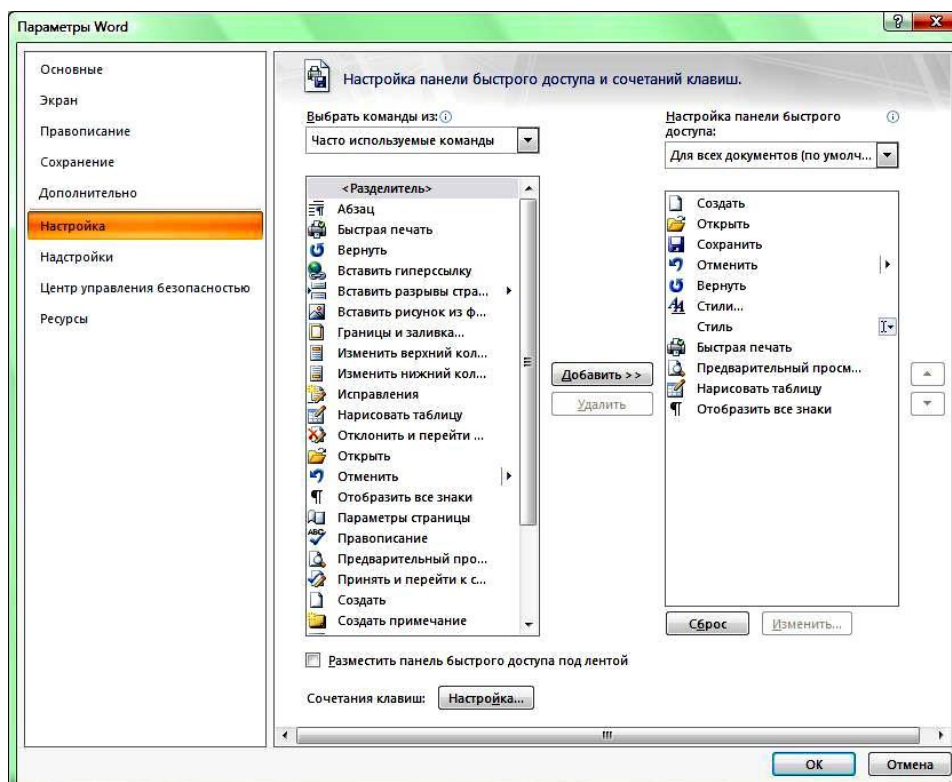


Рисунок 3.34. Настройка панели быстрого доступа

Кроме набора основных вкладок ленты, новая версия **Office** имеет контекстные вкладки, появляющиеся при работе с объектами, которые могут быть созданы с использованием вкладки ленты **Вставка**:

- Работа с таблицами (**Конструктор, Макет**);
- Работа с рисунками (**Формат**);
- Средства рисования (**Формат**);
- Работа с рисунками SmartArt (**Конструктор, Формат**);
- Работа с диаграммами (**Конструктор, Макет, Формат**);
- Работа с колонтитулами (**Конструктор**);
- Работа с надписями (**Формат**);
- Работа с объектами WordArt (**Формат**);
- Работа с формулами (**Конструктор**).

Например, при работе с таблицами, появляется надпись на заголовке окна **Работа с таблицами** над двумя новыми вкладками – **Конструктор** и **Макет** (см. рисунок 3.35).

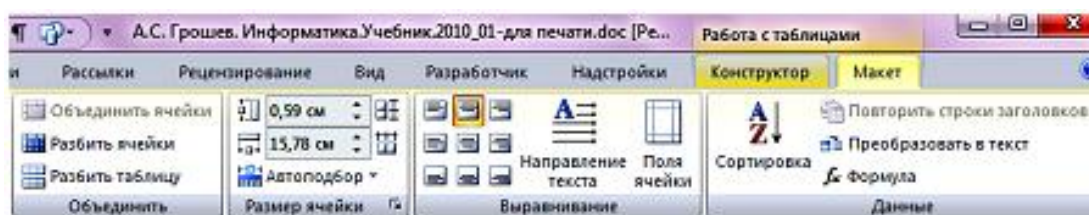


Рисунок 3.35. Контекстно-зависимые ленты работы с таблицами

Вместо сложных диалоговых окон с многочисленными параметрами настройки различных объектов (стилей, таблиц, диаграмм и пр.) при работе с документом, электронной таблицей, презентацией или базой данных предлагается использовать галереи для получения необходимых результатов. Представляя простой набор возможных вариантов, галереи упрощают создание

профессионально оформленных работ. Пользователи, которым требуется большая степень контроля результатов работы, могут по-прежнему использовать традиционные диалоговые окна.

Благодаря новой технологии динамического предварительного просмотра пользователь может увидеть результаты изменения формата, наведя указатель мыши на вариант оформления, содержащийся в галерее. Эта новая динамическая возможность позволяет в процессе разметки, редактирования и форматирования затрачивать меньше времени на выбор необходимого варианта оформления (см. рисунок 3.36).

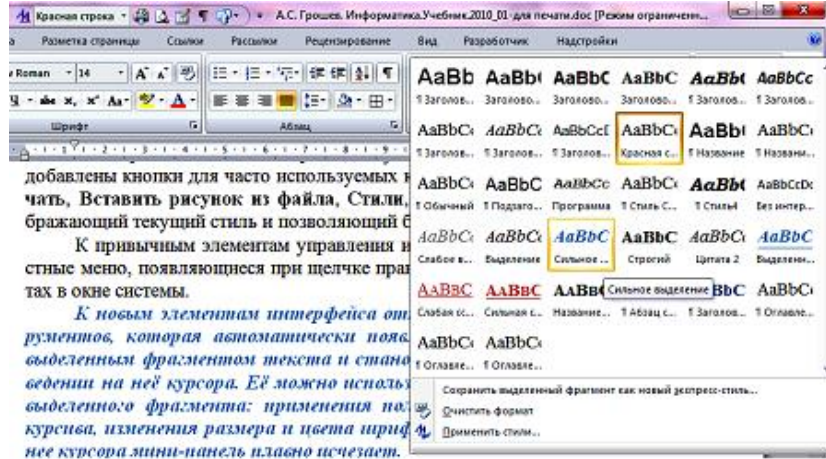


Рисунок 3.36. Галерея выбора стиля с динамическим изменением стиля текущего абзаца текста

1.4 Подготовка текстовых документов в системе Microsoft Office Word

Текстовый процессор Microsoft Office Word может быть использован для подготовки документов сложной структуры, состоящих из большого количества глав, разделов, подразделов, имеющих таблицы,

214

3.3 Microsoft Office System

сложные математические, химические и пр. формулы, рисунки и диаграммы, которые могут создаваться в данной системе или добавляться, как объект другой системы с помощью технологии OLE.

Эти документы должны соответствовать определенным стандартам, обязательным при оформлении деловых документов и научных отчетов, или строгим правилам оформления статей для различных журналов, текстов научных монографий для издательств, стилю оформления вебстраниц различных сайтов. Требования стандартов на научные отчеты и требования различных издательств могут сильно отличаться друг от друга. В связи с этим высокую актуальность имеет правильная технология подготовки документов с использованием стилей оформления и шаблонов документов.

Стиль - именованный набор параметров оформления абзаца, знака или таблицы.

Стиль определяет вид шрифта, параметры абзаца, табуляции, границ; язык, текст без рамки или в рамке, абзац без нумерации или нумерованный и горячие клавиши задания данного стиля (см. рисунок 3.37).

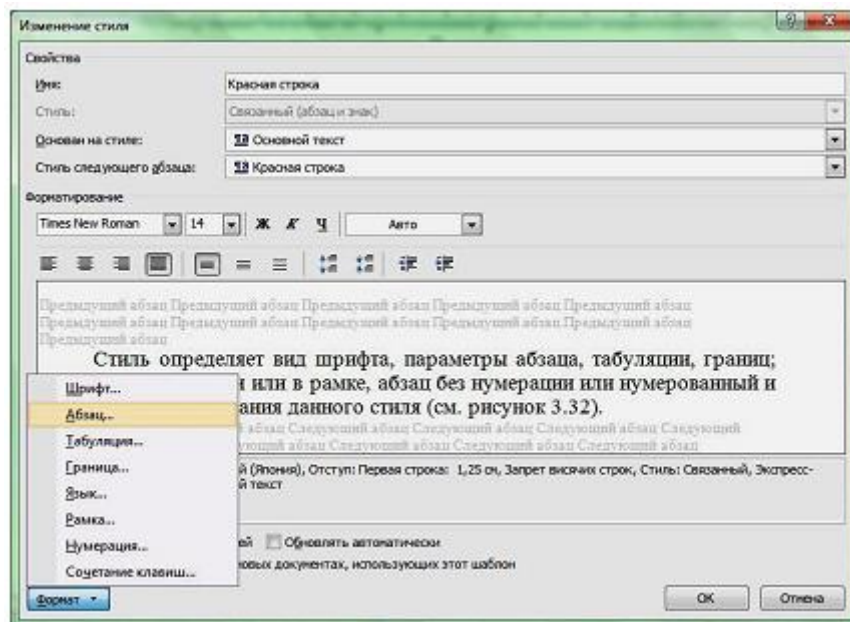
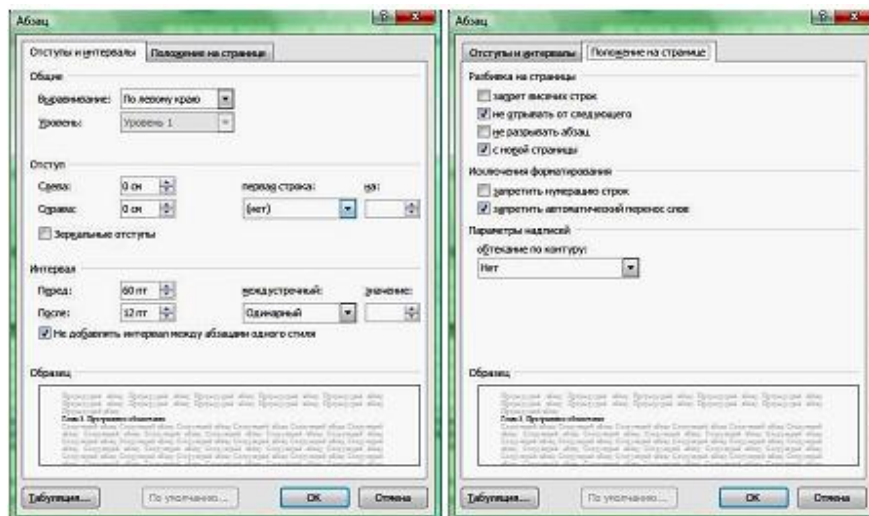


Рисунок 3.37. Окно настройки стилей

Основные параметры абзацев в стилях:

- выравнивание (по левому краю, по центру, по правому краю, по двум сторонам)
- отступы (слева, справа, первой строки)
- интервалы (перед абзацем, после него, межстрочный)
- положение на странице – запрет отрывать первую или последнюю строки, разрывать абзац между страницами, переносить слова по слогам, нумеровать строки; начинать абзац с новой страницы (см. рисунок 3.38).

Рисунок 3.38. Окно настройки параметров абзаца (стиль **Заголовок 1**)

При создании нового документа в приложении **Microsoft Office Word 2007** команда круглой кнопки **Office ► Создать** открывает диалоговое окно с вариантами:

- новый документ; – XML- документ; – на основе шаблонов
- отчеты ○ письма
- резюме ○ факсы.

В **Office Word 2003** команда меню **Файл ► Создать...** предлагает в области задач следующие варианты:

- новый документ;
- XML- документ;
- веб-страница;
- сообщение электронной почты;
- из имеющегося документа;
- с использованием шаблонов на компьютере или на веб-узлах **Microsoft**.

После инсталляции **Microsoft Office Word 2003** имеет достаточно много шаблонов документов, которые сгруппированы по разделам:

- публикации (брошюра, диссертация, руководство, справочник);
- отчеты (изысканный, современный, стандартный);
- записки;
- письма и факсы;
- документы слияния;
- прочие.

Все шаблоны имеют собственные определенные стили оформления, которые при необходимости можно отредактировать. Можно изменить также вид самого шаблона и использовать его далее в своей работе над документами.

Значительный интерес для желающих научиться правильно работать со сложными документами представляют шаблоны системы **Microsoft Office 2003 Руководство, Брошюра** (правильнее назвать **Буклет**) и **Справочник** из раздела **Публикации**. Открыв эти шаблоны, можно познакомиться с основными инструкциями по использованию шаблонов и стилей в документах, оформлению буклета на одном листе бумаги, использованию символов в качестве рисунков, созданию оглавлений и алфавитных указателей.

Шаблоны раздела **Документы слияния** являются мастерами, которые помогают освоить оформление и рассылку писем или факсов по адресам, выбранным из таблиц существующих файлов или из адресной книги **Outlook**.

Шаблоны раздела **Отчеты** содержат краткую инструкцию по созданию оглавлений и указателей, маркированных и нумерованных списков, таблиц, колонтитулов, сносок.

В системе **Microsoft Office Word 2007** отсутствуют указанные выше шаблоны с краткими инструкциями по использованию стилей и других элементов оформления.

При оформлении документов может использоваться также компонент оформления документов – **Тема документа**. В системе **Office** до 2007 эта настройка присутствует в меню **Формат ► Тема**, в 2007 – на ленте **Разметка страницы**.

Тема системы Office - набор унифицированных элементов, определяющих внешний вид документа с помощью цвета, шрифтов и графических объектов.

Тема документа представляет собой набор вариантов форматирования, включающих цветовую тему (набор цветов), тему шрифтов (стили заголовков и основного текста), тему эффектов (набор линий и заливок), настройки списков, гиперссылок и границ таблиц.

Такие приложения, как **Microsoft Office Word, Excel** и **PowerPoint** имеют несколько заранее определенных тем документов, но пользователь может также создавать и свои темы путем настройки существующих тем документов, а затем сохранять их в виде пользовательских тем документов. Темы документов используются приложениями **Office 2007** совместно, так что все документы **Office** могут иметь единообразный вид.

Методика подготовки текстового документа

1. Задать параметры страницы

На ленте **Разметка страницы** (версии до 2007 - в меню **Файл ► Параметры страницы...**) следует задать для страницы размер, ориентацию и поля - левое, правое, верхнее и нижнее. При необходимости можно выбрать тему документа или задать количество колонок на листе, цвет страницы и текст подложки, границы страницы.

2. Определить параметры необходимых стилей

На ленте **Главная** открыть окно стилей (до 2007 - на панели **Форматирование** кнопка слева от раскрывающегося списка стилей открывает панель настройки стилей в области задач или меню **Формат ► Стили и форматирование...**). Вид окна стилей показан на рисунке 3.39.

Следует учитывать, что список стилей может быть неполный, в ссылке Параметры... можно выбрать режимы Рекомендованные, Используемые, В текущем документе, Все. (В версиях до 2007 список Все настраивался при выборе режима Показать: Специальное....)

Для выбранного стиля можно открыть список задач (см рисунок 3.39). Полные возможности по настройке стилей дает команда Изменить, диалоговое окно которой показано ранее на рисунке 3.37.

Кнопка Создать стиль позволяет определить параметры стиля с новым названием на основе текущего.

Для пользовательского формирования оглавления сложного документа (с использованием ленты Ссылки или в версиях до 2007 – пункта меню Вставка ► Ссылка ► Оглавление и указатели...) следует определить параметры стилей Заголовок 1 (уровень 1), Заголовок 2 (уровень 2), Заголовок 3 (уровень 3) и т. д., если необходимо.

Для основного текста следует определить параметры основного абзаца. При этом следует учитывать, что традиционный русский стиль – абзац с красной строкой (первая строка с отступом), без интервалов до и после абзаца, без отступов слева и справа. Стиль некоторых новых журналов, близкий к западным стилям – первая строка абзаца без отступа, но добавлен интервал после абзаца.

Не следует изменять стиль абзаца по умолчанию Обычный, который является основой стиля Обычная таблица, иначе в таблицах могут появиться нежелательные отступы первой строки, слева и справа; интервалы сверху и снизу в ячейках, увеличенный межстрочный интервал и пр.

Если удалить какой-либо стиль абзаца, то ко всем абзацам, которые были оформлены удаленным стилем будет применен стиль Обычный. Аналогично действует команда Очистить формат.

При определении внешнего вида документа может быть использована Тема документа. По умолчанию никакая Тема не используется, при задании темы параметры стилей документа будут переопределены.

Если в документе используется много таблиц, можно определить стиль таблицы, который будет использоваться по умолчанию (в меню Таблица ► Автоформат таблицы... или в области задач Стили и форматирование). После этого при создании таблицы с помощью кнопки Вставить таблицу панели инструментов Стандартная она будет иметь стиль таблицы по умолчанию.

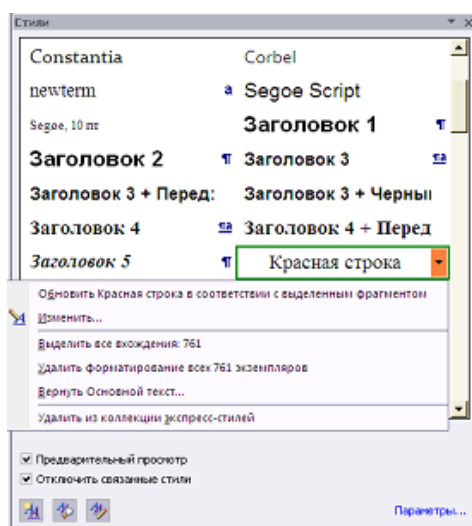


Рисунок 3.39. Окно «Стили»

Если необходимо, можно определить текст верхнего и нижнего колонтитулов. Для этого следует воспользоваться лентой Вставка или в версиях до 2007 командой меню Вид ► Колонтитулы. Кроме постоянного текста колонтитулы могут содержать поля, показывающие информацию текущего файла: номер страницы, автор, дата создания, имя файла, всего страниц и пр.

Колонтитулы первой страницы, четных и нечетных страниц могут различаться, что определяется на контекстной ленте Работа с колонтитулами или до версии 2007 в настройках Параметры страницы.

Каждый раздел документа может иметь собственный колонтитул. Это можно использовать для нумерации страниц альбомной ориентации, оформленных как отдельные разделы (например, для широких таблиц или рисунков) при книжной ориентации основных страниц. При альбомной ориентации номер страницы должен иметь иную ориентацию, чем при книжной. Например, если

обычная страница имеет автоматическую нумерацию в правом нижнем углу, альбомная страница того же документа (при печати на одной стороне бумаги) должна иметь номер в левом нижнем углу, развернутый по часовой стрелке на 90° (см. рисунок 3.40).

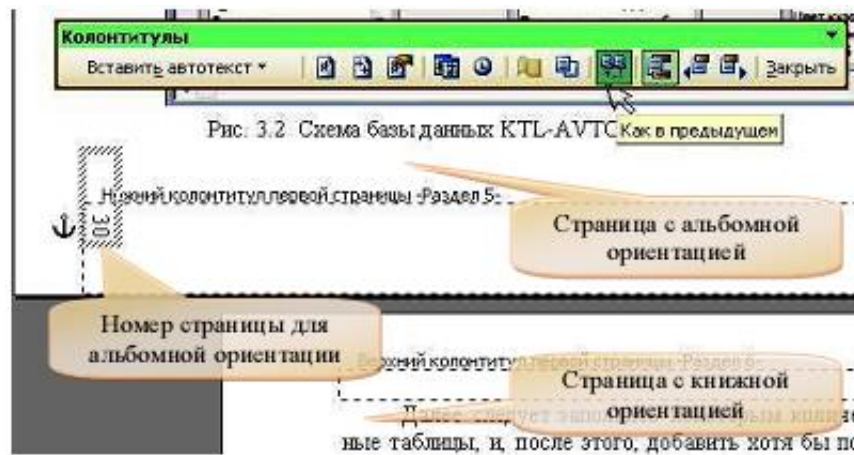


Рисунок 3.40. Номер страницы в колонтитуле для раздела с альбомной ориентацией в версии Word 2003

Это можно сделать, если для колонтитула нового раздела убрать свойство **Как в предыдущем** (кнопкой на панели инструментов **Колонтитулы**), после чего командой меню **Вставка ► Номера страниц...** вставить поля номеров страниц отдельно в разных разделах в нужном месте. Для колонтитулов альбомных страниц в меню (Word до 2007) командой **Формат ► Направление текста** задать направление сверху-вниз (см рисунок 3.40).

В версии **Word 2007** номер страницы командой ленты **Вставка – Номер страницы** размещается в колонтитуле без рамки. Поэтому для номеров с иной ориентацией, чем обычно, нужно сначала разместить в колонтитуле объект **Надпись**, потом в него (в текущее положение) вставить поле **Номер страницы**, далее можно развернуть номер в нужном направлении и убрать линии у рамки надписи (см. рисунок 3.41).

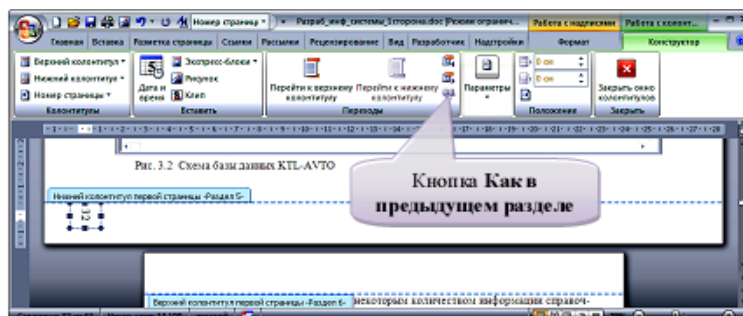


Рисунок 3.41. Работа с колонтитулами для раздела с альбомной ориентацией в версии Word 2007

4. Приступить к набору текста

Перед началом набора текста можно определить, будет система автоматически переносить слова по слогам при достижении границы строки в абзаце или нет (по умолчанию переносы отключены). Делается это в Word 2007 на вкладке **Разметка страницы – Расстановка переносов**, в Word 2003 – меню **Сервис – Язык – Расстановка переносов**.

Если документ начинается с титульного листа или обложки, на них не следует использовать стили заголовков 1 – 3 уровня, которые обычно входят в оглавление при его автоматическом формировании.

Основной текст документа обычно начинается с заголовка 1 уровня, стиль его лучше задать перед началом написания этого абзаца. В конце заголовков точек не ставят, в стилях заголовков следует

задать запрет автоматического переноса слов. Если заголовок должен быть написан прописными буквами, это следует задать в настройках стиля для раздела **Формат ► Шрифт**. Часто для заголовков задают отступ слева и справа 1-1,5 см, обычно интервал до абзаца и после, выравнивание может быть по центру или по левому краю. Стиль заголовков 2 и 3 уровня может быть основан на стиле Заголовок 1, но несколько модифицирован.

Для абзацев основного текста следует задать стиль с отступом первой строки или без него, но с интервалом после абзаца. В тексте могут присутствовать абзацы иного стиля, чем основной текст (например, аннотация в начале статьи или выделенный текст).

При необходимости можно использовать оформление текста в несколько колонок для всего документа или для одного раздела.

При вводе текста параметры абзаца могут быть изменены с использованием лент **Главная ► разделы Абзац, Шрифт и Разметка страницы ► раздел Абзац**.

Таблицы в тексте могут занимать всю ширину страницы или часть ее с обтеканием текстом или без обтекания. Ширина таблицы может быть задана в процентах от ширины страницы (без полей) или в сантиметрах (или других единицах, определенных в параметрах **Office**). Следует помнить, что в свойствах таблицы можно задать поля для ее ячеек (верхнее, нижнее, левое и правое), вертикальное выравнивание в ячейке, направление текста (слева-направо, снизу-вверх или сверху-вниз) и параметры абзацев в ячейках таблицы. Для многих операций при работе с таблицей удобно использовать контекстное меню, однако полный набор операций присутствует в двух контекстных лентах – **Конструктор** и **Макет** или в версиях до 2007 – в пункте меню **Таблица**.

Иногда таблицы без разлиновки используются для размещения структурированного текста на странице документа.

Рисунки могут быть добавлены в текст из файлов графических форматов или созданы с помощью команды ленты **Вставка ► Фигуры** или панели **Рисование Word 2003** (рисунки векторного типа, состоящий из отдельных объектов). При создании рисунков может использоваться полотно, которое позволяет выполнять операции со всеми объектами, размещенными в нем (перемещение всего рисунка, изменение масштаба, задание положения полотна). Положение рисунков может быть в тексте, перед ним или за ним, с обтеканием текстом вокруг рамки, контура или сверху и снизу.

В системе Word 2003 рисунок в формате *.bmp, скопированный из редактора Paint в буфер обмена, можно вставить в документ через пункт меню **Правка ► Специальная вставка** с заданием **Вставить как Bitmap Image** (объект), в этом случае рисунок можно будет редактировать прямо в документе, куда он вставлен (в редакторе Word запускается Paint).

Если рисунок или таблица имеют название, их можно написать в виде объекта **Рамка**, либо воспользоваться командой ленты **Вставка ► Надпись** или до версии Word 2007 - команда меню **Вставка ► Ссылка ► Название...** (этот режим позволяет также выполнять автоматическую вставку надписей с нумерацией таблиц и рисунков).

Несколько небольших рисунков могут быть размещены в ячейках нелинованной таблицы, при этом название рисунка или группы рисунков также можно разместить в отдельных ячейках таблицы (или в одной ячейке, объединенной из нескольких по горизонтали).

Если предполагается автоматическое формирование списка таблиц и рисунков с помощью команды ленты **Ссылки ► Названия ► кнопка Список иллюстраций** или до версии 2007 - командой меню **Вставка ► Ссылка ► Оглавление и указатели... ► Список иллюстраций**, не следует писать названия в графических объектах типа **Надпись** и пр., такие названия не попадут в сформированный список.

Формулы - математические, химические и другие могут быть вставлены в текст с использованием ленты **Вставка ► группа Символы ► команда Формулы**, или как объект **Microsoft Equation** (редактор формул) в Word 2003 и более ранних (а также и в Word 2007). В версии Word 2007 команда **Вставка ... Формулы** содержит достаточно много примеров написания формул:

✓ бином
Ньютона

$$(x + a)^n = \sum_{k=0}^n \binom{n}{k} x^k a^{n-k}$$

✓ квадратное
уравнение

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

✓ ряд Фурье

$$f(x) = a_0 + \sum_{n=1}^{\infty} \left(a_n \cos \frac{n\pi x}{L} + b_n \sin \frac{n\pi x}{L} \right)$$

Лента **Работа с формулами** ► **Конструктор** и объект **Microsoft Equation** содержат многочисленные шаблоны для написания различных математических символов, дробей, индексов, интегралов, сумм и пр. Пример формулы, написанной с помощью **Microsoft Equation**:

$$\frac{d\psi}{d\varphi} = -\sqrt{\frac{\cos \varphi_0 - \cos \varphi_1}{b(\operatorname{ctg}^2 \varphi_0 - \operatorname{ctg}^2 \varphi_1)}} \frac{1}{\sin^2 \varphi \sqrt{c - \cos \varphi}}$$

5. Вставить оглавление и указатели

Для сложного структурированного документа Word предоставляет полезные возможности формирования оглавления, списка таблиц и рисунков, предметного указателя.

Как указывалось выше, для автоматического формирования оглавления можно все заголовки, включаемые в оглавление, оформлять стилями соответствующего уровня. Если этого своевременно не сделано, в **Word 2003** можно задать уровни заголовков с помощью панели инструментов **Структура**, после чего сформировать оглавление (меню **Вставка** ► **Ссылка** ► **Оглавление и указатели**, вкладка окна **Оглавление**). Есть и другие способы создания оглавления, описанные в разделе справки «Создание оглавления».

В **Word 2007** оглавление формируется с помощью группы **Оглавление** на ленте **Ссылки**, которая позволяет сформировать автособираемое оглавление выбранного стиля, создать ручное оглавление выбранного стиля или задать уровни заголовков для абзацев (**Добавить текст** ► Уровень 1, 2, 3).

Для формирования списка таблиц и рисунков можно использовать оформление их особыми стилями, например **Название таблицы** и **Название рисунка**.

Методы создания предметного указателя описаны в разделе справки «Создание предметного указателя», один из способов – пометить слова предметного указателя сочетанием клавиш ALT+SHIFT+X, затем создать указатель в нужном месте командой ленты **Ссылки** ► **Предметный указатель** или в Word 2003 командой меню **Вставка** ► **Ссылка** ► **Оглавление и указатели**, вкладка окна **Указатель**.

6. Проверить правописание

В параметрах системы **Word** можно задать автоматическую проверку правописания и параметры проверки. Принудительно задать проверку можно в **Word 2007** – на ленте **Рецензирование** кнопкой **Правописание**, в **Word 2003** – в меню **Сервис** ► **Правописание**.

ЛЕКЦИЯ №2

Тема: Электронные таблицы

1.1 Назначение и основные возможности электронных таблиц	17
1.2 Режимы работы табличного процессора:	18
1.3 Объекты электронных таблиц MS Excel	18
1.4 Операции над объектами Microsoft Excel	20
1.5 Ввод и редактирование данных в ячейках	23
1.6 Формат представления данных в ячейках	25
1.7 Условное форматирование	26
1.8 Новые возможности Excel 2010	27

1.1 Назначение и основные возможности электронных таблиц

История

Идею электронных таблиц впервые сформулировал американский учёный австрийского происхождения Рихард Маттезих (нем. Richard Mattesich), опубликовав в 1961 г. исследование под названием «Budgeting Models and System Simulation»[3]. Концепцию дополнили в 1970 г. Пардо (англ. Rene Pardo) и Ландау (англ. Remy Landau), подавшие заявку на соответствующий патент (U.S. Patent 4 398 249). Патентное ведомство отклонило заявку, но авторы через суд добились отмены этого решения.

Общепризнанным родоначальником электронных таблиц как отдельного класса ПО является Дэн Бриклин, который совместно с Бобом Фрэнкстоном разработал программу VisiCalc в 1979 г. Эта электронная таблица для компьютера Apple II стала очень популярной, превратив персональный компьютер из игрушки для технофилов в массовый инструмент для бизнеса.

Впоследствии на рынке появились многочисленные продукты этого класса — SuperCalc, Microsoft MultiPlan, Quattro Pro, Lotus 1-2-3, Microsoft Excel, OpenOffice.org Calc, таблицы AppleWorks и gnumeric, минималистический Spread32.

Современные технологии обработки информации часто приводят к тому, что возникает необходимость представления данных в виде таблиц. В языках программирования для такого представления служат двумерные массивы. Для табличных расчетов характерны относительно простые формулы, по которым производятся вычисления, и большие объемы исходных данных. Такого рода расчеты принято относить к разряду рутинных работ, для их выполнения следует использовать компьютер. Для этих целей созданы электронные таблицы (табличные процессоры) — прикладное программное обеспечение общего назначения, предназначенное для обработки различных данных, представимых в табличной форме.

Электронная таблица (ЭТ) позволяет хранить в табличной форме большое количество исходных данных, результатов, а также связей (алгебраических или логических соотношений) между ними. При изменении исходных данных все результаты автоматически пересчитываются и заносятся в таблицу. Электронные таблицы не только автоматизируют расчеты, но и являются эффективным средством моделирования различных вариантов и ситуаций. Меняя значения исходных данных, можно следить за изменением получаемых результатов и из множества вариантов решения задачи выбрать наиболее приемлемый.

Применение электронных таблиц упрощает работу с данными и позволяет получать результаты без проведения расчетов вручную или специального программирования. Наиболее широкое применение электронные таблицы нашли в экономических и бухгалтерских расчетах, но и в научно-технических задачах электронные таблицы можно использовать эффективно.

Назначение электронных таблиц:

- проведения однотипных расчетов над большими наборами данных;
- автоматизации итоговых вычислений;
- решения задач путем подбора значений параметров, табулирования формул;
- обработки результатов экспериментов;
- проведения поиска оптимальных значений параметров;

- подготовки табличных документов;
- построения диаграмм и графиков по имеющимся данным.

1.2 Режимы работы табличного процессора:

- 1) формирование электронной таблицы;
- 2) управление вычислениями;
- 3) режим отображения формул;
- 4) графический режим;
- 5) режим базы данных.

1). Режим формирования электронных таблиц предполагает заполнение и редактирование документа. При работе с табличными процессорами создаются документы, которые можно просматривать, изменять, записывать на носители внешней памяти для хранения, распечатывать на принтере. При этом используются команды, изменяющие содержимое клеток (очистить, редактировать, копировать), и команды, изменяющие структуру таблицы (удалить, вставить, переместить).

2) Режим управления вычислениями. Все вычисления начинаются с ячейки, расположенной на пересечении первой строки и первого столбца электронной таблицы. Вычисления проводятся в естественном порядке, т.е. если в очередной ячейке находится формула, включающая адрес еще не вычисленной ячейки, то вычисления по этой формуле откладываются до тех пор, пока значение в ячейке, от которого зависит формула, не будет определено. При каждом вводе нового значения в ячейку документ пересчитывается заново, — выполняется автоматический пересчет. В большинстве табличных процессоров существует возможность установки ручного пересчета, т.е. таблица пересчитывается заново только при подаче специальной команды.

3) Режим отображения формул задает индикацию содержимого клеток на экране. Обычно этот режим выключен, и на экране отображаются значения, вычисленные на основании содержимого клеток.

4) Графический режим дает возможность отображать числовую информацию в графическом виде: диаграммы и графики. Это позволяет считать электронные таблицы полезным инструментом автоматизации инженерной, административной и научной деятельности.

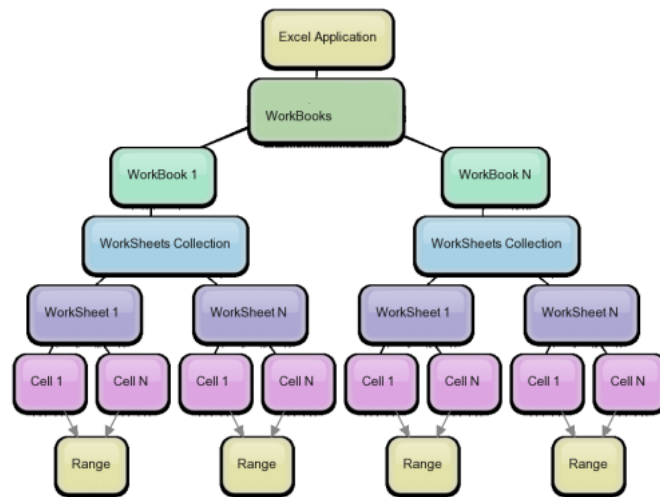
5) Режим базы данных.

В современных табличных процессорах, например, в Microsoft Excel, в качестве базы данных можно использовать список (набор строк таблицы, содержащий связанные данные). При выполнении обычных операций с данными, например, при поиске, сортировке или обработке данных, списки автоматически распознаются как базы данных. Перечисленные ниже элементы списков учитываются при организации данных:

- столбцы списков становятся полями базы данных;
- заголовки столбцов становятся именами полей базы данных;
- каждая строка списка преобразуется в запись данных.

1.3 Объекты электронных таблиц MS Excel

В табличном процессоре Excel 256 столбцов и более 16 тысяч строк.



Ячейка – минимальный объект табличного процессора;

Строка – горизонтальный набор ячеек, заголовки столбцов – A, B, C,...,IV;

Столбец – вертикальный набор ячеек, заголовки строк – 1, 2, 3,...,65536;

Адрес ячейки – определяется пересечением столбца и строки (A1, F123, AC72);

Указатель ячейки – рамка;

Активная ячейка – выделенная рамкой, с ней можно производить какие-либо операции;

Смежные ячейки – ячейки расположенные последовательно;

Диапазон (блок) ячеек (Range)– выделенные смежные ячейки, образующие прямоугольный участок таблицы;

Адрес диапазона (блока) ячеек - определяется адресом верхней левой и нижней правой ячейки, разделенных двоеточием (:), B2:C7 → B2, B3, B4, B5, B6, B7, C2, C3, C4, C5, C6, C7.

Ссылка в Microsoft Excel — адрес ячейки или связанного диапазона ячеек. В каждом листе Excel может быть 256 (2 в 8 степени) столбцов и 65536 (2 в 16 степени) строк (В Excel 2007 16384 (2 в 14 степени) столбцов и 1048576 (2 в 20 степени) строк). Адрес ячейки определяется пересечением столбца и строки, как в шахматах или морском бое, например: A1, C16. Адрес диапазона ячеек задается адресом верхней левой ячейки и нижней правой, например: A1:C5.

Виды представления ссылок

1. Классический;

2. Стилль ссылок R1C1 (здесь R — row (строка), C — column (столбец)).

Включить стилль ссылок R1C1 можно в настройках «Сервис» → «Параметры» → закладка «Общие» → галочка «Стилль ссылок R1C1»

Если включен классический вид и в ячейке с адресом A1 находится формула «=B3» (B3 — относительная ссылка), то после переключения в вид R1C1 она примет вид «=R[2]C[1]». В режиме R1C1 в квадратных скобках указывается относительное смещение. В скобках после R указано смещение в строках (row) — две строки вниз (вниз, потому что число положительное, если бы стояло «R[-2]» — было бы вверх). В скобках после C указано смещение в столбцах — 1 столбец вправо (если бы было C[-1] — влево). В общем случае направления вниз и вправо считаются положительными, вверх и влево — отрицательными.

Если включен классический вид и в ячейке с адресом A1 находится формула «=\$B\$3» (\$B\$3 — абсолютная ссылка), то после переключения в вид R1C1 она примет вид «=R3C2». Квадратных скобок нет, то есть указано не относительно смещение по отношению к положению формула, а абсолютное смещение по отношению к всему листу (вспомните абсолютную и относительную системы координат).

Типы ссылок (типы адресации)

Ссылки в Excel бывают 3-х типов:

1. Относительные ссылки (пример: A1);

2. Абсолютные ссылки (пример: \$A\$1);

3. Смешанные ссылки (пример: \$A1 или A\$1, они наполовину относительные, наполовину абсолютные).

Знак \$ здесь никакого отношения к денежным единицам не имеет, это лишь способ указать Excel тип ссылки. Различия между разными типами ссылок можно увидеть, если потянуть за маркер автозаполнения активной ячейки или диапазона ячеек, содержащих формулу со ссылками.

Относительные ссылки

Если вы ставите в какой то ячейке знак "=", затем щелкаете левой кнопкой мыши на какой то ячейке, Excel подставляет после "=" относительную ссылку на эту ячейку. Эта ссылка "запоминает", на каком расстоянии (в строках и столбцах) Вы щелкнули ОТНОСИТЕЛЬНО положения ячейки, где поставили "=" (смещение в строках и столбцах). Например, вы щелкнули на ячейку 3-мя столбцами левее и на 2 строки выше. Если после нажатия Enter потянуть вниз за маркер автозаполнения, эта формула скопируется во все ячейки, через которые мы протянули. И в каждой ячейке эта ссылка будет указывать на ячейку, расположенную на 3 столбца влево и 2 строки вверх ОТНОСИТЕЛЬНО положения ссылки. Это можно проверить, дважды щелкнув на одной из скопированных формул, или выделив ее, и нажав F2. Для лучшего понимания вспомните, как ходит шахматный конь. Он ходит буквой "Г" и из центра доски бьет 8 клеток. «Упростим» немного правило хода коня: представим, что он может ходить только одной буквой «Г» — 2 клетки вперед и одну вправо. На какую бы клетку доски мы не поставим коня, каждый раз он ОТНОСИТЕЛЬНО своего положения отсчитывает смещение в строках и столбцах — 2 строки вверх и один столбец влево. Точно таким же образом работают относительные ссылки, только правило их «хода» задает пользователь. Каждый раз, когда мы тянем за маркер автозаполнения формула, содержащую относительные ссылки, Excel пересчитывает адреса всех относительных ссылок в ней в соответствии с их «правилом хода» (у каждой относительной ссылки в формуле может быть свое «правило»).

Абсолютные ссылки

Как было сказано выше, если потянуть за маркер автозаполнения формулу, содержащую относительные ссылки, Excel пересчитает их адреса. Если же в формуле присутствуют абсолютные ссылки, их адрес останется неизменным. Проще говоря — абсолютная ссылка всегда указывает на одну и ту же ячейку.

Чтобы сделать относительную ссылку абсолютной, достаточно поставить знак «\$» перед буквой столбца и адресом строки, например \$A\$1. Более быстрый способ — выделить относительную ссылку и нажать один раз клавишу «F4», при этом Excel сам проставит знак «\$». Если второй раз нажать «F4», ссылка станет смешанной такого типа A\$1, если третий раз — такого \$A1, если в четвертый раз — ссылка опять станет относительной. И так по кругу.

Смешанные ссылки

Смешанные ссылки являются наполовину абсолютными и наполовину относительными. Знак доллара в них стоит или перед буквой столбца или перед номером строки. Это самый сложный для понимания тип ссылки. Например, в ячейке записана формула «=A\$1». Ссылка A\$1 относительная по столбцу A и абсолютная по строке 1. Если мы потянем за маркер автозаполнения эту формулу вниз или вверх, то ссылки во всех скопированных формулах будут указывать на ячейку A1, то есть будет вести себя как абсолютные. Однако, если потянем вправо или влево — ссылки будут вести себя как относительные, то есть Excel будет пересчитывать ее адрес. Таким образом, формулы, созданные автозаполнением, будут использовать один и тот же номер строки (\$1), но изменится номер столбца (A, B, C...).

Книга (Workbook)– документ электронной таблицы, состоящий из листов, объединенных одним именем и являющихся файлом;

Лист (Worksheet, Sheet) – рабочее поле, состоящее из ячеек.

1.4 Операции над объектами Microsoft Excel

Формы курсора мыши

курсор ожидания загрузки программы (песочные часы);
 курсор выделения ячейки или группы ячеек ("санитарный крест");
 курсор автозаполнения (крест);
 курсор перемещения (стрелка, наклоненная влево);
 курсор редактирования (I);
 курсор выделения строк/столбцов;
 курсор изменения размера строк/столбцов;

Операции над листами (ярлычками) рабочей книги

С использованием левой кнопки мыши:

щелчок левой кнопкой по закладке (ярлычку) — переключение на соответствующее окно (либо использовать сочетание клавиш Ctrl + PageUp/Ctrl + PageDown);
 перетаскивание левой кнопкой мыши — изменение порядка расположения листов;
 Ctrl + перетаскивание левой кнопкой мыши — создание копии листа;
 Ctrl + последовательные щелчки по ярлычкам — выделение нескольких листов;
 Shift + щелчок на начальном и конечном ярлычке — выделение диапазона листов;
 двойной щелчок по ярлычку — переименование листа. "Enter" для завершения.
 По нажатию правой кнопки мыши над ярлычком открывается **контекстное меню**, в котором доступны следующие команды:

- **"Добавить..."**. Открывается окно, в котором выбираем, что добавить: новый лист, диаграмму, готовый шаблон и т.д.;
- **"Удалить"**. Надеюсь, без комментариев;
- **"Переименовать"**. Надеюсь, без комментариев;
- **"Переместить/Скопировать..."**.
 - в выпадающем списке "в книгу:" выбираем, куда мы копируем лист: в один из открытых документов или в новую книгу;
 - В списке "перед листом:" будут отображены листы выбранной книги. Выбираем, переместить лист в конец или перед каким листом вставить;
 - Если не устанавливать галочку "Создавать копию", лист будет **перемещен** в выбранную книгу (т. е. в этой его не останется!), если установить — в выбранной книге будет создана копия этого листа.
- **"Выделить все листы"**.
- **"Цвет ярлычка..."**. Тоже вроде без комментариев. Разве что хочу отметить, что активный "цветной" ярлычок отображается "некорректно" (ИМНО);
- **"Исходный текст"**. "Господа программисты" разберутся, для чего эта команда, обычным же пользователям сюда не ходить.

Перемещение по таблице (активация нужной ячейки)

Самый простой способ — щелкнуть левой кнопкой на нужной ячейке.

С помощью клавиш управления курсором (стрелочек) можно переместить "курсор" (рамку) на ячейку вверх/вниз/вправо/влево.

Нажатие клавиши **"Tab"** приводит к перемещению в **следующую** (справа) ячейку в строке, а **"Shift+Tab"** — в предыдущую. Нажатие клавиши **"Enter"** приводит к перемещению на ячейку **вниз**, а **"Shift+Enter"** — **вверх**.

Нажатие сочетаний клавиш **"Ctrl + клавиши управления курсором"** приводит к "прыжку" курсора к ближайшей ячейке, где имеются какие-то данные. Если же таких ячеек на пути не встречается, курсор прыгает к одному из краёв таблицы.

Если выделен **диапазон ячеек**, одна из них все равно остается белой, "невыделенной" (см. выше). При этом нажатие клавиш "Tab", "Enter" [+Shift] приведет к циклическому перемещению

активной ячейки. В процессе этого можно вводить данные, выделение не снимется. Если в процессе перемещения активной ячейки нажать **"Ctrl+Enter"** (не выходя из режима редактирования), значение из этой ячейки **скопируется** во **все** ячейки выделенного диапазона;

Нажатие клавиш **"Page Up"** / **"Page Down"** приводит к перемещению **на страницу** вниз / вверх;

Выделение ячеек, диапазонов ячеек, столбцов и строк

Чтобы выделить **одну** ячейку, щелкнете на ней **левой** кнопкой мыши;

Чтобы выделить **связный диапазон** ячеек, наведите курсор мыши на ячейку, которая будет началом/концом выделенного диапазона, зажмите левую кнопку мыши, и, **не отпуская ее**, двигайте мышь до той ячейки, которая будет концом/началом выделенного диапазона, затем отпустите левую кнопку мыши;

Если нужно выделить два и более **несвязных** диапазона/ячейки, выделите сначала первый/первую, затем зажмите **"Ctrl"**, и не отпуская его, выделите остальные диапазоны/ячейки. Они добавятся в выделение;

Нажатие сочетания клавиш **"Ctrl+A"** приводит к выделению **всей таблицы**. Но не всегда, есть маленький **нюанс**. Если справа, справа и на одну строку вниз, или снизу есть ячейка, содержащая данные, после первого нажатия этого сочетания выделится диапазон, начальной ячейкой которого будет активная ячейка, конечной — ячейка с данными. Повторное нажатие **"Ctrl+A"** приведет к выделению **всей** таблицы;

Чтобы выделить **всю таблицу**, можно щелкнуть левой кнопкой мыши **на пересечении столбцов и строк** ("между" столбцом "А" и строкой "1");

Чтобы выделить **весь столбец**, щелкнете на **заголовке** столбца ("шапка" таблицы сверху);

Чтобы выделить **всю строку**, щелкнете на **заголовке** строки ("шапка" таблицы слева);

Чтобы выделить **диапазон столбцов или строк**, наведите курсор мыши на начало диапазона, зажмите левую кнопку, и, не отпуская ее, передвигайте курсор;

С зажатой клавишей **"Ctrl"** можно выделять не только несвязные ячейки и диапазоны, но и **строки и столбцы**;

Если зажать клавишу **"Shift"** и нажимать на **клавиши управления курсором**, выделяется диапазон, началом которого будет активная в момент нажатия клавиш управления курсором ячейка, а концом ячейка с координатами, зависящими от того, сколько раз и какие вы нажали клавиши управления курсором.

Изменение ширины столбцов и высоты строк

При наведении курсора мыши на границу между двумя строками или столбцами (в "шапке" таблицы), курсор изменяет форму. Если зажать левую кнопку мыши и, не отпуская ее, переместить курсор, будет изменяться ширина столбца или высота строки. При этом появится всплывающая подсказка, в которой будет отображаться числовое значение ширины (высоты).

При изменении **ширины столбцов**, первое число будет означает, **сколько символов** шрифта текущего размера поместится в этой ячейке. Число в скобках показывает ширину в **пикселях**.

При изменении **высоты строк**, первое число показывает, какого максимального **размера (шрифта)** текст поместится в эту ячейку. Число в скобках показывает ширину в **пикселях**.

Если **выделить несколько столбцов (строк)**, затем изменить ширину (высоту) одной из них, все выделенные столбцы (строки) станут такой же ширины (высоты).

Если выполнить **двойной щелчок** левой кнопкой мыши на **правой границе** столбца (**нижней границе** строки), то ширина столбца (высота строки) **подгонится под содержимое** (под самый "длинный" ("высокий") элемент. Аналогично, если выделены **несколько столбцов (строк)**, то двойной щелчок на правой (нижней) границе **любого** из них подгонит **все выделенные** столбцы (строки) под содержимое;

Если щелкнуть правой кнопкой на заголовке столбца (строки), в контекстном меню появятся команды **"Ширина столбца..."** (**"Высота строки"**). После выбора одной из них откроется окно "Ширина столбца" ("Высота строки"), в котором пользователь может ввести **числовое значение** ширины (высоты) в количестве символов (размер шрифта).

1.5 Ввод и редактирование данных в ячейках

Для ввода данных в одну ячейку достаточно сделать ее активной и начать набирать текст, при этом мы входим в режим редактирования данных в ячейке. Для выхода из режима редактирования нужно нажать "Enter" или "Tab" ([+Shift]). Если до этого в ячейке были данные, они сотрутся;

Для ввода одинаковых данных во многие ячейки, выделяем нужные ячейки, пишем текст, и, не выходя из режима редактирования, нажимаем "Ctrl+Enter";

Для изменения данных в ячейке, делаем ее активной и делаем на ней двойной щелчок;

Для изменения данных в ячейке, делаем ее активной и нажимаем "F2";

Для изменения данных в ячейке, делаем ее активной, щелкаем в строке формул, и в ней изменяем содержимое ячейки, для завершения нажимаем "Enter", либо зеленую галочку слева.

Для принудительного ввода разрыва строки достаточно в нужном месте текста ячейки нажать "Alt+Enter"

Форматирование ячеек

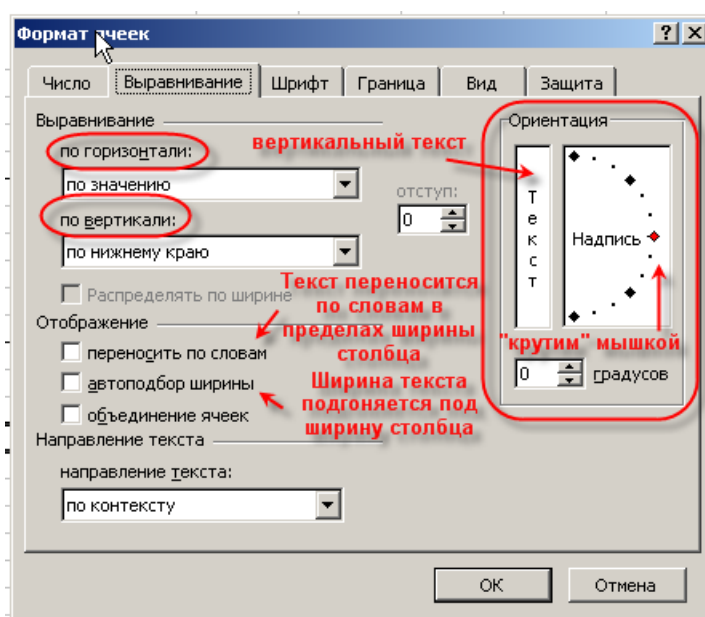
Выравнивание данных в ячейках

По умолчанию, текст выравнивается по левому краю (по горизонтали), а числа — по правому (по горизонтали). По вертикали данные выровнены по нижнему краю. **Выравнивание по горизонтали** можно поменять с помощью кнопок на панели инструментов "Форматирование" которую можно вызвать сочетанием кнопок **ctrl+1**:



Здесь (слева направо) кнопки: по левому краю, по центру и по правому краю.

Выравнивание по вертикали изменяется с помощью окна "Формат ячеек", закладка "Выравнивание":



Объединение ячеек

Крайне часто возникает необходимость создать одну большую ячейку, которая находится на пересечении нескольких столбцов и нескольких строк:

Пример объединенных ячеек				
Зима			Зима	Декабрь
Декабрь	Январь	Февраль		Январь
				Февраль

Это можно сделать двумя способами:

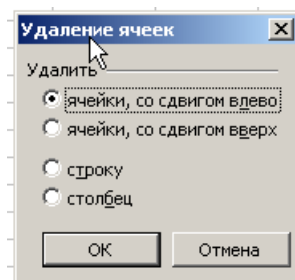
кнопка "Объединить и поместить в центре" на панели инструментов "Форматирование" ; **галочка "объединение ячеек"** в окне "Формат ячеек", закладка "Выравнивание".

Сначала выделяем ячейки, которые хотим объединить (можно выделять несвязные диапазоны), затем выполняем одно из вышеуказанных действий. Если в выделенном диапазоне была (одна) ячейка с текстом, текст выравнивается по центру объединенной ячейки (только по горизонтали). Если в выделенном диапазоне было несколько ячеек с текстом, то сохранится только значение, которое было верхним левым и тоже выравнивается.

При активной объединенной ячейке на панели инструментов "Форматирование" будет вдавнена кнопка "Объединить и поместить в центре". Если при этом нажать ее, объединенная ячейка опять распадется на "родные" маленькие, а значение перенесется в верхнюю левую ячейку.

Адресом объединенной ячейки станет адрес верхней левой ячейки, остальные ячейки выделенного диапазона Excel будет рассматривать как пустые.

Очистка данных в ячейках и удаление ячеек

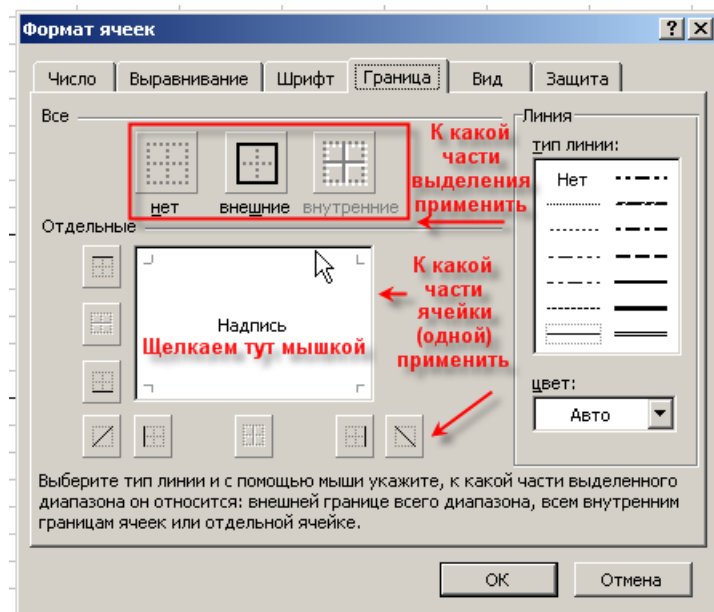


- Выделяем "нужные" ячейки и нажимаем "Delete";
- Выделяем "нужные" ячейки, щелкаем правой кнопкой и выбираем "Очистить содержимое";
- Выделяем "нужные" ячейки, заходим в меню "Правка" → "Очистить" → "Выбираем нужный вариант";
- Клавиша ← очищает содержимое **активной** (одной!) ячейки и включает режим редактирования. Ее удобно использовать при необходимости удалить одно (не все, как по "Delete") значение из выделенного диапазона;
- Если "родную" неотформатированную ячейку протащить с помощью маркера автозаполнения по отформатированному, произойдет своего рода удаление;
- пункт меню "Правка" → "Удалить..." и пункт контекстного меню "Удалить..." приведет к открытию окна "Удаление ячеек".

Добавление границ ячеек

Лист Excel по умолчанию представляет из себя таблицу. Однако сетка таблицы не выводится на печать, пока мы их не наведем. Существует три способа добавить границы к выделенным ячейкам:

Кнопка "Границы" на панели инструментов ; окно "Формат ячеек", закладка "Граница".



1.6 Формат представления данных в ячейках

1. Формат "Общий" (по умолчанию)

По умолчанию после создания документа все ячейки находятся в формате "Общий". Этот формат имеет ряд хитростей:

числа выравниваются по правому краю, а текст — по левому;

если, изменяя ширину столбца, сделать ее меньше определенной, то число в ячейках заменяется на символы "#". Это не ошибка. Это означает, что нужно сделать столбец пошире;

если число очень большое ("6000000000000") или очень маленькое ("0,00000000000001"), оно автоматически преобразуется в экспоненциальный (научный) формат ("6E+12" и "1E-14" соответственно);

при изменении ширины столбца округляются десятичные дроби. Например, если написать "3,1415", затем изменить ширину так, чтобы "5" перестала помещаться, в ячейке отобразится "3,142".

2. Формат "Числовой"

можно настроить число десятичных знаков, автоматически добавляемых после запятой;

можно настроить отделять или не отделять пробелом триады разрядов (сотни, сотни тысяч и т.п.);

можно настроить формат отображения отрицательных чисел:

черные с минусом;

красные без минуса;

черные с минусом и отступом (справа);

красные с минусом и отступом.

Внимание! Несмотря на то, что при изменении количества отображаемых знаков после запятой визуально число изменяется, в ячейке по-прежнему хранится неокругленное число и операции будут осуществляться именно с ним. К примеру, пусть в ячейку было занесено число 1,23456. Пользователь оставил лишь 4 знака после запятой. В ячейке будет отображаться число 1,2346. Но при умножении на 100 000 в ячейке будет результат 123 456, а не 123 460! При расчетах пользуйтесь функцией ОКРУГЛ, если есть необходимость работать с округленным числом.

3. Формат "Денежный"

Те же настройки, что и в "Числовой" (кроме отключения разделения разрядов), плюс выбор денежной единицы. По умолчанию отображается денежная единица, указанная в настройках "Панель управления" —> "Язык и региональные стандарты" —> закладка "Региональные параметры".

4. Формат "Финансовый"

5. Формат "Дата"

Обычно сами даты вводятся в ячейки вручную (см. Автоматическое изменение формата ячейки после ввода данных), а затем в окне "Формат ячеек" выбирается нужный вид представления даты. Также можно сменить текущий формат представления даты на формат, принятый в других странах.

6. Формат "Время"

Обычно время вводятся в ячейки вручную (см. Автоматическое изменение формата ячейки после ввода данных), а затем в окне "Формат ячеек" выбирается нужный вид представления времени. Также можно сменить текущий формат представления времени на формат, принятый в других странах.

7. Формат "Процентный"

В формате "Процентный" все числа умножаются на 100 и добавляется знак "%". Также можно настроить количество десятичных разрядов после запятой.

8. Формат "Дробный"

В формате "Дробный" Excel пытается преобразовать десятичные дроби в обыкновенные. Например, "1,2" (одна целая, две десятых) будет преобразовано в "1 1/5" (одна целая, одна пятая).

9. Формат "Экспоненциальный"

Формат "Экспоненциальный" удобен для представления очень больших (расстояние до Солнца в метрах) или очень маленьких (масса атома водорода в килограммах) чисел. Например, число "299 792 458" (скорость света в метрах) в этом формате преобразуется в "3,E+08". Знак "+" здесь означает, что запятую нужно передвинуть **вправо**, а "08" — на какое количество разрядов. Также можно настроить количество знаков после запятой.

10. Формат "Текстовый"

Значения в ячейках, оформленных в этом формате, отображаются точно так же, как вводятся. Они обрабатываются как строки вне зависимости от их содержания. Например, если в ячейке, оформленной в формате "Текстовой", написать "1.2.3", Excel не будет пытаться преобразовать это в дату.

Ввод в ячейку 1-го символа " ' " (клавиша "Э" русской раскладки) автоматически приводит содержимое ячейки к текстовому формату. Символ " ' " на экране не отображается.

11. Формат "Дополнительный"

В выпадающем списке "Язык" выберите "Русский". В списке "Тип" появятся следующие варианты: "Почтовый индекс", "Индекс + 4", "Номер телефона", "Табельный номер". Попробуйте оформить ячейку типом "Номер телефона" и введите туда 10-значный номер. Думаю объяснять не надо. Для других языков могут отобразиться другие варианты.

12. Пункт "(все форматы)"

В этом пункте можно создать свой формат (например, "шт", "кг.", "м/с" и т.п.), отсутствующий среди стандартных. Для этого используется специальный язык описания форматов. В текстовом поле сверху описываете его и нажимаете "Enter". Для удаления формата выделяете "нужный" формат и нажимаете кнопку "Удалить". Удалить встроенный формат невозможно. (Совет: форматы хранятся в рабочей книге; удаление ненужных форматов удобно завершать Отменой, чтоб не "испортить" формат текущей ячейки.)

1.7 Условное форматирование

Условное форматирование обеспечивает наглядность при исследовании и анализе данных, обнаружении критических проблем, шаблонов и тенденций.

Условное форматирование помогает давать визуальные ответы на определенные вопросы о данных. Условное форматирование можно применить к диапазону ячеек, таблице Microsoft Excel или отчету сводной таблицы. При использовании условного форматирования в отчете сводной таблицы необходимо учитывать ряд важных моментов.

Преимущества условного форматирования

При анализе данных часто возникают указанные ниже вопросы.

- Каковы исключения в сводных доходах за последние пять лет?
- Каковы рыночные тенденции по результатам опроса общественного мнения за два последних года?

- Кто в этом месяце продал товаров на сумму более 50 000 рублей?
- Каково общее распределение сотрудников по возрасту?
- Для каких продуктов рост дохода составил более 10% в год?
- Кто из студентов является лучшим и худшим на младшем курсе?

Условное форматирование помогает ответить на эти вопросы, упрощая выделение интересующих пользователя ячеек или диапазона ячеек, выделение нетипичных значений и визуализацию данных с помощью гистограмм, цветовых шкал и наборов значков. Условный формат изменяет вид диапазона ячеек на основе условий (или критериев). Если условие истинно, диапазон ячеек форматируется на основе этого условия; если условие ложно, диапазон ячеек не форматируется.

Как при ручном, так и при условном форматировании можно выполнять сортировку и фильтрацию ячеек по формату, включая цвет ячейки и шрифта.

Имеется возможность указать Excel, чтобы при вводе данных, соответствующих определенному условию, он изменял оформление ячейки: заливку, границы, шрифт.

1. Форматирование всех ячеек с использованием двухцветной шкалы
2. Форматирование всех ячеек с использованием трехцветной шкалы
3. Форматирование всех ячеек с использованием гистограммы
4. Форматирование всех ячеек с использованием набора значков
5. Форматирование ячеек с текстовыми, числовыми значениями или значениями даты или времени

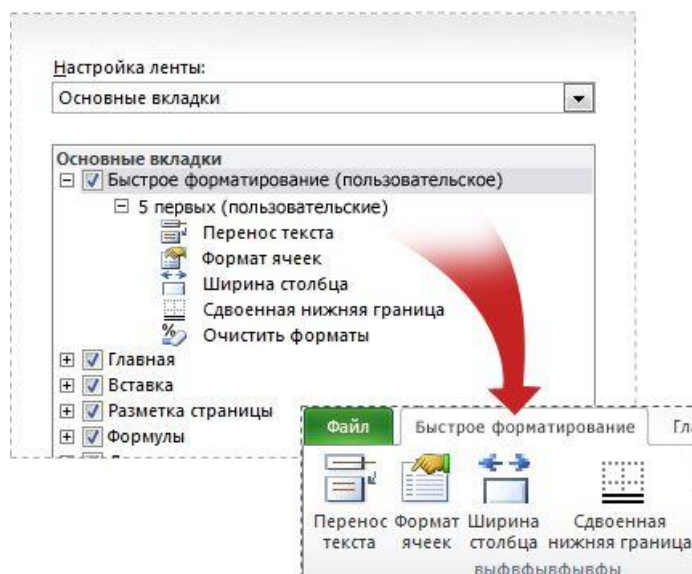
1.8 Новые возможности Excel 2010

Нужные инструменты всегда под рукой

Новые и улучшенные функции позволяют существенно повысить производительность, но только если они всегда доступны в нужный момент. Как и в других программах пакета Microsoft Office 2010, в приложении Excel 2010 присутствует интерфейс Microsoft Office Fluent — настраиваемая визуальная система инструментов и команд.

Улучшенная лента

Лента, впервые представленная в версии Excel 2007, упрощает поиск команд и функций, которые раньше были скрыты в сложных меню и панелях инструментов. В приложении Excel 2007 можно было настроить панель быстрого доступа, однако не было возможности создавать собственные вкладки и группы на ленте. В Excel 2010 можно создавать настраиваемые вкладки и группы, а также переименовывать и переупорядочивать встроенные.



Представление Microsoft Office Backstage

Перейдите на вкладку Файл, чтобы открыть представление Backstage, где можно создавать новые файлы и открывать существующие, сохранять их, отправлять, защищать, предварительно просматривать и выводить на печать, а также настраивать параметры Excel и выполнять многие другие действия.

Инструменты для работы с книгами

В приложении Excel 2010 предусмотрены инструменты для управления содержимым, его защиты и совместного использования.

1. Восстановление предыдущих версий документов Теперь можно восстановить версии файлов, закрытые без сохранения изменений. Это помогает в ситуациях, когда пользователь забыл сохранить файл вручную, случайно были сохранены ненужные изменения или просто возникла необходимость вернуться к одной из предыдущих версий книги. Ознакомьтесь с дополнительными сведениями о восстановлении файлов.

2. Защищенный просмотр В приложении Excel 2010 предусмотрен режим защищенного просмотра, позволяющий принять обоснованное решение, прежде чем подвергнуть компьютер возможным уязвимостям. По умолчанию документы, полученные из Интернета, открываются в режиме защищенного просмотра. В этом случае на панели сообщений появляется предупреждение, а также команда, позволяющая включить возможность редактирования. Можно настраивать список источников, документы из которых открываются в режиме защищенного просмотра. Кроме того, можно сделать так, что файлы определенных типов будут открываться в этом режиме независимо от своего происхождения. Ознакомьтесь с дополнительными сведениями о защищенном просмотре.

3. Надежные документы Функция подтверждения надежности документов предназначена для упрощения открытия книг и других документов, содержащих активное содержимое, например подключения к данным или макросы. Теперь достаточно подтвердить один раз, что активное содержимое конкретной книги является безопасным, чтобы в будущем не повторять эту операцию. В приложении Excel 2010 сохраняются сведения о том, какие книги отнесены к надежным, поэтому больше не потребуется подтверждать их статус каждый раз при открытии. Ознакомьтесь с дополнительными сведениями о надежных документах.

Новые способы доступа к книгам

Теперь обращаться к файлам и работать с ними можно из любого местоположения: с работы, из дома или находясь в пути.

Microsoft Excel Web App

Excel Web App позволяет использовать возможности Excel в веб-браузере и работать с книгами непосредственно на сайте, на котором они хранятся. Excel Web App — это одно из веб-приложений Microsoft Office Web Apps, доступное в Windows Live SkyDrive и в организациях, в которых веб-приложения Office Web Apps настроены в среде SharePoint 2010.

Ниже перечислены возможности, доступные благодаря веб-приложению Excel Web App.

1. Просмотр книги в браузере После того как пользователь щелкнет книгу, чтобы открыть ее в Excel Web App, книга будет отображена в режиме просмотра. В этом режиме можно сортировать и фильтровать содержащиеся в ней данные, разворачивать сводные таблицы, отслеживая зависимости и тренды в данных, пересчитывать значения и просматривать различные листы.

2. Редактирование книги в браузере Для работы с книгами в Excel Web App необходим только браузер. При этом можно работать совместно с коллегами независимо от того, какую версию приложения Excel они используют. Когда пользователь щелкает книгу Excel, хранящуюся на сайте SharePoint или в SkyDrive, она открывается непосредственно в веб-браузере и выглядит точно так же, как в приложении Excel. При этом листы в веб-браузере можно редактировать, используя знакомый интерфейс Excel. Работая в браузере, можно изменять данные, вводить и редактировать формулы, а также применять к электронной таблице основное форматирование. Кроме того, над книгой можно работать одновременно с другими пользователями.

Ознакомьтесь с дополнительными сведениями о приложении Excel Web App.

Быстрое и эффективное сравнение списков данных

В приложении Excel 2010 новые компоненты (такие как спарклайны и срезы), улучшения сводных таблиц и других существующих функций помогают выявлять закономерности и тренды в данных.

Спарклайны

Спарклайны – маленькие диаграммы, помещающиеся в одну ячейку, — позволяют визуально отображать тренды непосредственно рядом с данными. Поскольку спарклайны показывают тренды на ограниченном пространстве, с их помощью удобно создавать панели мониторинга и другие аналогичные компоненты, демонстрирующие текущее состояние дел в понятном и наглядном виде. На рисунке ниже показаны спарклайны в столбце "Тренд", позволяющие моментально оценить деятельность каждого из отделов в мае.

Показатели продаж по отделам за май 2010 г.
(объем продаж в миллионах)

	Текущий	Тенденция	За год	
			Верхний	Нижний
Все отделы	159,6 руб.		159,6 руб.	130,5 руб.
Женская одежда	32,6 руб.		32,6 руб.	30,2 руб.
Товары для дома и сада	16,2 руб.		16,9 руб.	15,1 руб.
Игрушки	15,7 руб.		15,7 руб.	13,5 руб.
Электроника	14,9 руб.		18,3 руб.	14,9 руб.
Детская одежда	14,2 руб.		14,8 руб.	7,5 руб.
Мужская одежда	13,5 руб.		13,2 руб.	12,8 руб.
Детская одежда	12,6 руб.		12,9 руб.	11,9 руб.
Мебель	11,4 руб.		14,5 руб.	8,7 руб.
Товары для спальни и ванной	11,1 руб.		11,1 руб.	9,3 руб.
Товары для кухни	9,9 руб.		10,1 руб.	2,3 руб.
Товары для дома	7,5 руб.		9,1 руб.	4,5 руб.

Усовершенствованные сводные таблицы

Сводные таблицы стали еще производительнее и удобнее в использовании. Ниже перечислены основные усовершенствования.

Повышенная производительность Многопоточность в приложении Excel 2010 позволяет ускорить извлечение данных, сортировку и фильтрацию в сводных таблицах.

Подписи в сводных таблицах В сводных таблицах теперь можно заполнять подписи. С помощью повторения подписей в сводных таблицах также можно отображать названия элементов для вложенных полей во всех строках и столбцах. [Просмотрите видеоролик о повторении подписей элементов.](#)

Расширенные возможности фильтрации С помощью срезов можно быстро фильтровать данные в сводной таблице одним нажатием кнопки и просматривать примененные фильтры без необходимости открывать дополнительные меню. Кроме того, в интерфейсе фильтра есть полезное поле поиска, с помощью которого можно найти нужные данные среди тысяч (или даже миллионов) элементов сводных таблиц.

Поддержка обратной записи В приложении Excel 2010 можно изменять значения в области значений сводной таблицы OLAP и выполнять их обратную запись в куб служб аналитики на сервере OLAP. Функцию обратной записи можно использовать в режиме анализа "что если", а затем выполнять откат изменений, когда они становятся не нужны, или сохранять их. Обратную запись можно применять с любыми поставщиками OLAP, поддерживающими оператор UPDATE CUBE.

Функция "Дополнительные вычисления" Функция **Дополнительные вычисления** позволяет выполнять ряд новых автоматизированных вычислений, например % от суммы по родительской строке, % от суммы по родительскому столбцу, % от родительской суммы, % от суммы с нарастающим итогом, Сортировка от минимального к максимальному и Сортировка от максимального к минимальному. [Просмотрите видеоролик об изменениях в функции "Дополнительные вычисления".](#)

Улучшения сводных диаграмм Теперь стало еще проще работать с отчетами сводных диаграмм, а именно фильтровать данные непосредственно в сводной диаграмме и вносить изменения в макет сводной диаграммы путем добавления и удаления полей. Кроме того, одним щелчком можно скрыть все

кнопки полей на отчете сводной диаграммы. [Просмотрите видеоролик об использовании интерактивных элементов управления в отчете сводной диаграммы.](#)

Первый интерфейс сводных таблиц, называемых также сводными отчеты, был включен в состав Excel еще в 1993м году (версии Excel 5.0). Несмотря на множество полезных функциональных возможностей, он практически не применяется в работе большинством пользователей Excel. Даже опытные пользователи зачастую подразумевают под термином «сводный отчет» нечто построенное с помощью сложных формул. Попробуем популяризировать использование сводных таблиц в повседневной работе экономистов. В статье обсуждаются теоретические основы создания сводных отчетов, даются практические рекомендации по их использованию, а также приводится пример доступа к данным на основе нескольких таблиц.

Термины многомерного анализа данных

Большинство экономистов слышали термины «многомерные данные», «виртуальный куб», «OLAP-технологии» и т.п. Но при детальном разговоре обычно выясняется, что почти все не очень представляют, о чем идет речь. То есть люди подразумевают нечто сложное и обычно не имеющее отношение к их повседневной деятельности. На самом деле это не так.

Многомерные данные, измерения

Можно с уверенностью утверждать, что экономисты практически постоянно сталкиваются с многомерными данными, но пытаются представить их в предопределенном виде с помощью электронных таблиц. Под многомерностью здесь подразумевается возможность ввода, просмотра или анализа одной и той же информации с изменением внешнего вида, применением различных группировок и сортировок данных. Например, план продаж можно проанализировать по следующим критериям:

- виды или группы товаров;
- бренды или категории товаров;
- периоды (месяц, квартал, год);
- покупатели или группы покупателей;
- регионы продаж
- и т.п.

Каждый из приведенных критериев в терминах многомерного анализа данных называется «измерением». Можно сказать, что измерение характеризует информацию по определенному набору значений. Специальным типом измерения многомерной информации являются «данные». В нашем примере данными плана продаж могут являться:

- объем продаж;
- цена продажи;
- индивидуальная скидка
- и т.п.

Теоретически данные могут также являться стандартным измерением многомерной информации (например, можно сгруппировать данные по цене продажи), но обычно все-таки данные являются специальным типом значений.

Таким образом, можно сказать, что в практической работе экономисты используются два типа информации: *многомерные данные* (фактические и плановые числа, имеющие множество признаков) и справочники (характеристики или измерения данных).

OLAP

Аббревиатура OLAP (online analytical processing) в дословном переводе звучит как «аналитическая обработка в реальном времени». Определение не очень конкретное, под него можно подвести практически любой отчет любого программного продукта. По смыслу OLAP подразумевает технологию работы со специальными отчетами, включая программное обеспечение, для получения и

анализа как раз многомерных структурированных данных. Одним из популярных программных продуктов, реализующих OLAP-технологии, является SQL Server Analysis Server. Некоторые даже ошибочно считают его единственным представителем программной реализации данной концепции.

Виртуальный куб данных

«Виртуальный куб» (многомерный куб, OLAP-куб) - это специальный термин, предложенный некоторыми поставщиками специализированного программного обеспечения. OLAP-системы обычно готовят и хранят данные в собственных структурах, а специальные интерфейсы анализа (например, сводные отчеты Excel) обращаются к данным этих виртуальных кубов. При этом использование подобного выделенного хранилища совсем не обязательно для обработки многомерной информации. В общем случае, *виртуальный куб* – это и есть массив специально оптимизированных многомерных данных, который используется для создания сводных отчетов. Он может быть получен как через специализированные программные средства, так и через простой доступ к таблицам базы данных или любой другой источник, например к таблице Excel.

Сводная таблица

«Сводный отчет» (сводная таблица, Pivot Table) - это пользовательский интерфейс для отображения многомерных данных. С помощью данного интерфейса можно группировать, сортировать, фильтровать и менять расположение данных с целью получения различных аналитических выборок. Обновление отчета производится простыми средствами пользовательского интерфейса, данные автоматически агрегируются по заданным правилам, при этом не требуется дополнительный или повторный ввод какой-либо информации. Интерфейс сводных таблиц Excel является, пожалуй, самым популярным программным продуктом для работы с многомерными данными. Он поддерживает в качестве источника данных как внешние источники данных (OLAP-кубы и реляционным базам данных), так и внутренние диапазоны электронных таблиц. Начиная с версии 2000 (9.0), Excel поддерживает также графическую форму отображения многомерных данных – сводная диаграмма (Pivot Chart).

Реализованный в Excel интерфейс сводных таблиц позволяет расположить измерения многомерных данных в области рабочего листа. Для простоты можно представлять себе сводную таблицу, как отчет, лежащий сверху диапазона ячеек (на самом деле есть определенная привязка форматов ячеек к полям сводной таблицы). Сводная таблица Excel имеет четыре области отображения информации: фильтр, столбцы, строки и данные. Измерения данных именуются *полями сводной таблицы*. Эти поля имеют собственные свойства и формат отображения.

Еще раз хочется обратить внимание, что сводная таблица Excel предназначена исключительно для анализа данных без возможности редактирования информации. Ближе по смыслу было бы повсеместное употребление термина «сводный отчет» (Pivot Report), и именно так этот интерфейс и назывался до 2000го года. Но почему-то в последующих версиях разработчики от него отказались.

Редактирование сводных таблиц

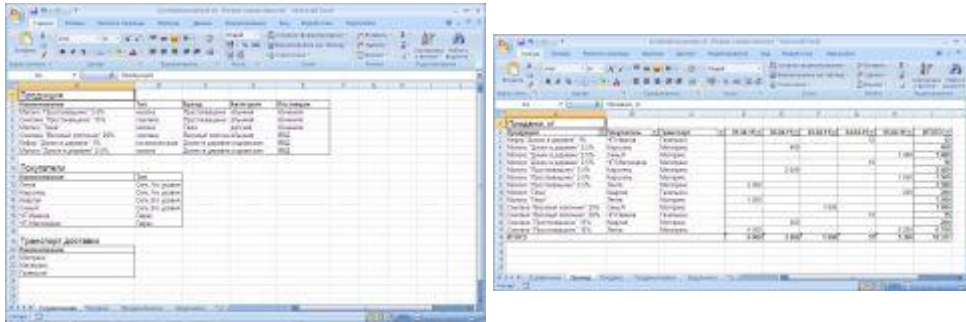
По своему определению OLAP-технология, в принципе, не подразумевает возможность изменения исходных данных при работе с отчетами. Тем не менее, на рынке сформировался целый класс программных систем, реализующих возможности как анализа, так и непосредственного редактирования данных в многомерных таблицах. В основном такие системы ориентированы на решение задач бюджетирования.

Используя встроенные средства автоматизации Excel, можно решить множество нестандартных задач.

Подготовка многомерных данных

Подойдем к практическому применению сводных таблиц. Попробуем проанализировать данные о продажах в различных направлениях. Файл **pivottableexample.xls** состоит из нескольких листов. Лист **Пример** содержит основную информацию о продажах за определенный период. Для простоты примера будем анализировать единственный числовой показатель – объем продажи в кг. Имеются следующие

ключевые измерения данных: продукция, покупатель и перевозчик (транспортная компания). Кроме того, имеются несколько дополнительных измерений данных, являющихся признаками продукта: тип, бренд, категория, поставщик, а также покупателя: тип. Эти данные собраны на листе Справочники. На практике подобных измерений может быть гораздо больше.



Лист **Пример** содержит стандартное средство анализа данных – автофильтр. Глядя на пример заполнения таблицы, очевидно, что нормальному анализу поддаются данные о продажах по датам (они расположены по столбцам). Кроме того, используя автофильтр можно попробовать просуммировать данные по сочетаниям одного или нескольких ключевых критериев. Совершенно отсутствует информация о брендах, категориях и типах. Нет возможности сгруппировать данные с автоматическим суммированием по определенному ключу (например, по покупателям). Кроме того, набор дат зафиксирован, и просмотреть итоговую информацию за определенный период, например, 3 дня, автоматическими средствами не удастся.

Вообще, наличие предопределенного расположения даты в данном примере – главный недостаток таблицы. Расположив даты по столбцам, мы как бы предопределили измерение этой таблицы, таким образом, лишив себя возможности использовать анализ с помощью сводных таблиц.

Во-первых, надо избавиться от этого недостатка – т.е. убрать предопределенное расположение одного из измерений исходных данных. Пример корректной таблицы – лист **Продажи**.

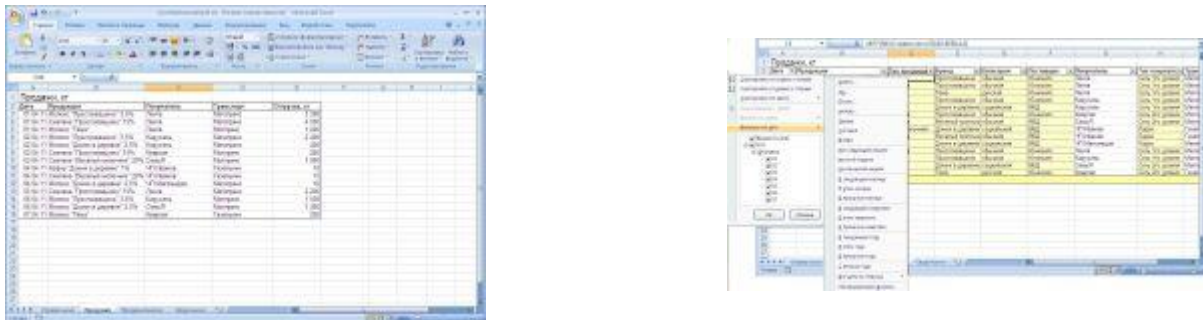
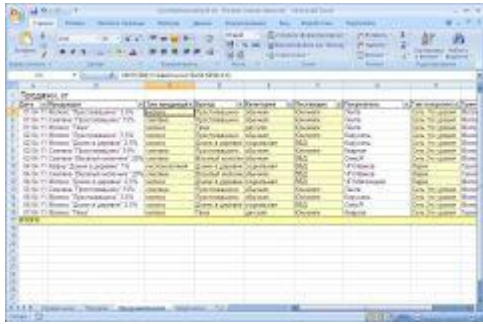


Таблица имеет форму журнала ввода информации. Здесь дата является равноправным измерением данных. Также следует заметить, что для последующего анализа в сводных таблицах совершенно безразлично относительное положение строк друг относительно друга (иначе говоря, сортировка). Этими свойствами обладают записи в реляционных базах данных. Именно на анализ больших объемов баз данных ориентирован в первую очередь интерфейс сводных таблиц. Поэтому необходимо придерживаться этих правил и при работе с источником данных в виде диапазонов ячеек. При этом никто не запрещает использовать в работе интерфейсные средства Excel – сводные таблицы анализируют только данные, а форматирование, фильтры, группировки и сортировки исходных ячеек могут быть произвольными.

От автофильтра к сводному отчету

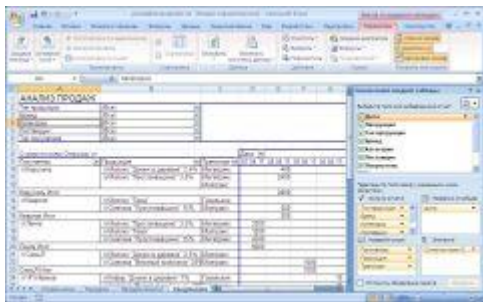
Теоретически на данных листа Продажи уже можно проводить анализ в трех измерениях: товары, покупатели и перевозчики. Данные о свойствах продукции и покупателей на данном листе отсутствуют, что, соответственно, не позволит показать их и в сводной таблице. В нормальном режиме создания сводной таблицы для исходных данных Excel не позволяет связывать данные нескольких таблиц по определенным полям. Чтобы не прибегать к программным методам обработки информации

(тем более, что они и не универсальны), следует добавить дополнительные характеристики непосредственно в форму ввода журнала – см. лист ПродажиАнализ.



Применение функций ВПР позволяет легко дополнить исходные данные недостающими характеристиками. Теперь, применяя автофильтр, можно анализировать данные в различных измерениях. Но остается нерешенной проблема группировок. Например, отследить сумму только по брендам на определенные даты достаточно проблематично. Если ограничиваться формулами Excel, то нужно строить дополнительные выборки, используя функцию СУММЕСЛИ.

Теперь посмотрим какие возможности дает интерфейс сводных таблиц. На листе **СводАнализ** построено несколько отчетов на основе диапазона ячеек с данными листа **ПродажиАнализ**.



АНАЛИЗ ПРОДАЖ 2

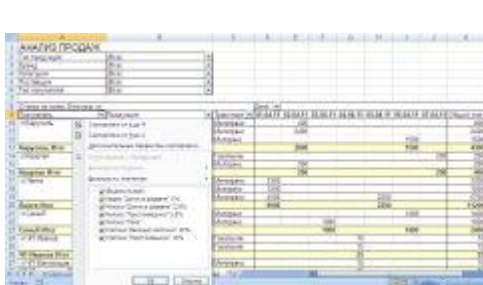
	Гаральман	Мегатранс	Молотранс	Grand Total
Карусель	200	2 800	1 500	4 500
Лента		19 000	1 200	11 200
Семей			2 400	2 400
ЧП Иванов	27			27
ЧП Масленников			10	10
Grand Total	227	12 810	5 300	18 337

АНАЛИЗ ПРОДАЖ 3

Сумма по полю Отгрузки, и	Дата	Гаральман	Мегатранс	Молотранс	Общий итог
Гарусель	02.04.11		2 800		2 800
Гарусель	05.04.11			1 500	1 500
Карусель Итого			2 800	1 500	4 300
Гарусель	02.04.11			200	200
Лента Итого			200	200	400
Лента	01.04.11		7 800	1 200	9 000
Лента	05.04.11		2 200		2 200
Лента Итого			10 000	1 200	11 200
Семей	02.04.11			1 000	1 000
Семей	05.04.11			1 400	1 400
Семей Итого				2 400	2 400
ЧП Иванов	04.04.11		27		27
ЧП Масленников	04.04.11		27		27
ЧП Масленников Итого			10	10	10
Общий итог		227	12 810	5 300	18 337

Первая таблица анализа построена через интерфейс Excel 2007 **Лента \ Вставка \ Сводная таблица** (в Excel 2000-2003 меню **Данные \ Сводная таблица**).

Вторая и третья таблицы созданы через копирование и последующую настройку. Источник данных для всех таблиц один и тот же. Можете это проверить, изменив исходные данные, затем надо обновить данные сводных отчетов.



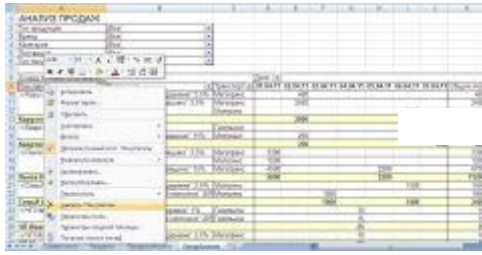
АНАЛИЗ ПРОДАЖ

Сумма по полю Отгрузки, и	Дата	Гаральман	Мегатранс	Молотранс	Общий итог
Гарусель	02.04.11		2 800		2 800
Гарусель	05.04.11			1 500	1 500
Карусель Итого			2 800	1 500	4 300
Гарусель	02.04.11			200	200
Лента Итого			200	200	400
Лента	01.04.11		7 800	1 200	9 000
Лента	05.04.11		2 200		2 200
Лента Итого			10 000	1 200	11 200
Семей	02.04.11			1 000	1 000
Семей	05.04.11			1 400	1 400
Семей Итого				2 400	2 400
ЧП Иванов	04.04.11		27		27
ЧП Масленников	04.04.11		27		27
ЧП Масленников Итого			10	10	10
Общий итог		227	12 810	5 300	18 337

С нашей точки зрения, преимущества в наглядности информации очевидны. Вы можете менять местами фильтры, столбцы и строки и, скрывать определенные группы значений любых измерений, применять ручное перетаскивание и автоматическую сортировку.

Свойства и форматирование

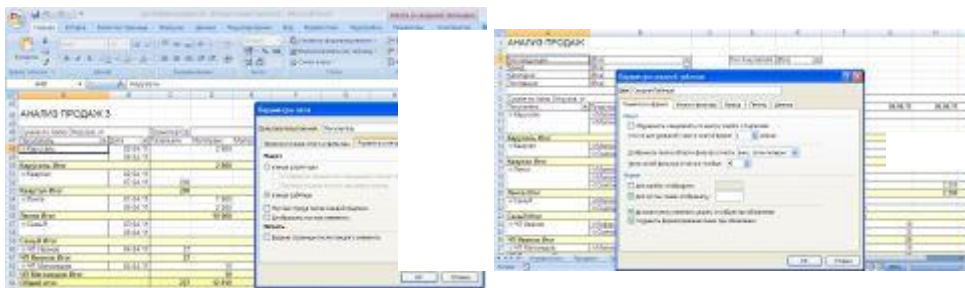
Кроме непосредственного отображения данных, имеется большой набор возможностей по отображению внешнего вида сводных таблиц. Лишние данные можно скрывать, используя фильтры. Для единичного элемента или поля проще пользоваться пунктом контекстного меню **Удалить** (в версии 2000-2003 **Скрыть**).



Для полей данных таблицы можно задать единый формат отображения. Это делается не через формат ячейки, а через специальный диалог настройки формата поля.

Задавать отображение других элементов сводной таблицы также желательно не через форматирование ячейки, а через настройку поля или элемента сводной таблицы. Для этого необходимо подвести указатель мыши к нужному элементу, дождаться появления специальной формы курсора (в виде стрелки), затем через одинарный клик выделить выбранный элемент. После выделения можно изменять вид через ленту, контекстное меню или вызывать стандартный диалог формата ячейки:

Кроме формата отображения в ячейках, сводная таблица включает несколько специальных свойств, управляющих внешним видом и расположением элементов. Диалоги настройки проще всего вызываются через контекстное меню: **Параметры сводной таблицы** или **Параметры поля сводной таблицы**.



Кроме того, в Excel 2007 появилось множество predefined стилей отображения сводной таблицы:

Сводная диаграмма

Нажав кнопку на ленте «Сводная диаграмма», можно сформировать специальный тип диаграммы, отображающей данные сводной таблицы:

Обратите внимание, что в диаграмме активны управляющие фильтры и области перетаскивания.

Доступ к внешним данным

Как уже отмечалось, пожалуй, наибольший эффект от применения сводных таблиц можно получить при доступе к данным внешних источников – OLAP-кубам и запросам к базам данных. Такие источники обычно хранят большие объемы информации, а также имеют predefined реляционную структуру, что позволяет легко определить измерения многомерных данных (поля сводной таблицы).

Срезы

Срезы — это визуальные элементы управления, которые позволяют быстро и интуитивно фильтровать данные в сводных таблицах в интерактивном режиме. Вставив срез, можно с помощью кнопок быстро выделить и отфильтровать данные, представив их в нужном виде. Кроме того, если к данным сводной таблицы применено более одного фильтра, больше не нужно открывать список, чтобы выяснить, что это за фильтры: они отображаются непосредственно на срезе. Срезы можно форматировать в соответствии с форматом книги и повторно использовать их в других сводных таблицах, сводных диаграммах и функциях куба.

Год		
2006	2007	2008
2009		
Метки строк	Сумма за год	Итоговый объем продаж
Горные велосипеды	134 067 600 р.	174 195 623 р.
Австралия	49 572 900 р.	65 197 982 р.
Канада	6 221 700 р.	7 971 132 р.
Китай	4 214 700 р.	7 114 979 р.
Франция	9 232 200 р.	10 923 083 р.
Германия	8 830 800 р.	11 326 298 р.
Италия	401 400 р.	677 498 р.
Япония	3 411 900 р.	5 752 483 р.
Норвегия	802 800 р.	1 354 996 р.
Соединенное королевство	14 851 800 р.	18 960 199 р.
Соединенные Штаты	36 527 400 р.	44 916 974 р.
Уличные велосипеды	474 856 200 р.	592 899 729 р.
Австралия	122 828 400 р.	150 230 507 р.
Канада	39 136 500 р.	54 189 107 р.

Улучшенное условное форматирование

С помощью условного форматирования можно легко выделять необходимые ячейки или диапазоны, подчеркивать необычные значения и визуализировать данные с помощью гистограмм, цветовых шкал и наборов значков. Ниже описаны новые возможности, делающие форматирование в приложении Excel 2010 еще более гибким.

Новые наборы значков Впервые появившиеся в Office Excel 2007 наборы значков позволяют пометить различные категории данных в зависимости от заданного порогового значения. Например, можно обозначить зеленой стрелкой вверх большие значения, желтой горизонтальной стрелкой — средние, а красной стрелкой вниз — малые. В приложении Excel 2010 доступны дополнительные наборы значков, включая треугольники, звездочки и рамки. Кроме того, можно смешивать и сопоставлять значки из разных наборов и легко скрывать их из вида — например, отображать значки только для показателей высокой прибыли и не показывать их для средних и низких значений.

Дополнительные параметры гистограмм В приложении Excel 2010 доступны новые параметры форматирования гистограмм. Теперь можно применять сплошную заливку и границы, а также задавать направление столбцов "справа налево" вместо "слева направо". Кроме того, столбцы для отрицательных значений теперь отображаются с противоположной стороны от оси относительно положительных.

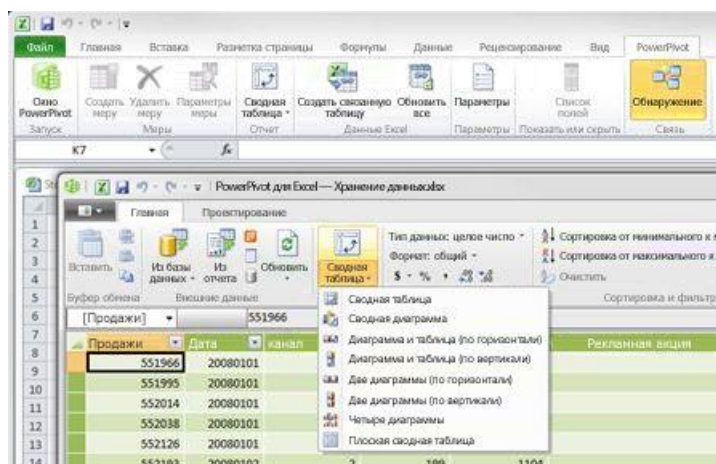
Другие улучшения Задавая критерии для условного форматирования и правил проверки данных, теперь можно ссылаться на значения на других листах книги.

Мощные инструменты анализа на рабочем столе

Как дома, так и на работе приходится оперировать различными данными и анализировать их с целью принятия оптимальных решений — и чем быстрее удастся выполнить задачу, тем лучше. В приложении Excel 2010 доступны новые и усовершенствованные инструменты анализа, позволяющие добиваться намеченных целей.

Настройка PowerPivot для Excel

Если необходимо проанализировать большие объемы данных, можно загрузить надстройку Microsoft SQL Server PowerPivot для Excel. При этом на ленте Excel появится вкладка PowerPivot.



С помощью надстройки PowerPivot для Excel можно импортировать миллионы строк данных из нескольких источников данных в одну книгу Excel, создавать отношения между разнородными данными, создавать вычисляемые столбцы и измерения с помощью формул, строить сводные таблицы и сводные диаграммы с последующим анализом данных, чтобы можно было своевременно принимать деловые решения, причем без содействия со стороны ИТ-специалистов.

Улучшенная надстройка "Поиск решения"

В состав приложения Excel 2010 входит новая версия надстройки "Поиск решения", позволяющая выполнять анализ "что если" и находить оптимальные решения. Последняя версия этой надстройки обладает улучшенным пользовательским интерфейсом Evolutionary Solver, основанным на алгоритмах генетического анализа, для работы с моделями, в которых используются любые функции Excel. В ней предусмотрены новые глобальные параметры оптимизации, улучшенные методы линейного программирования и нелинейной оптимизации, а также новые отчеты о линейности и допустимости. Кроме того, надстройка "Поиск решения" теперь доступна и в 64-разрядной версии.

Дополнительные справочные сведения о надстройке "Поиск решения" компании Frontline Systems см. в справке Solver на веб-сайте www.solver.com.

Повышенная точность функций

В ответ на пожелания академического, инженерного и научного сообществ в приложении Excel 2010 представлен набор более точных статистических и других функций. Кроме того, некоторые из существующих функций переименованы таким образом, чтобы имена лучше описывали их назначение.

Повышенная точность Ряд функций был оптимизирован в целях повышения точности. Например, функции бета-распределения и распределения хи-квадрат в приложении Excel 2010 возвращают более точные результаты, чем в предыдущих версиях.

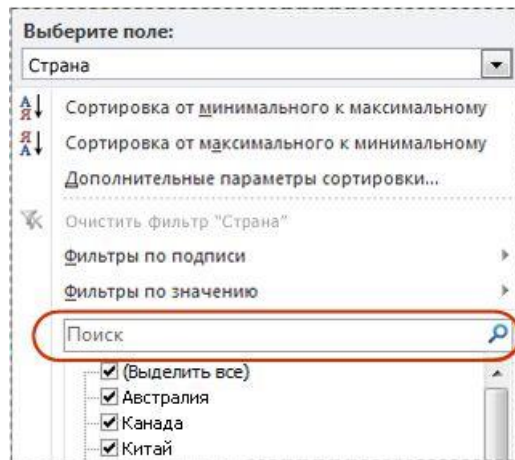
Согласованность функций Имена некоторых статистических функций приведены в соответствие с терминами, используемыми научным сообществом, а также с другими именами функций Excel. Новые имена более точно описывают назначение соответствующих функций. Книги, созданные в более ранних версиях Excel, будут продолжать работать несмотря на эти изменения, поскольку исходные функции по-прежнему доступны в категории **Совместимость**.

Расширенные возможности фильтрации

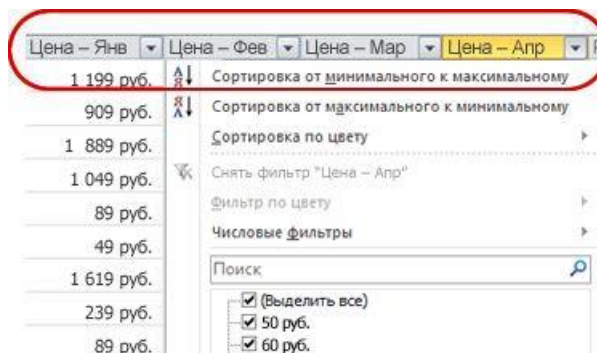
В дополнение к срезам, которые описаны выше в этой статье, в приложении Excel 2010 доступны новые возможности, которые упрощают сортировку и фильтрацию данных.

Новый фильтр поиска При фильтрации данных в таблицах Excel, сводных таблицах и сводных диаграммах доступно новое поле поиска, позволяющее находить нужные данные в больших списках. Например, чтобы найти определенный товар в каталоге, содержащем более 100 000 позиций, просто

начните вводить искомое наименование — и подходящие элементы мгновенно появятся в списке. Можно дополнительно сузить результаты поиска, отменив выделение элементов, которые не требуется отображать.



Фильтрация и сортировка независимо от расположения При прокручивании длинной таблицы Excel ее заголовки заменяют собой обычные заголовки листа, находящиеся над столбцами. Кнопки автофильтра теперь остаются видимыми вместе с заголовками таблицы в ее столбцах, что позволяет быстро сортировать и фильтровать данные, не прокручивая таблицу для отображения ее верхней части.



Повышенная производительность

Улучшения производительности в приложении Excel 2010 помогают повысить эффективность работы с данными. Ниже перечислены некоторые из них.

Общие улучшения В ответ на пожелания пользователей в приложении Excel 2010 повышена производительность в ряде областей. Например, в Excel 2010 ускорена реакция на перемещение диаграмм и изменение их размера, на действия в режиме разметки страницы, а также на операции с фигурами на листе.

Поддержка больших наборов данных В приложении Excel 2010 повышена эффективность работы с книгами, содержащими большие объемы данных. В частности, меньше времени теперь занимают стандартные операции с крупными наборами данных (такие как фильтрация и сортировка, копирование и вставка данных с одного листа на другой и копирование формул с помощью функции заполнения).

Улучшенная поддержка многоядерности Усовершенствования в приложении Excel 2010, связанные с многопоточной обработкой, позволяют ускорить получение, сортировку и фильтрацию данных в сводных таблицах и таблицах Excel. Кроме того, открытие и сохранение больших файлов в среднем выполняется быстрее, чем раньше.

Ускоренные вычисления В ситуациях, когда книги Excel являются ключевым элементом бизнес-процессов организации, крайне важно сделать так, что скорость вычислений не превратилась в

узкое место процесса. Чтобы обеспечить более высокую производительность при вычислениях, в приложении Excel 2010 предусмотрена поддержка асинхронных пользовательских функций, которые могут работать одновременно, не задействуя несколько вычислительных потоков Excel. Эта возможность особенно полезна при импорте данных на листы нестандартными способами, а также при выполнении ресурсоемких вычислений.

Улучшенные диаграммы

В приложении Excel 2010 упрощена работа с диаграммами. Ниже перечислены конкретные улучшения.

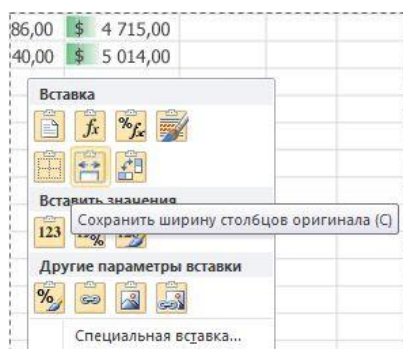
Новые ограничения для диаграмм В приложении Excel 2010 снято ограничение на количество точек данных, которые могут быть созданы на диаграмме. Количество точек данных ограничивается только объемом доступной оперативной памяти. Это позволяет пользователям, в особенности занимающимся научными исследованиями, более эффективно представлять и анализировать большие объемы данных.

Быстрый доступ к параметрам форматирования В приложении Excel 2010 можно получить мгновенный доступ к параметрам форматирования, дважды щелкнув элемент диаграммы.

Запись макросов для элементов диаграмм В приложении Office Excel 2007 в результате записи макроса при форматировании диаграммы или другого объекта никакой код не возвращался. Однако в Excel 2010 при записи макроса можно записать изменения формата, вносимые в диаграммы и другие объекты.

Вставка с динамическим просмотром

Функция динамического просмотра при вставке помогает сэкономить время при повторном использовании данных в приложении Excel 2010 и других программах. С ее помощью можно оценивать результат применения различных параметров вставки, таких как Сохранить ширину столбцов оригинала, Без рамок и Сохранить форматы оригинала. Функция динамического просмотра позволяет визуально оценить вставляемые данные еще до их фактической вставки на лист. При наведении указателя мыши на тот или иной параметр вставки и просмотре возможных результатов появляется меню с элементами, которые меняются в зависимости от контекста в соответствии с типом повторно используемых данных. Дополнительные сведения, помогающие принять правильное решение, отображаются во всплывающих подсказках.



Новые способы совместной работы с книгами

В приложении Excel 2010 улучшены возможности публикации и редактирования книг, а также их совместного использования с другими сотрудниками организации.

Совместное редактирование книг

С помощью веб-приложения Excel Web App, которое входит в пакет Office Web Apps, пользователи, находящиеся в различных расположениях, теперь могут одновременно изменять одну и ту же книгу. Сотрудникам небольших компаний и индивидуальным пользователям, работающим дома или в учебных заведениях, для одновременной работы над книгами понадобится только бесплатная учетная запись Windows Live. Корпоративные пользователи, использующие технологию Microsoft SharePoint 2010, также могут использовать эти функции в пределах действия брандмауэра.

Усовершенствованный пользовательский интерфейс К числу наиболее заметных изменений относится возможность обновлять не всю страницу, а только ее отдельные элементы, а также добавление полос прокрутки, позволяющих легко перемещаться по листу.

Более тесная интеграция с функциями SharePoint 2010 В этой версии продукта Службы Excel улучшена интеграция с важнейшими функциями SharePoint 2010 (такими как обеспечение безопасности, управление контентом, управление версиями, управление подключениями к данным и администрирование служб). Кроме того, Службы Excel еще теснее интегрированы со встроенными функциями бизнес-аналитики SharePoint.

Улучшенная поддержка возможностей книг Ранее книги с неподдерживаемыми возможностями не открывались в браузере ни при каких условиях. Теперь такую книгу Excel в большинстве случаев удастся открыть с помощью браузера. Кроме того, Службы Excel теперь поддерживают больше функций приложения Excel 2010, включая новые возможности, например спарклайны и срезы.

Улучшенная поддержка разработки приложений Разработчики и другие пользователи оценят новые инструменты, например программный интерфейс REST, предназначенный для создания бизнес-приложений.

ЛЕКЦИЯ №3

Тема: Введение в Visual Basic .Net

1 Платформа .Net	40
2 Пространство имен .Net	40
3 Константы и переменные.....	44
4 Структура текста алгоритма.....	46
5 Математические операторы. Функции.....	47

1 Платформа .Net

Visual Basic .Net является лишь частью платформы .Net. Поэтому, прежде чем изучать Visual Basic, рассмотрим в общих чертах платформу .Net.

Платформа .Net — это совокупность компонентов и технологий, в число которых входят CLR (Common Language Runtime), ADO.Net (ActiveX Data Objects), промежуточный язык Microsoft (Intermediate Language) и др. Скажем о них несколько слов.

Среда Common Language Runtime

Среда CLR — это общая среда исполнения приложений для всех языков .Net.

Данная среда используется и для Visual Basic, и для C#, а также для многих других языков программирования. В настоящее время таких языков насчитывается более шестнадцати.

Все языки платформы .Net используют одну и ту же графическую оболочку для создания приложений, одинаковый механизм обработки исключительных ситуаций, одинаковый механизм форм и многое другое.

Преимущество данного подхода очевидно при создании больших коллективных проектов. Например, одна группа программистов пишет свою часть кода на языке Visual Basic, а другая на C#. Так как и тот и другой язык используют одинаковую CLR, эти разные части одного проекта можно без труда объединить. Более того, части кода, созданные на Visual Basic, можно использовать в программах, написанных на C#, и наоборот.

Еще одно преимущество CLR заключается в том, что для всех языков программирования, поддерживающих CLR, используются одинаковые средства отладки программы.

Так как программа, написанная на любом из языков .Net, и используемые ею ресурсы полностью управляются средой CLR, то такие программы называются управляемыми.

Любая программа, написанная на одном из языков платформы .Net, компилируется в программу на промежуточном языке, который получил название Intermediate Language (IL). Среда CLR понимает только язык IL! Таким образом, если какой-либо язык программирования можно перекомпилировать в язык IL, то этот язык будет относиться к группе .Net-языков.

В общем виде, процесс компиляции программы на .Net-языке выглядит так:

1. Создается исходный текст на базовом языке (Visual Basic .Net, Visual C++ .Net, Visual C# или другом).
2. Производится проверка синтаксиса исходного текста программы.
3. Программа перекомпилируется в промежуточный язык IL.
4. С помощью синхронного компилятора JITter программа на языке IL окончательно компилируется в машинный код для конкретного процессора.

2 Пространство имен .Net

Обращаем ваше внимание на то, что программы на языке IL не зависят от конкретного процессора. Таким образом, решается проблема переносимости кода программы с одного типа процессора на другой.

Платформа .Net основана на использовании классов. Так как классов огромное количество, то их имена могут повторяться. Чтобы избежать пересечения имен классов, платформа .Net использует пространство имен.

Пространство имен — это метод, который применяется для создания иерархической структуры всех классов с целью избежания ситуации пересечения имен классов.

Так как пространство имен обеспечивает иерархию классов, то допускается наличие двух различных классов с одинаковым именем, но существующих в разных пространствах имен. Таким образом, пространство имен — это не что иное, как область действия класса.

Основное, базовое пространство имен платформы .Net носит название **System namespace**. Оно содержит классы для обработки исключительных ситуаций, типов данных, сборки мусора и др.

Платформа .Net содержит множество пространств имен. В табл. 1.1 перечислены некоторые из наиболее часто используемых пространств имен.

Таблица Часто используемые пространства имен

Категория	Пространство имен	Описание
Компонентная модель	<code>System.CodeDom</code>	Содержит элементы и структуру исходного кода программы
	<code>System.ComponentModel</code>	Содержит описание компонентов, включая лицензионную информацию
Данные	<code>System.Data</code>	Содержит классы, обеспечивающие доступ к базам данных и источникам данных (классы ADO.Net)
	<code>System.XML</code>	Содержит классы, обеспечивающие поддержку языка XML
Сервисы среды	<code>System.Diagnostics</code>	Содержит классы для отладки приложений и отслеживания работы программ
	<code>System.Timers</code>	Содержит классы для обработки событий таймера
Локализация приложений	<code>System.Globalization</code>	Содержит классы, обеспечивающие поддержку создания многоязыковых приложений
Сеть	<code>System.Net</code>	Содержит классы для отправки и получения данных по компьютерной сети с использованием различных сетевых протоколов
Объемные задачи	<code>System.Collections</code>	Содержит классы, поддерживающие коллекции объектов, таких как массивы, словари и др.
	<code>System.IO</code>	Содержит классы для записи и чтения данных из потоков и файлов

Категория	Пространство имен	Описание
(прод.)	<code>System.Text</code>	Содержит классы для работы с текстом и строками
	<code>System.Threading</code>	Содержит классы для поддержки многопоточности, включая механизмы синхронизации потоков
Графический интерфейс (GUI)	<code>System.Drawing</code>	Содержит классы для доступа к GDI и рисования двумерных картинок
	<code>System.Windows.Forms</code>	Содержит классы для создания приложений Windows, которые используют интерфейс пользователя в стиле Microsoft Windows
Безопасность	<code>System.Security</code>	Содержит классы, обеспечивающие безопасность CLR
	<code>System.Security.Cryptography</code>	Содержит классы для криптографии, включая кодирование и декодирование данных, генерирование случайных чисел и поддержку цифровых подписей
Сервисы Web	<code>System.Web</code>	Содержит классы, обеспечивающие интерфейсы взаимодействия с Web-серверами

Понятие типов данных

Как вы знаете, все языки программирования дают возможность оперировать различными данными. Это могут быть целые и вещественные числа, строки текста, байты и многое другое. Все приведенные нами примеры данных относятся к различным типам. Они несовместимы друг с другом: ведь нельзя же, например, к предложению "Мне нравится Visual Basic .Net" прибавить число 235.23. Результат такого сложения трудно себе представить, т. к. предложение будет относиться к строковому типу данных, а число 235.23 — к типу данных вещественных чисел. Таким образом, в языках программирования существует четкое деление разнообразной информации на *типы данных*. Над различными типами данных допустимы различные операции, которые не могут выполняться над другими.

В табл. 2.1 приведены основные типы данных, с которыми может работать Visual Basic .Net. Как вы можете видеть, любые типы данных имеют свои границы, выходить за которые недопустимо.

Таблица 2.1 – Типы данных

Тип данных	Размер	Диапазон значений
Integer (Целое) Предназначенный для обработки целочисленных значений	4 байта	От –2 147 483 648 до 2 147 483 647
Long (Длинное целое) Предназначенный для обработки целочисленных значений из увеличенного диапазона	4 байта	От –9,2E+18 до 9,2E+18
Single (Одинарной точности с плавающей десятичной точкой) Предназначенный для хранения чисел в экспонентной форме одинарной точности	4 байта	От –3,402824E+38 к –1,401298E-45 для отрицательных значений; от 1,401298E-45 до 3,4028235E+38 для положительных значений
Double (Двойной точности с плавающей десятичной точкой) Предназначенный для хранения чисел в экспонентной форме двойной точности	8 байт	От –1,798E+308 к –4.9407E-324 для отрицательных значений; От 4,9407E-324 до 1,7977E+308 для положительных значений
String (Срочный) Предназначенный для хранения строк разной длины	зависит от платформы	До 2 миллиардов знаков в кодировании Юникод
Boolean (Логический)	зависит от платформы	True / False
Date (Дата)	8 байт	От 0:00:00 (в полночь) 1 января 0001 года до 11:59:59 вечера 31 декабря 9999 года

Чтобы правильно использовать приведенные в табл. 3.1 типы данных, можно дать несколько рекомендаций:

- если вам необходимо хранить только два возможных значения, используйте **тип Boolean**;
- для хранения целых или вещественных значений используйте тот тип данных, который больше всего подходит по возможным допустимым значениям. Всегда стремитесь, по возможности, использовать тот тип данных, который занимает меньше байт оперативной памяти;
- для хранения текстовой или смешанной информации (текст, цифры и символы) используйте тип данных **String**;
- для хранения денежных значений рекомендуется использовать тип **Decimal**;
- для хранения и работы с датой/временем используйте специальный тип данных **Date**.

Скажем несколько слов о специфичном типе данных **object**. Этот тип данных позволяет не задумываться над тем, какого типа данные в нем хранятся, поскольку **object** позволяет хранить в себе

любые данные. На этом преимущества типа данных `object` в принципе заканчиваются. Недостатками являются большой объем памяти, расходуемой на поддержание такого типа данных, а также медленность выполнения операций над данными такого типа. Тип данных `object` можно рекомендовать к использованию только в том случае, когда действительно неизвестно, какой тип данных вам будет необходим.

Преобразование типов данных

Иногда возникает необходимость преобразовать один тип данных в другой. Например, числовое значение перевести в строковое. Такой процесс изменения типа данных носит название *преобразования*. Для преобразования данных из одного типа в другой в Visual Basic .Net используется специальный набор функций. Перечень этих функций приведен в табл.

Функция	Описание
<code>CBool</code>	Преобразует выражение в тип <code>Boolean</code>
<code>CByte</code>	Преобразует выражение в тип <code>Byte</code>
<code>CChar</code>	Преобразует выражение в тип <code>Char</code>
<code>CDate</code>	Преобразует выражение в тип <code>Date</code>
<code>CDbl</code>	Преобразует выражение в тип <code>Double</code>
<code>CDec</code>	Преобразует выражение в тип <code>Decimal</code>
<code>CInt</code>	Преобразует выражение в тип <code>Integer</code>
<code>CLng</code>	Преобразует выражение в тип <code>Long</code>
<code>CObj</code>	Преобразует выражение в тип <code>Object</code>
<code>CShort</code>	Преобразует выражение в тип <code>Short</code>
<code>CSng</code>	Преобразует выражение в тип <code>Single</code>
<code>CStr</code>	Преобразует выражение в тип <code>String</code>

Примеры

```
CInt (12.4)
' в результате получим число 12 типа Integer
CDbl (2895)
' в результате получим число 2895 типа Double
CStr (7468.30)
' в результате получим строку "7468.30".
```

3 Константы и переменные

При написании программ вам часто нужно будет хранить какие-либо значения в памяти компьютера. Для этого в Visual Basic .Net имеются константы, переменные и массивы.

Константы

Константы — это постоянные величины, которые не изменяют свои значения на всем протяжении работы вашей программы.

Константам присваиваются какие-либо определенные значения на этапе создания программы, и эти значения не изменяются во время ее работы.

Использование в вашей программе констант дает некоторые преимущества, приведенные ниже.

1) Упрощается проблема ввода данных. Часто в приложениях, где используется число Π , приблизительно равное 3.141592653, применяется константа, которой присвоено данное значение, например `c_pi`.

2) Упрощается процедура внесения изменений в программу. Например, нужно срочно поменять во всей программе процентное значение подоходного налога. Пусть он был 13 %, а стал 10 %. Если в программе это значение не было внесено в константу, то будет достаточно трудно поменять все числа 13 на 10 по ходу программы.

3) Текст программы будет легче читаться. Проще читается выражение

`2*c_pi*R`, чем `2 * 3.141592653 * 283`.

`Const < Имя > [ AS < Тип > ] = < Значение >`

`Const Pi=3.14159`

Переменные

Переменная — это некоторая область памяти, предназначенная для хранения каких-либо изменяющихся в процессе выполнения программы данных.

Имя переменной (как и константы) может иметь длину до 255 символов и должно обязательно начинаться с буквы, хотя допускается использование переменных (и констант), начинающихся символом подчеркивания `_`. Например, `MyVariable` и `MyVariable` – Правильные Имена Переменной, а `2Variable`, `?Variable` – Неправильные Имена.

Внутри имени переменной (константы) допускается использовать произвольные комбинации букв, цифр и символов подчеркивания. Регистр символов КОМПИЛЯТОРОМ Игнорируется. Например, `MyVariable` и `myvariable` – одна и та же переменная.

Переменные похожи на константы, но константы всегда содержат только одно постоянное значение, а переменные могут изменять свое значение по ходу выполнения программы.

В среде Visual Basic .Net обязательно перед использованием переменной описывать ее. Такое описание носит название *явного описания переменной*. Кроме того, вы должны указывать тип данных для каждой описываемой переменной.

Для задания (описания, объявления) переменной используется следующая синтаксическая конструкция:

`Dim имя переменной As тип данных = начальное значение переменной`

Например, для того чтобы описать целочисленную переменную `intA` с начальным значением 10, вы можете написать строку:

`Dim intA As Integer = 10.`

Переменные одного типа данных в Visual Basic .Net могут описываться одной строкой:

`Dim переменная1, переменная2, переменнаяN As тип данных`

Например:

`Dim intA, intB, intC, intD As Integer`

Здесь описаны четыре целочисленные переменные: `intA`, `intB`, `intC` и `intD`. Хотя можно было их описать и стандартным способом:

```
Dim intA As Integer
Dim intB As Integer
Dim intC As Integer
Dim intD As Integer
```

Как вы уже, наверное, заметили, начальное значение переменной можно не задавать. Если же начальное значение задано, то не стоит думать, что переменная превратится в константу. Это просто начальное значение, которое можно изменять в процессе выполнения программы.

Отметим, что для каждого типа данных в Visual Basic .Net имеется так называемое *значение по умолчанию*. Это значение присваивается переменной, если ей не было задано начальное значение. Для всех числовых типов данных — это значение 0, а для строк — пустая строка `""`.

Таким образом, все четыре переменные, объявленные нами ранее (intA, intB, inte и into) получат по умолчанию начальное значение 0.

Для того чтобы *присвоить* переменной какое-либо значение, используется знак равенства =. Например:

```
intA = 67
```

Для присвоения значения строковой переменной (string) это значение должно быть заключено в кавычки:

```
strName = "Иванов Иван Иванович"
```

Для присвоения значения переменной даты и времени (Date) это значение должно быть заключено между символами #:

```
dteNewDate = #28/03/2003#
```

4 Структура текста алгоритма

Комментарии

Кроме команд и выражений можно включить у методы и процедуры произвольный текст или комментарии. Комментарии, которые объясняют текст программы, сделают ее более читабельной и помогают лучше ориентироваться в программном коде.

Для включения в текст программы комментария необходимо ввести символ ['], который может быть первым символом строке или находиться в будь-какому ее месте. Этот символ означает начало комментария. Будь-какой текст, расположенный в строке вслед за этим символом, будет восприниматься как комментарий, а не как программный код для трансляции.

```
Imports System.Math 'импорт пространства имен System.Math
```

Размещение оператора на нескольких строках

В случае, если оператор имеет большую длину, его можно разбить на несколько строк, используя символы продолжения строки, которые представляют собой пробел, по которому следует символ подчеркивания [_].

Размещение нескольких операторов на одной строке

Как правило, при написании программ операторы размещают на отдельной строке. Если операторы имеют небольшую длину, можно их поместить на одной строке, разделив двоеточием [:].

Таблица 2.3 – Примеры размещения операторов

a = 2 b = 5 c = 10	Операторы на отдельной строке
a = 2 : b = 5 : c = 10	Операторы в одну строку
Msgbox("x =" & Str(x) & Chr(13) & "cos(x) =" & Str(Cos(x)), _ vbinformation + vbokcancel, "Результат")	Конструкция разделена знаком переноса строки [_]

5 Математические операторы. Функции

В табл. приведенные условные обозначения шести базовых математических операций.

Таблица – Арифметические операции

Оператор	Операция
+	Сложение
–	Вычитание / обозначение отрицательных чисел
/	Деление
\	Целочисленное деление
*	Умножение
^	Возведение в степень

Стандартные типы функций

Функция – это оператор, который выполняет определенные действия и возвращает результат работы в программу. Функция может иметь один или несколько аргументов, которые берутся в скобки и отделяются между собой запятыми.

Круглые скобки и приоритет операций

При использовании круглых скобок вам не придется запоминать приоритеты разных операций (табл.). В VB. NET, как и во многих языках программирования, операции имеют приоритет, который определяет последовательность их выполнения. Умножение обладает более высоким приоритетом, чем сложение; таким образом, выражение $3+4*5$ равняется 23, поскольку умножение ($4*5$) выполняется раньше, чем сложение.

Ниже перечислены математические операции в порядке снижения приоритета.

Таблица – Арифметические операции по приоритету

Операторы	Порядок вычисления
()	Значения в круглых скобках
^	Возведение числа в степень
–	Создание отрицательного числа (изменение знака)
* /	Умножение / деление
\	Целочисленное деление
Mod	Остаток от деления
+ –	Сложение/вычитание

Если две операции имеют одинаковый приоритет, порядок выполнения определяется порядком их следования в выражении (слева направо).

Математические функции и математические константы

Математические функции и математические константы в VB. NET реализованные в виде метода класса Math, что входит в .NET Framework.

Все перечисленные методы являются общими методами класса Math, поэтому они должны вызываться с префиксом Math – например: `Math.Log10 (10)`, или импортировать класс Math в модуль: `Imports System.Math.`

Таблица – Основные математические функции

Математическая функция	Описание
Abs(арг)	Возвращает абсолютное значение (модуль) числа (арг)
Acos(арг)	Возвращает угол, косинус которого равняется заданному числу (арг)
Asin(арг)	Возвращает угол, синус которого равняется заданному числу (арг)
Atan(арг)	Возвращает угол, тангенс которого равняется заданному числу (арг)
Cos(арг)	Возвращает косинус заданного угла (арг)
Exp(арг)	Возвращает число e ($\approx 2,71828182845905$), введенное в заданную степень (арг)
Log(арг)	Возвращает натуральный логарифм числа (арг)
Log10(арг)	Возвращает десятичный логарифм числа (арг)
Log(арг, осн)	Возвращает логарифм числа (арг) по заданному основанию (осн)
Round(арг[, точн])	Округляет число (арг) до ближайшего целого или указанного количества десятичных знаков после запятой (точн)
Sign(арг)	Возвращает величину, которая определяет знак числа (арг)
Sin(арг)	Возвращает синус заданного угла (арг)
Sqrt(арг)	Возвращает квадратный корень числа (арг)
Tan(арг)	Возвращает тангенс заданного угла (арг)

Функции ввода/вывода

Для реализации более гибкого интерфейса пользователя целесообразно вместо жесткого присваивания значений переменным или объектам использовать функцию ввода Inputbox. Для вывода результатов или сообщений программы, возможное использование функции вывода MsgBox.

Оператор ввода Inputbox

Синтаксис:

```
Inputbox (<сообщение> , [<заголовок>] , [<значение>])
```

<сообщение> / [<заголовок>]

Обязательный / необязательный параметр типа String. произвольная последовательность символов в двойных кавычках (""); текст, который содержится в окне Inputbox в качестве объяснения и в заголовке соответственно. Если текст состоит из нескольких строк, то можно разделить строки с помощью знака возврата каретки (Chr(13)), знака перехода на красную строку (Chr(10)) или их комбинации между каждой строкой;

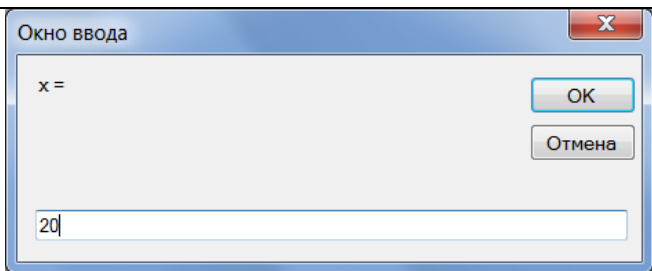
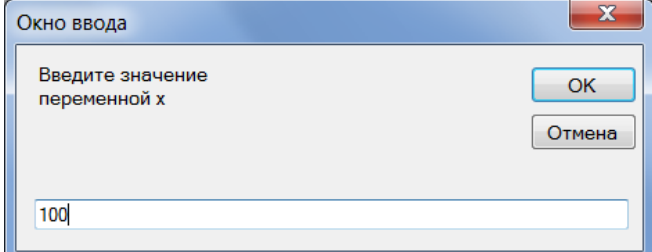
<значение>

значение по умолчанию переменной соответствующего типа.

Если необязательный параметр оператора умалчивается, ставятся запятые.

```
x = Inputbox("x = ", "Окно ввода", 20, 100, 500)
```

Таблица – Примеры использования функции ввода Inputbox

<pre>x = Inputbox(«x = », «Окно ввода», 20)</pre>	
<pre>x = Inputbox("Введите значение" & Chr(13) & "переменной x", "Окно ввода", 100)</pre>	

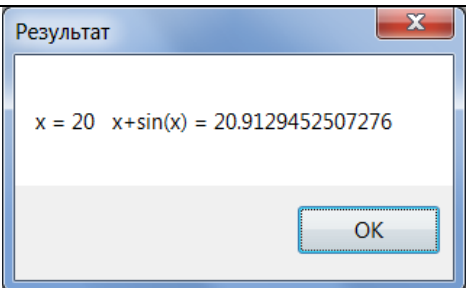
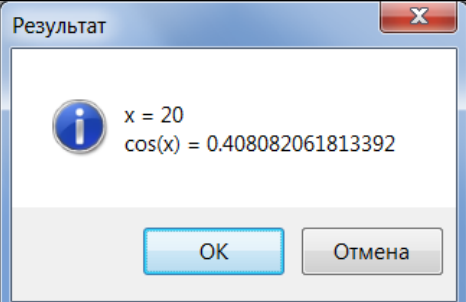
Оператор **вывода** данных MsgBox

Синтаксис:

```
Msgbox("<список вывода>" [, <кнопки>] [, "<заголовок>"])
```

<список вывода>	Обязательный параметр типа String. Текст, который отображается в диалоговом окне в виде сообщения. Если текст состоит из нескольких строк, то можно разделить строки с помощью знака возврата каретки (Chr(13)), знака перехода на красную строку (Chr(10)) или их комбинации между каждой строкой. В качестве списка вывода также удобно использовать функцию Str (аргумент). Элементы списка вывода соединяются между собой знаком &;
[<кнопки>]	Необязательный параметр. Значения, которые задают номер и тип отображаемых кнопок, стиль используемого значка, тип кнопки по умолчанию и признак модальности окна сообщения. Если данный параметр опущен, по умолчанию используется нулевое значение;
[<заголовок>]	Необязательный параметр типа String. Текст, который отображается в строке заголовка диалогового окна. Если параметр опущен, строка заголовка содержит имя приложения.

Таблица – Примеры использования функции вывода MsgBox

<pre>Msgbox("x =" & Str(x) & " " & "x+sin(x) =" & Str(x + Sin(x)), , "Результат")</pre>	
<pre>Msgbox("x =" & Str(x) & Chr(13) & "cos(x) =" & Str(Cos(x)), vbinformation + vbokcancel, "Результат")</pre>	

ЛЕКЦИЯ №4

Тема: Алгоритм. Типы алгоритмов

1 Понятие и свойства алгоритма.....	50
2 Типовые алгоритмические структуры.....	51
3. Линейный алгоритм. Вычисление выражения	53

1 Понятие и свойства алгоритма

Алгоритм («алгоритм») происходит от имени хорезмского учёного Абу Абдуллах Мухаммеда ибн Муса аль-Хорезми (алгоритм — аль-Хорезми)) — набор инструкций, описывающих порядок действий исполнителя для достижения результата решения задачи за конечное число действий. В старой трактовке вместо слова «порядок» использовалось слово «последовательность», но по мере развития параллельности в работе компьютеров слово «последовательность» стали заменять более общим словом «порядок». Это связано с тем, что работа каких-то инструкций алгоритма может быть зависима от других инструкций или результатов их работы. Таким образом, некоторые инструкции должны выполняться строго после завершения работы инструкций, от которых они зависят. Независимые инструкции или инструкции, ставшие независимыми из-за завершения работы инструкций, от которых они зависят, могут выполняться в произвольном порядке, параллельно или одновременно, если это позволяют используемые процессор и операционная система.

Исполнителем алгоритма может быть как человек (кулинарные рецепты, различные инструкции, алгоритмы математических вычислений), так и техническое устройство. Различные машины (компьютеры, промышленные роботы, современная бытовая техника) являются формальными исполнителями алгоритмов. От формального исполнителя не требуется понимание сущности решаемой задачи, но требуется точное выполнение последовательности команд.

Исходные данные для создания алгоритма

- 1) полный набор исходных данных задачи (начальное состояние объекта);
- 2) цель создания алгоритма (конечное состояние объекта);
- 3) систему команд исполнителя (то есть набор команд, которые исполнитель понимает и может выполнить).

Формальные свойства алгоритмов

Различные определения алгоритма в явной или неявной форме содержат следующий ряд общих требований:

Дискретность – алгоритм должен представлять процесс решения задачи как последовательное выполнение некоторых простых шагов. При этом для выполнения каждого шага алгоритма требуется конечный отрезок времени, то есть преобразование исходных данных в результат осуществляется во времени дискретно.

Детерминированность (определённость). В каждый момент времени следующий шаг работы однозначно определяется состоянием системы. Таким образом, алгоритм выдаёт один и тот же результат (ответ) для одних и тех же исходных данных. В современной трактовке у разных реализаций одного и того же алгоритма должен быть изоморфный граф. С другой стороны, существуют вероятностные алгоритмы, в которых следующий шаг работы зависит от текущего состояния системы и генерируемого случайного числа. Однако при включении метода генерации случайных чисел в список «исходных данных», вероятностный алгоритм становится подвидом обычного.

Понятность – алгоритм должен включать только те команды, которые доступны исполнителю и входят в его систему команд.

Завершаемость (конечность) – при корректно заданных исходных данных алгоритм должен завершать работу и выдавать результат за конечное число шагов. С другой стороны, вероятностный алгоритм может и никогда не выдать результат, но вероятность этого равна 0.

Массовость (универсальность). Алгоритм должен быть применим к разным наборам исходных данных.

Результативность – завершение алгоритма определёнными результатами.

Алгоритм содержит ошибки, если приводит к получению неправильных результатов либо не даёт результатов вовсе.

Алгоритм не содержит ошибок, если он даёт правильные результаты для любых допустимых исходных данных.

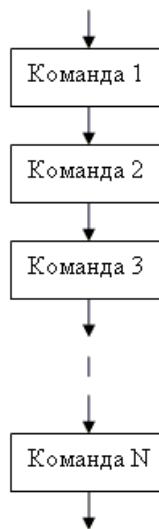
2 Типовые алгоритмические структуры

В рамках структурного программирования задачи, имеющие алгоритмическое решение, могут быть описаны с использованием следующих алгоритмических структур:

1. Линейная структура (следование).
2. Ветвление.
3. Цикл.
4. Функция (подпрограмма). Команды, отделенные от основной программы, выполняются лишь в случае их вызова из основной программы (из любого ее места). Одна и та же функция может вызываться из основной программы сколько угодно раз.

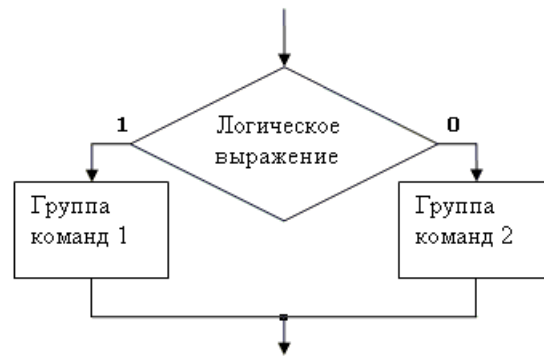
1). Линейная структура (следование).

Наиболее простой алгоритмической структурой является **линейная**. В ней все операции выполняются один раз в том порядке, в котором они записаны.

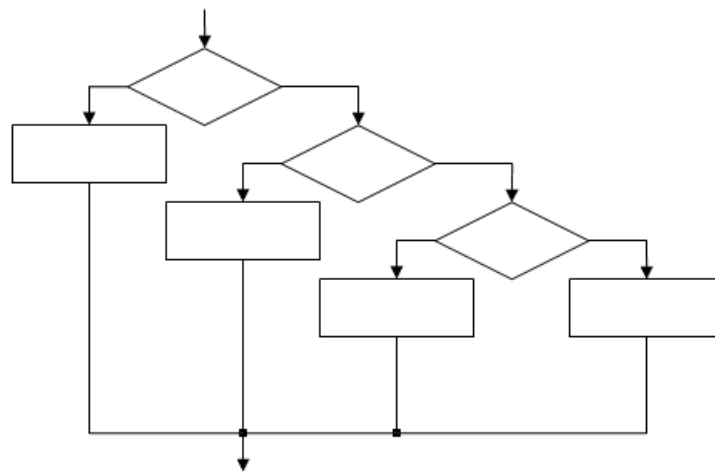


2). Ветвление.

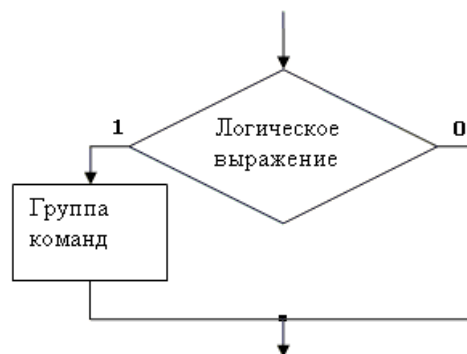
а). **Полное ветвление (ветвление IF-ELSE)** – предусматривает два варианта действий исполнителя в зависимости от значения логического выражения (условия). *Если условие истинно, то выполняться будет только первая ветвь, иначе только вторая ветвь.*



Количество условий может быть различно. Если выполняется первое, то после выполнения действий, программа переходит к основной ветке, не проверяя дальнейшие условия. Если первое условие возвращает ложь, то проверяется второе условие. Если второе условие возвращает правду, то выполняются действия, включенные в вторую ветку конструкции. Последнее условие проверяется лишь в том случае, если ни одно до него не дало в результате True.



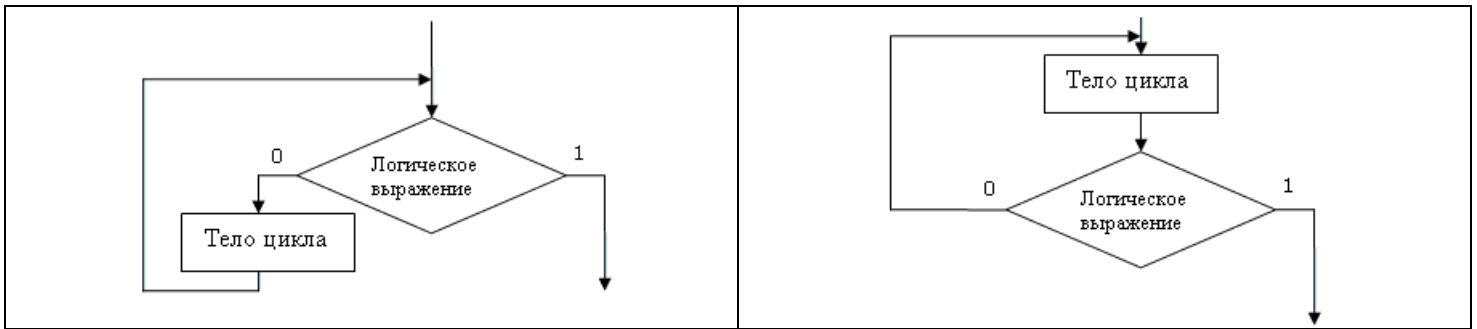
б). **Неполное ветвление или обход (ветвление IF).** Если результат вычисления выражения-условия возвращает true (правда), то выполнение алгоритма идет по ветке «Да», в которую включены дополнительные выражения-действия. Если условие возвращает false (ложь), то выполнение алгоритма идет по ветке «нет», т.е. продолжает выполняться основная ветка программы.



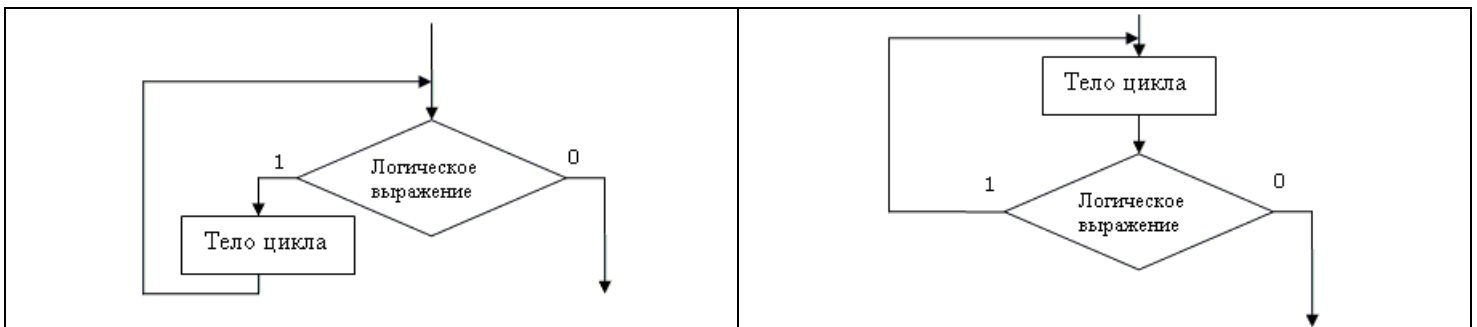
3). Цикл (повторение).

Цикл позволяет организовать многократное повторение одной и той же последовательности команд — она называется телом цикла. В различных видах циклических алгоритмов количество повторений может зависеть от значения логического выражения (условия) или может быть жестко задано в самой структуре. Различают циклы: «до», «пока», циклы со счётчиком. В циклах «до» и «пока» логическое выражение (условие) может предшествовать телу цикла (цикл с предусловием) или завершать цикл (цикл с послеусловием).

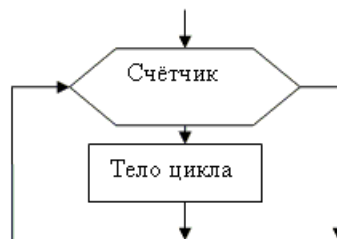
Циклы «до» - повторение тела цикла до выполнения условия:



Циклы «пока» - повторение тела цикла **пока** условие выполняется (истинно):



Циклы со счётчиком (с параметром) – повторение тела цикла заданное число раз:



4. Вспомогательный алгоритм (подпрограмма, процедура)

Вспомогательный алгоритм представляет собой модуль, к которому можно многократно обращаться из основного алгоритма. Использование вспомогательных алгоритмов может существенно уменьшить размер алгоритма и упростить его разработку.

3. Линейный алгоритм. Вычисление выражения

Условие. Вычислить значение выражения, с разработкой интерфейса пользователя:

$$\left(\sin(x) - 1 + \frac{1}{\sin^2(3x^5 + 67x^2 - 6x)} \right) * \left(\frac{x^2 - x}{(2 - x)^2} \right) \quad \text{при } x = 0,5$$

Получим эквивалентную математическую конструкцию согласно данному условию:

$$y = (\sin(x) - 1 + 1 / ((\sin(3 * x^5 + 67 * x^2 - 6 * x))^2)) * ((x^2 - x) / ((2 - x)^2))$$

Программный код алгоритма

Импортируем системный класс Math, который служит для использования математических функций и математических констант:

```
Imports System.Math ' импорт пространства имен System.Math
```

```
Public Class Form1
```

Объявляем переменные:

```
Dim y As Double, x As Double
```

```
Private Sub Form3_Load(Byval sender As System.Object, Byval e As  
System.EventArgs) Handles MyBase.Load
```

```
Label1.Text = "Вычислить, с разработкой соответствующего  
интерфейса пользователя: "
```

```
Label1.BackColor = Color.Gray
```

```
Label1.ForeColor = Color.WhiteSmoke
```

```
Label1.Font = New System.Drawing.Font("Arial", 10)
```

```
End Sub
```

```
Private Sub Numericupdown1_Click(Byval sender As Object, Byval e  
As System.EventArgs) Handles Numericupdown1.Click
```

В случае ошибки программа сразу перейдет к отметке 999

```
On Error Goto 999
```

```
x = Textbox1.Text
```

В следующем блоке вычислим выражение, согласно условию:

```
y = (Sin(x) - 1 + 1 / ((Sin(3 * x ^ 5 + 67 * x ^ 2 - 6 * x))  
2)) * ((x ^ 2 - x) / ((2 - x) ^ 2))
```

```
Textbox2.Text = y
```

Использование математической функции Round, которая округляет значение сменной y к установленной точности:

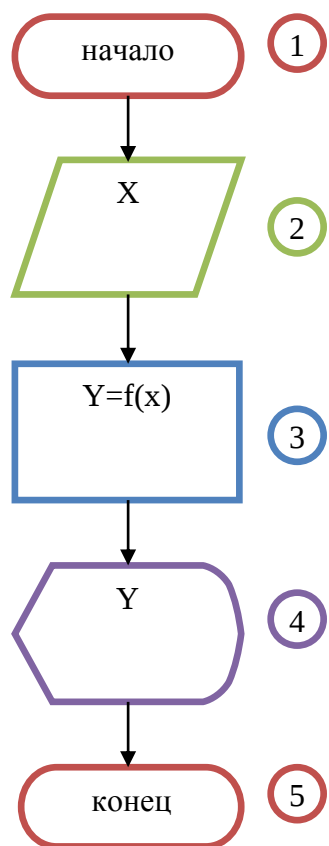
```
Textbox2.Text = Round(y, CInt(Numericupdown1.Value))
```

```
999:
```

```
End Sub
```

```
End Class
```

Принципиальная блок-схема алгоритма



- 1) начало алгоритма
- 2) формирование данных
- 3) процесс: вычисление выражения
- 4) вывод результата на дисплей
- 5) конец алгоритма

$$f(x) = \frac{(\sin(x) - 1 + 1 / ((\sin(3 * x^5 + 67 * x^2 - 6 * x)) - 2)) * ((x^2 - x) / ((2 - x)^2))$$

Тестирование алгоритма

Вычисление значения функции с заданной точностью

Вычислить, с разработкой соответствующего интерфейса пользователя:

$$\left(\sin(x) - 1 + \frac{1}{\sin^2(3x^5 + 67x^2 - 6x)} \right) * \left(\frac{x^2 - x}{(2 - x)^2} \right)$$

x= *при x = 0,5 - ?*

y=

Проверочный ответ: 0.06341253

ЛЕКЦІЯ №5

Тема: Алгоритм розгалуження. Циклічний алгоритм

1 Алгоритми розгалуження у VB.....	57
2 Циклічні алгоритми у VB	68

1 Алгоритми розгалуження у VB

Умовний оператор If...Then...Else

Умовний оператор використовується для подання розгалуженого обчислювального процесу у кодї програми мовою VB.Net. Детальна інформація про алгоритми розгалуження подана у додатку А (гл. А.1).

Умовний оператор If...Then...Else має два типи структур: рядкову та блочну. Синтаксис рядкової структури умовного оператора:

If <логічний вираз> Then <оператори1> [Else <оператори2>]

логічний вираз результатом може бути одне з двох значень (істина) або (хибність); логічний вираз складається з операндів логічного типу, між якими містяться знаки логічних операцій і може мати одне з двох значень: True (Істина) або False (Хибність).

оператори 1 виконуються при значенні логічного виразу «істина».

оператори 2 виконуються при значенні логічного виразу «хибність».

У табл. 1.1 наведено список операторів порівняння й умов.

Таблиця 1.1 – Логічні оператори та результати

Оператор	Істина, якщо	Хибність, якщо
< (менше)	expression1 < expression2	expression1 >= expression2
<= (менше або рівно)	expression1 <= expression2	expression1 > expression2
> (більше)	expression1 > expression2	expression1 <= expression2
>= (більше або рівно)	expression1 >= expression2	expression1 < expression2
= (рівно)	expression1 = expression2	expression1 <> expression2
<> (не рівно)	expression1 <> expression2	expression1 = expression2

Синтаксис блочної форми умовного оператора:

```
If <логічний вираз 1> Then
    оператори 1
[Else if <логічний вираз 2> Then
    оператори 2]
[Else
    оператори n]
End If
```

оператори 1 довільна кількість операторів, що виконуються за

логічний вираз 1	умовою:
логічний вираз 2	є «істина»;
оператори 2	повертає ненульове значення («істина») або нуль («хибність»);
логічний вираз 2	довільна кількість операторів, що виконуються за умовою:
оператори n	є «істина»
	довільна кількість операторів, що виконуються при інших умовах.

В наступному прикладі реалізований алгоритм визначення кількості цифр числа. Критерієм визначення є належність числа до одиниць, десятків, сотень (1 цифра, 2 цифри, 3 цифри відповідно).

Dim число, цифри **As Integer**

Dim строка **As String**

число = 53 ' задаємо число

'в логической конструкции проверяем принадлежность числа к одному из интервалов

If число < 10 **Then**

цифры = 1 'если число меньше 10 – оно содержит 1 цифру

ElseIf число < 100 **Then**

цифры = 2 'если число меньше 100 (и при этом больше 10, исходя из предыдущего условия) – оно содержит 2 цифры

Else

цифры = 3 'если число больше, или равно 100 (следует из предыдущих 2 условий) – оно содержит 3 цифры

End If

If цифры = 1 **Then** строка = "Одна цифра" **Else** строка = "Несколько цифр"

MsgBox(строка, **MsgBoxStyle.Information**, "Сообщение")

Логічна конструкція Select Case

Конструкція Select Case дозволяє обробляти в програмі декілька умов і аналогічна блоку конструкцій If ... Then ... Else. Ця конструкція складається з аналізованого виразу й набору операторів Case на кожне можливе значення виразу.

Синтаксис конструкції Select Case наступний:

Select Case <змінна>

CASE <значення 1>

<конструкція 1>

CASE <значення 2>

<конструкція 2>

...

End Select

змінна

значення 1, значення 2,...

змінна для порівняння;

обов'язковий в операторі Case. Список можливих значень для параметра <змінна>, що розділяються комами, який може мати вигляд (табл. 1.2);

<конструкція 1> / <конструкція 2> Один або група операторів, що реалізуються, якщо значенню <змінна> відповідає значення 1 / значення 2.

Таблиця 1.2 – Конструкції для оператора Case

значення1 То значення2	вказує діапазон значень для <змінна>
Is <оператор порівняння> <значення>	Використовується з оператором порівняння (=, <, <=, > або >=) для обмеження можливих значень <змінна>
значення	Вид, у якому надається тільки значення, розглядається як особливий випадок використання Is, де оператор порівняння – знак рівності (=). Дана форма обчислюється як змінна = значення

Логіка виконання конструкції наступна: обчислюється значення заданого в конструкції виразу. Потім отримане значення порівнюється зі значеннями, що задаються в операторах CASE конструкції. Якщо знайдено шукане значення, виконуються команди, приписані даному оператору CASE. Після завершення виконання конструкцій управління буде передано конструкції, наступної за ключовим словом End Select.

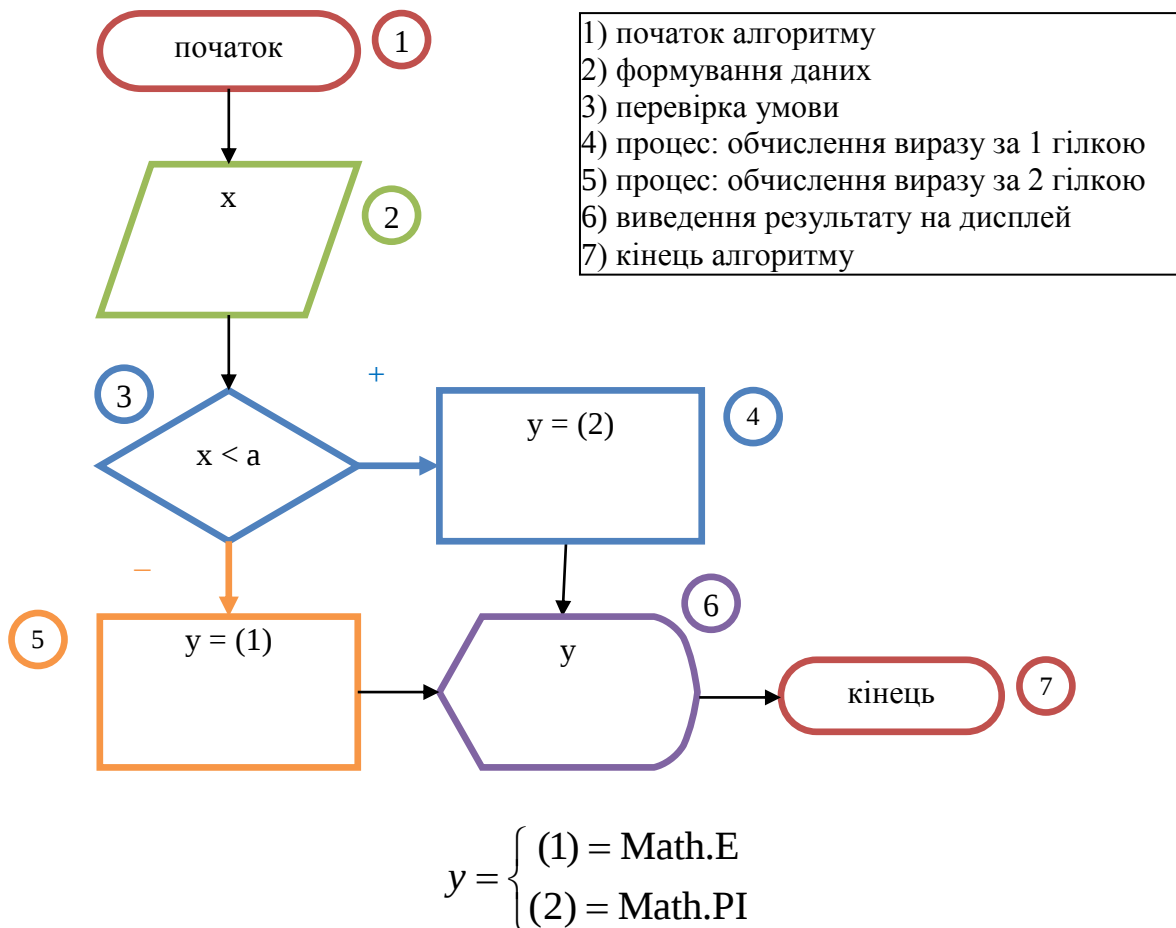
Розв'язок задачі «Алгоритм розгалуження 2 гілки» (If-рядок)

Умова. Обчислити значення кусково-аналітичної функції 2 гілки з використанням конструкції (If-рядок).

$$y = \begin{cases} e, x < a \\ \pi, x \geq a \end{cases}$$

Стислі відомості про елементи управління, які призначені для створення інтерфейсу користувача наведені у додатку Б.

Принципова блок-схема алгоритму



Програмний код алгоритму

```

Dim x, y, v, a
Dim qq As Label
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click

```

Якщо буде обраний RadioButton1, то значення аргументів будуть вводитися через текстові поля:

```

If RadioButton1.Checked = True Then
    a = CDb1(TextBox1.Text)
    x = CDb1(TextBox2.Text)

```

```

Else

```

Якщо буде обраний RadioButton2, то значення аргументів будуть вводитися через діалогові вікна, за допомогою оператора вводу InputBox:

```

    a = InputBox("Введіть значення узла a")
    x = InputBox("Введіть значення аргумента x")

```

```

End If

```

В наступному блоці реалізується алгоритм вибору гілки згідно з умовою розгалуження, представлений у вигляді строкової конструкції If...Then:

```

If x < a Then y = Math.E : v = 1 Else y = Math.PI : v = 2

```

```

MsgBox("ОТВЕТ: x=" & x & "; y=" & y & Chr(13) & "Вычисления прошли по " &
v & "-й ветви ", , "Ветвление: 2 ветви")

```

```

End Sub

```

Тестування алгоритму

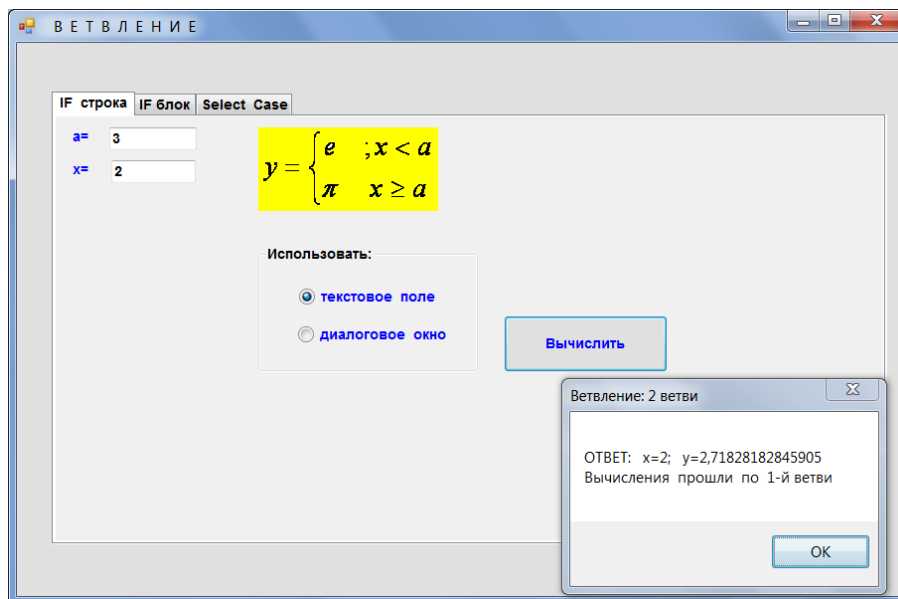


Рисунок 1.1 – Форма задачі «Алгоритм розгалуження 2 гілки»

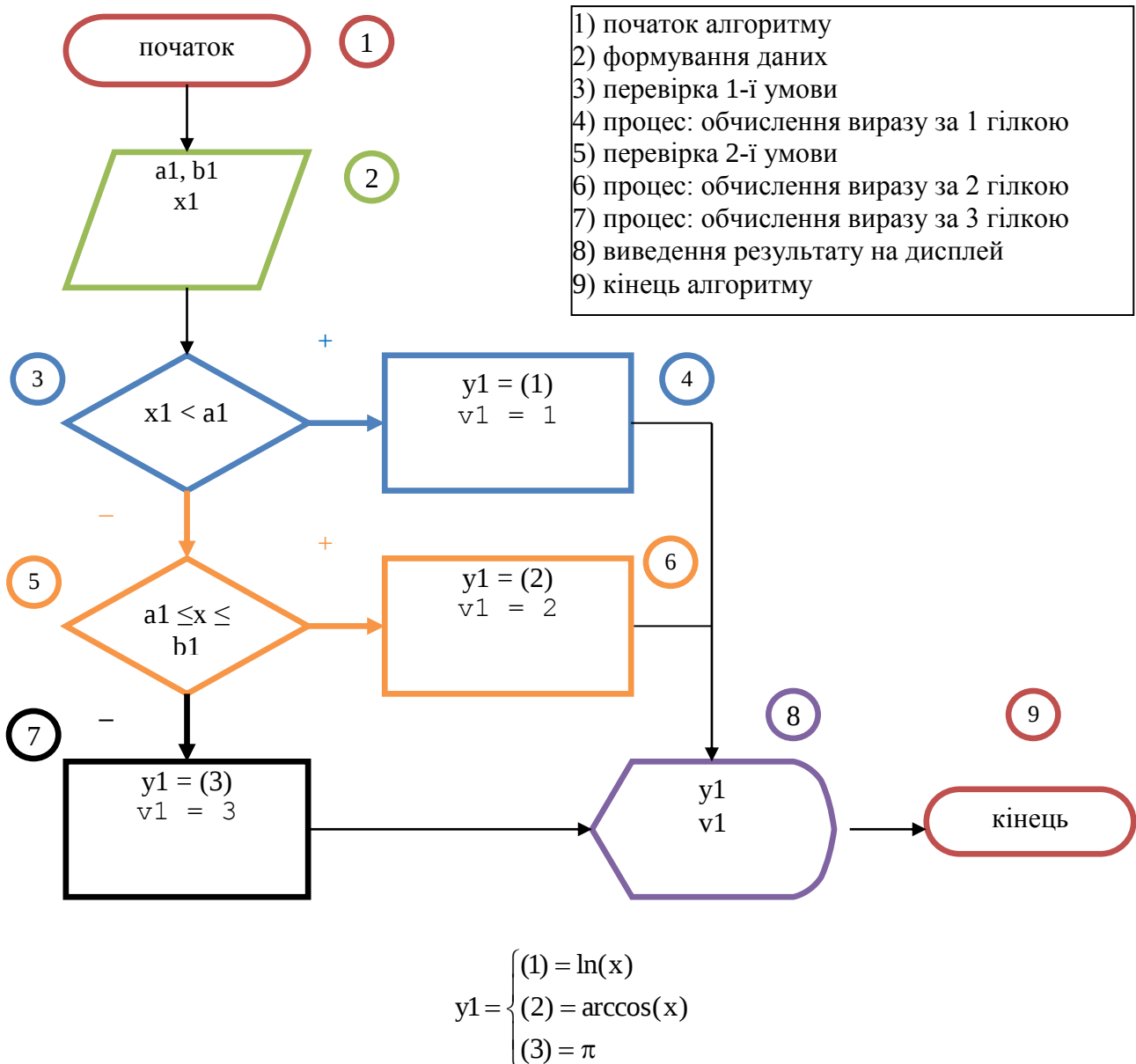
Розв’язок задачі «Алгоритм розгалуження 3 гілки» (If-блок)

Умова. Обчислити значення кусково-аналітичної функції 3 гілки з використанням конструкції (If-блок)

$$Y = \begin{cases} \ln(x), & x < a \\ \arccos(x), & a \leq x \leq b \\ \pi, & x \geq b \end{cases}$$

Стислі відомості про елементи управління, які призначенні для створення інтерфейсу користувача наведені у додатку Б.

Принципова блок-схема алгоритму



Програмний код алгоритму

```

Dim a1 As Double = -6
Dim b1 As Double = 5
Dim x1 As Double = 22
Dim Y1 As Double
Dim v1 As Double
Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button3.Click
    ListBox1.Items.Clear()
    On Error GoTo 999 'в случае ошибки управление передается метке 999
    ComboBox1.Items.Add(ComboBox1.Text) 'Если надо добавить новое значение
    a1 = ComboBox1.Text 'Задание значения узла A
  
```

```

ComboBox2.Items.Add(ComboBox2.Text) 'Если надо добавить новое значение
b1 = ComboBox2.Text 'Задание значения узла В
ComboBox3.Items.Add(ComboBox3.Text) 'Если надо добавить новое значение
x1 = ComboBox3.Text 'Задание значения узла С

```

В наступному блоці реалізується алгоритм вибору гілки згідно з умовою розгалуження, представлений у вигляді блокової конструкції If...Then:

```

If x1 < a1 Then
    Y1 = Log(x1) : v1 = 1
ElseIf x1 >= a1 And x1 <= b1 Then
    Y1 = Acos(x1) : v1 = 2
Else
    Y1 = PI : v1 = 3
End If

```

'Вывод результата в ListBox1

```

ListBox1.Items.Add("Получили значение функции:")
ListBox1.Items.Add("Y=" & Y1)
ListBox1.Items.Add("при значении аргумента:")
ListBox1.Items.Add("x=" & x1)
ListBox1.Items.Add("Вычисления прошли по " & v1 & "-й ветви") : GoTo 1000
999:  ListBox1.Items.Add("Проверь правильность ")
      ListBox1.Items.Add(" ввода данных")
1000:
End Sub
Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button4.Click
    'Запускающая кнопка
    If CheckBox1.Checked = True Then
        Button3.Visible = True
        Button4.Visible = False
        'Layout (пер с англ. – схема размещения)
        TableLayoutPanel1.Visible = True
    End If

```

***** IF 3 ветви (Стандартный вариант) *****

В наступному блоці реалізується алгоритм вибору гілки згідно з умовою розгалуження, представлений у вигляді блокової конструкції If...Then:

```

If x1 < a1 Then
    Y1 = Log(x1) : v1 = 1
ElseIf x1 >= a1 And x1 <= b1 Then
    Y1 = Acos(x1) : v1 = 2
Else
    Y1 = PI : v1 = 3
End If

```

```

Dim Ответ As Object = MsgBox("ОТВЕТ: x=" & x1 & "; y=" & Y1 & Chr(13) & _
    "Вычисления прошли по " & v1 & "-й ветви", MsgBoxStyle.Exclamation +
MsgBoxStyle.YesNo)

```

```

If Ответ = MsgBoxResult.No Then
    Dim Ответ1 As Object = MsgBox("ПРОДОЛЖАТЬ РАБОТУ ?",
MsgBoxStyle.Question + MsgBoxStyle.YesNo)
    If Ответ1 = MsgBoxResult.No Then
        TabControl1.SelectTab(2)
    End If
End If
End Sub

```

Тестування алгоритму

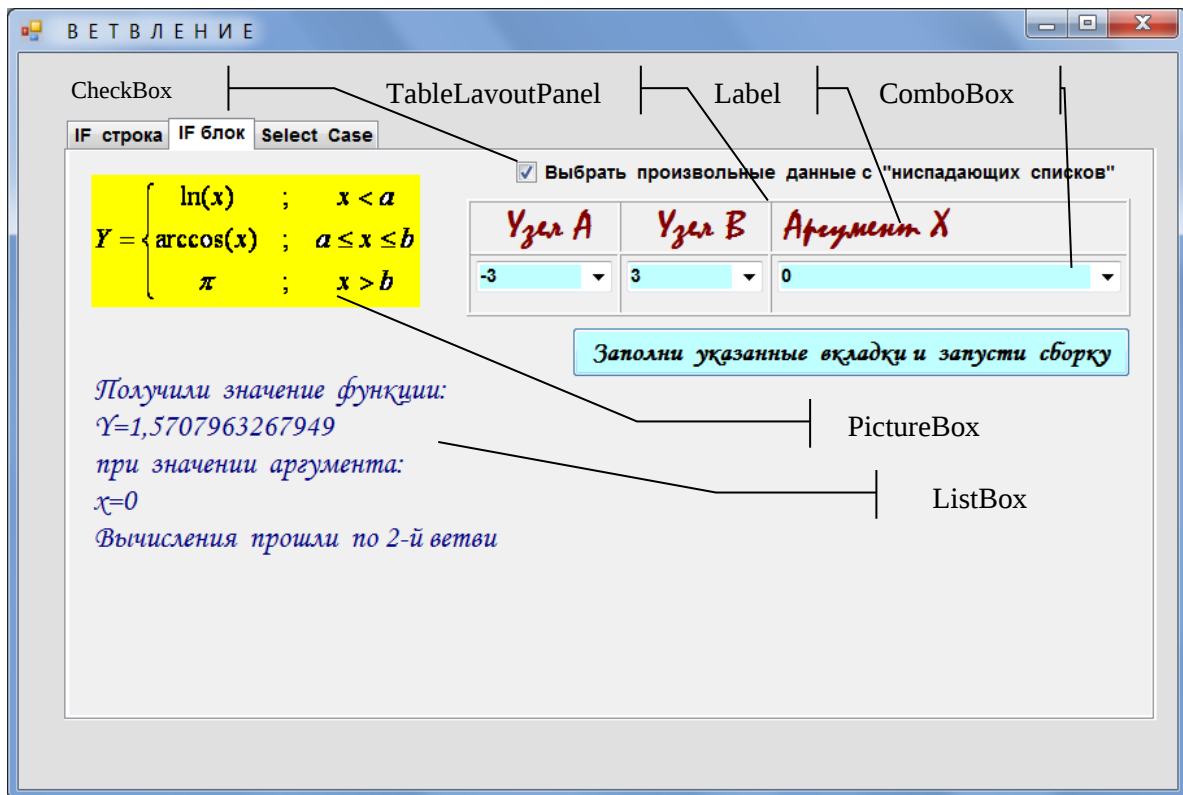


Рисунок 1.2 – Форма задачі «Алгоритм розгалуження 3 гілки»

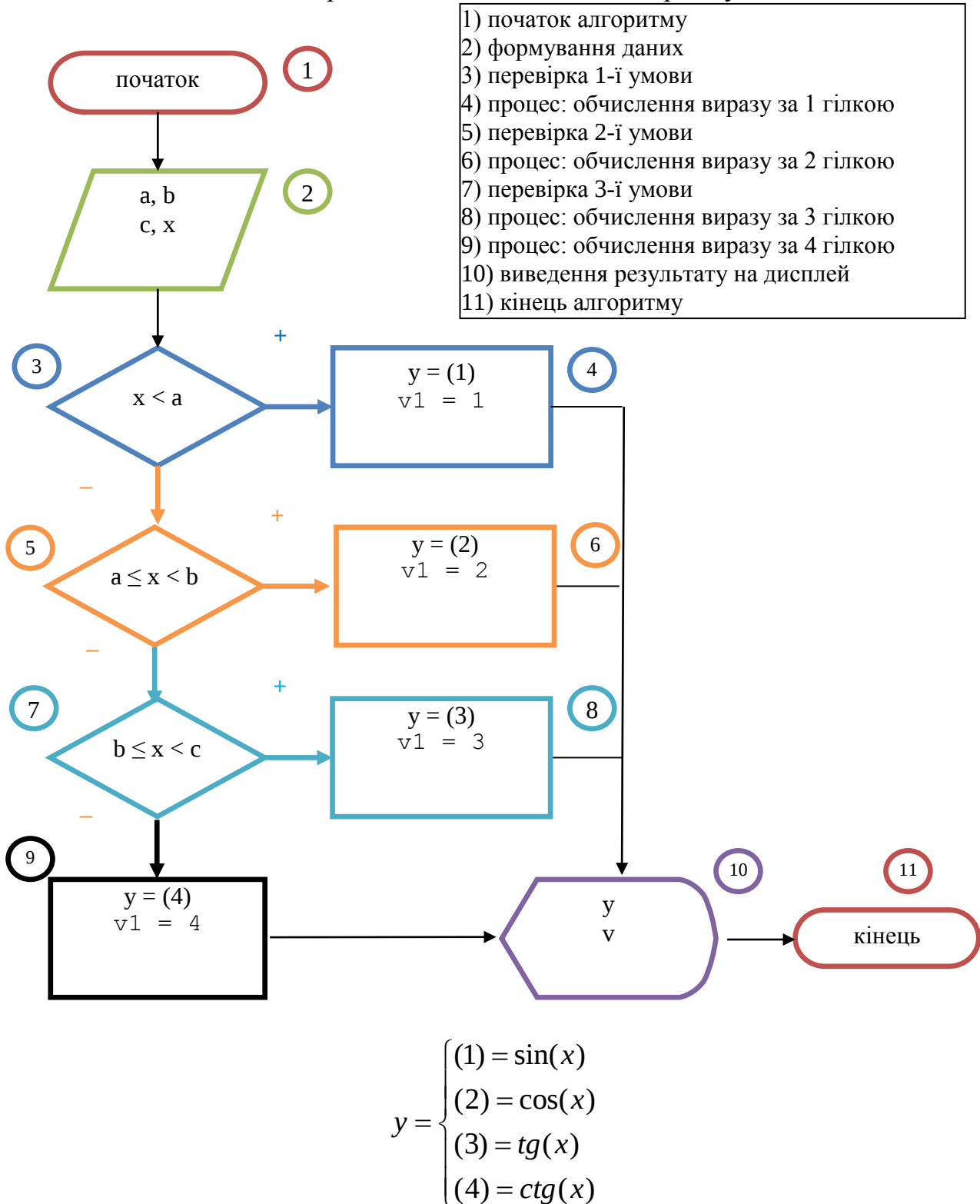
Розв'язок задачі «Алгоритм розгалуження 4 гілки» (Select Case)

Умова. Обчислити значення кусково-аналітичної функції 4 гілки з використанням конструкції (Select Case).

$$y = \begin{cases} \sin(x), & x < a \\ \cos(x), & a \leq x < b \\ tg(x), & b \leq x < c \\ ctg(x), & x \geq c \end{cases}$$

Стислі відомості про елементи управління, які призначені для створення інтерфейсу користувача наведені у додатку Б.

Принципова блок-схема алгоритму



Програмний код алгоритму

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button2.Click
```

```
Label10.Visible = False : Label11.Visible = False
```

```
Label12.Visible = False : Label13.Visible = False
```

```
Dim a As Double, b As Double, c As Double, v As Integer, x As Double
```

```
a = TextBox3.Text : b = TextBox4.Text : c = TextBox5.Text
```

```

x = InputBox("Введите значение аргумента", "Окно ввода", -5)
Select Case x
    Case Is < a
        y = Sin(x) : v = 1 : Label10.Text = "y=" & Round(y, 5) : Label10.Visible = True :
qqq = Label10
    Case a To b And x <> b
        y = Cos(x) : v = 2 : Label11.Text = "y=" & Round(y, 3) : Label11.Visible = True :
qqq = Label11
    Case b To c And x <> c
        y = Tan(x) : v = 3 : Label12.Text = "y=" & Round(y, 3) : Label12.Visible = True :
qqq = Label12
    Case Else
        y = 1 / Tan(x) : v = 4 : Label13.Text = "y=" & Round(y, 5) : Label13.Visible = True :
qqq = Label13
End Select
End Sub
#Region "МЕПЦАЮЩЕЕ ПОЛЕ"
Private Sub Form1_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles
Me.Load
    Timer1.Interval = 100
    Timer1.Enabled = True
    Timer2.Interval = 150
    Timer2.Enabled = True
    qqq = Label14
    '-----
    Button3.Visible = False
    'Layout-ПЕРЕВОД:Схема размещения
    TableLayoutPanel1.Visible = False
End Sub
Sub Timer1_Tick(ByVal sender As Object, ByVal e As System.EventArgs) Handles
Timer1.Tick
    With qqq
        .BackColor = Color.Blue
        .ForeColor = Color.Aqua
        .Font = New System.Drawing.Font("Arial", 8)
    End With
End Sub
Sub Timer2_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Timer2.Tick
    With qqq
        .BackColor = Color.Aqua
        .ForeColor = Color.Blue
        .Font = New System.Drawing.Font("Arial Black", 8)
    End With
End Sub
#End Region

```

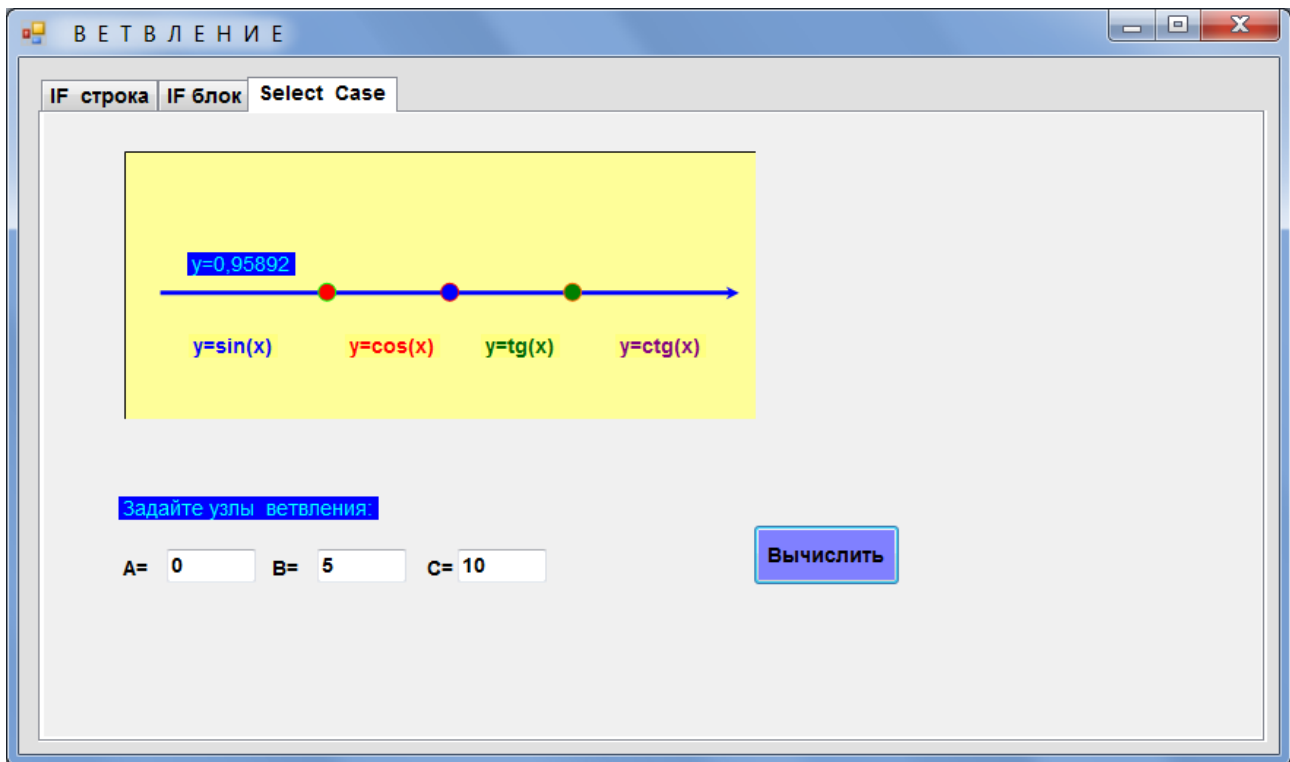


Рисунок 1.3 – Форма задачі «Алгоритм розгалуження 4 гілки»

2 Циклічні алгоритми у VB

Цикли без вкладення зі заздалегідь відомою кількістю повторень (For...Next)

Цикл For...Next використовується, коли наперед відоме початкове та кінцеве значення лічильника. Повторює виконання групи інструкцій зазначене число раз. Синтаксис запису циклу For...Next має вигляд:

```
For <лічильник> [As <тип даних>] = <початок> To <кінець> [Step <крок>]
    [інструкції]
[Exit For]
    [інструкції]
Next [лічильник]
```

лічильник обов'язковий. Числова змінна, використовувана в якості лічильника циклу;
початок обов'язковий. Початкове значення змінної лічильника;
кінець обов'язковий. Кінцеве значення змінної лічильника;
крок необов'язковий. Значення, на яке змінюється лічильник при кожному виконанні тіла циклу. Якщо це значення не задане, за замовченням крок дорівнює одиниці;
інструкції необов'язковий. Одна або декілька інструкцій між For і Next, що виконуються зазначене число раз.

Приклад використання циклу For...Next без вкладень:

```
Dim число As Integer
Dim сума As Integer
For число = 0 To 10
```

сумма = число + 1

Next число

MsgBox(число, MsgBoxStyle.Information, "Сообщение")

Цикли з вкладенням зі заздалегідь відомою кількістю повторень (For...Next)

Допускається організація вкладених циклів For... Next (один цикл For... Next розташовується в середині іншого). Лічильник кожного циклу повинен мати унікальне ім'я.

В наступному прикладі реалізований алгоритм формування та виведення на екран строк, які містять цифри від 0 до 9. Алгоритм заснований на застосуванні вкладених циклічних конструкцій For... Next.

Dim слова, цифра As Integer

Dim строка As String = ""

For слова = 10 To 1 Step -1

For цифра = 0 To 9

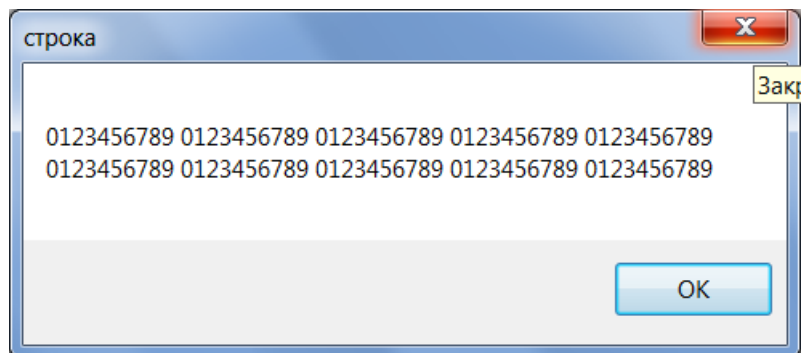
строка &= CStr(цифра)'

Next цифра

строка &= " "

Next слова

MsgBox(строка, , "строка")



Цикли з невідомою заздалегідь кількістю повторень (While ... End While)

Оператор While завжди перевіряє умову до виконання циклу. Виконання циклу продовжується доки значення умови залишається True (Істина). Якщо умова дорівнює False (Хибність) при першому вході в цикл, вона не виконається жодного разу. Синтаксис запису має вигляд:

While <умова>

[<інструкції>]

[Exit While]

[<інструкції>]

End While

Приклад використання циклу While ... End While. Алгоритм реалізує накопичення значень у змінній <счетчик>, та виведення на екран відповідно цьому інформації про кількість ітерацій.

```

Dim счетчик As Integer = 0
While счетчик < 20
    счетчик = счетчик + 1
End While
MsgBox("Цикл прошел " & CStr(счетчик) & "раз")

```

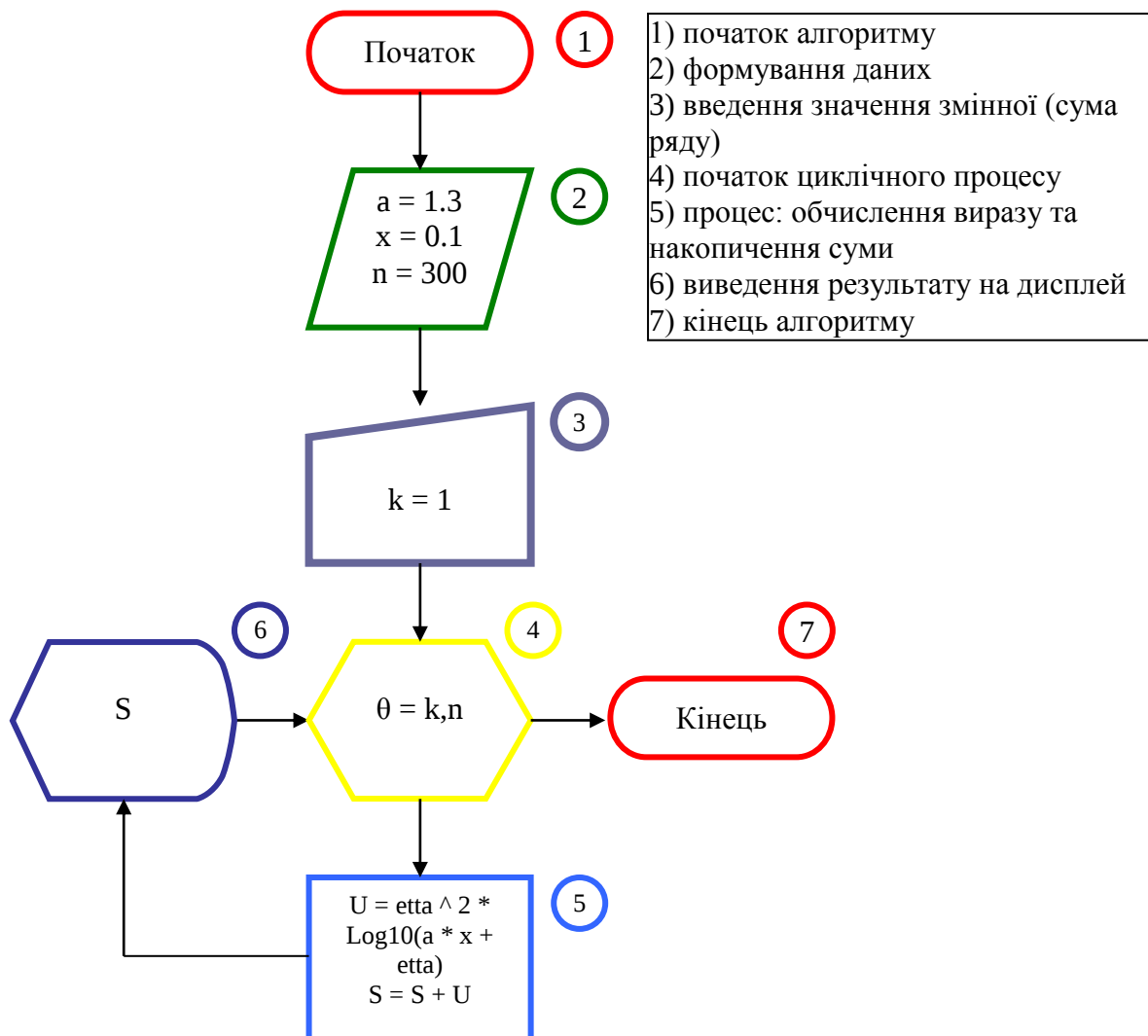
Розв'язок задачі «Одинарна сума / добуток числового ряду» (конструкція For...Next без вкладення)

Умова

$$\sum_{\theta=k}^n \theta^2 \lg(ax + \theta) \quad a = 1.3 \quad x = 0.1 \quad n = 300$$

Стислі відомості про елементи управління, які призначені для створення інтерфейсу користувача наведені у додатку Б.

Принципова блок-схема алгоритму



Програмний код алгоритму

```

Dim etta As Integer
Dim a As Double = 1.3

```

```

Dim x As Double = 0.1
Dim n As Integer = 300 ' смотри (.Maximum)
Dim k As Integer = 1, S As Double, U As Double
tbxInf0.Text = ""
tbxInf0.Visible = True
ProgressBar1.Visible = True
tbxInf0.ScrollBars = ScrollBars.Horizontal
tbxInf0.ScrollBars = ScrollBars.Vertical
tbxInf0.Multiline = True ' многоуровневое поле
' _____ Работаем с ProgressBar _____
With ProgressBar1
    .Maximum = n ' ВНИМАНИЕ:(.Maximum)=реальному кол-ву итераций
    .Minimum = 0
    .Step = 1
    .Value = 0

```

```
End With
```

```

k = InputBox("Введите начальное значение счетчика", "Определяет нижний
индекс суммирования", 1)

```

В наступному блоці реалізується циклічний алгоритм без вкладення, обчислення членів ряду U , накопичення їх суми, та виведення результату в текстове поле:

```

For etta = k To n
    ProgressBar1.PerformStep() 'Step-Приращение, на которое каждый вызов
    'метода PerformStep увеличивает текущее положение индикатора выполнения
    tbxInf0.Text = tbxInf0.Text & vbNewLine & "№" & etta & ": U=" & U & " S=" & S
    U = etta ^ 2 * Log10(a * x + etta)
    S = S + U
Next

```

В наступному блоці реалізується функціональність елементу управління ColorDialog, який задає колір заливання та шрифту текстового поля tbxInf0:

```

If cbxBGGround.Checked = True Then
    If ColorDialog1.ShowDialog() = DialogResult.OK Then
        tbxInf0.BackColor = ColorDialog1.Color
    End If
End If
' _____ Работа с ColorDialog1 _____
If cbxColorFont.Checked = True Then
    If ColorDialog1.ShowDialog() = DialogResult.OK Then
        tbxInf0.ForeColor = ColorDialog1.Color
    End If
End If

```

Тестування алгоритму

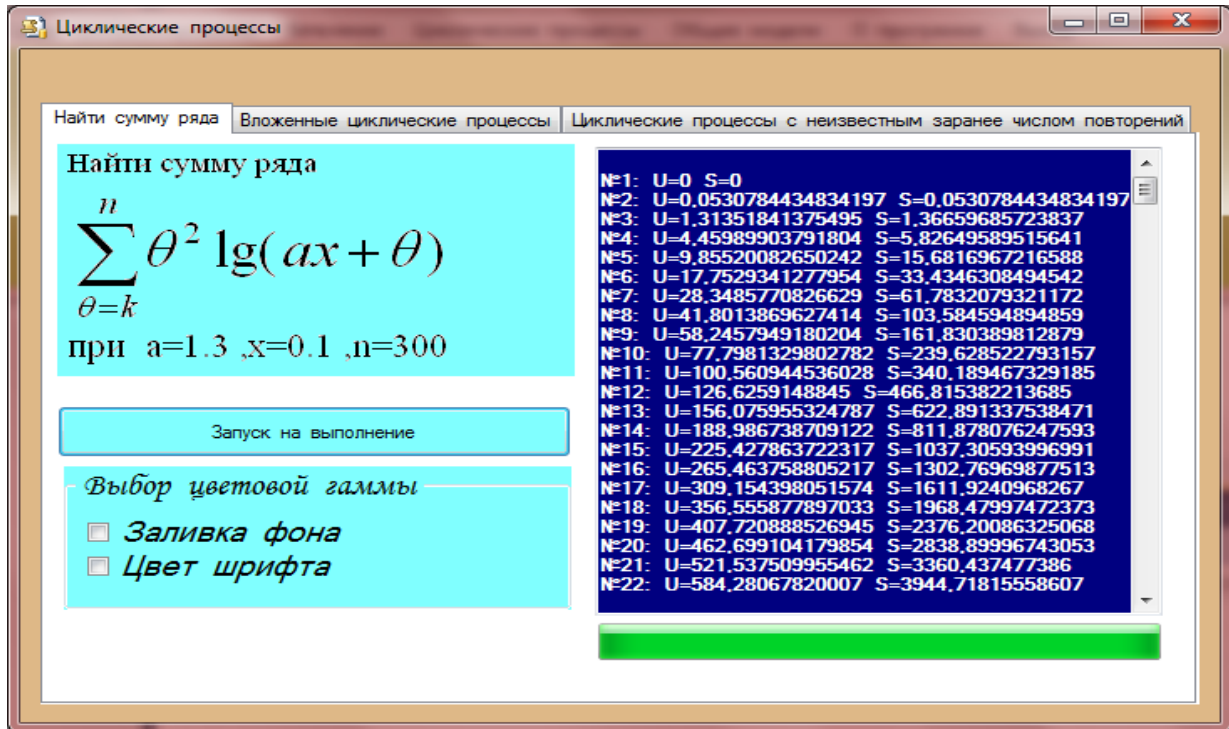


Рисунок 2.1 – Форма задачі «Пошук суми кінцевого ряду»

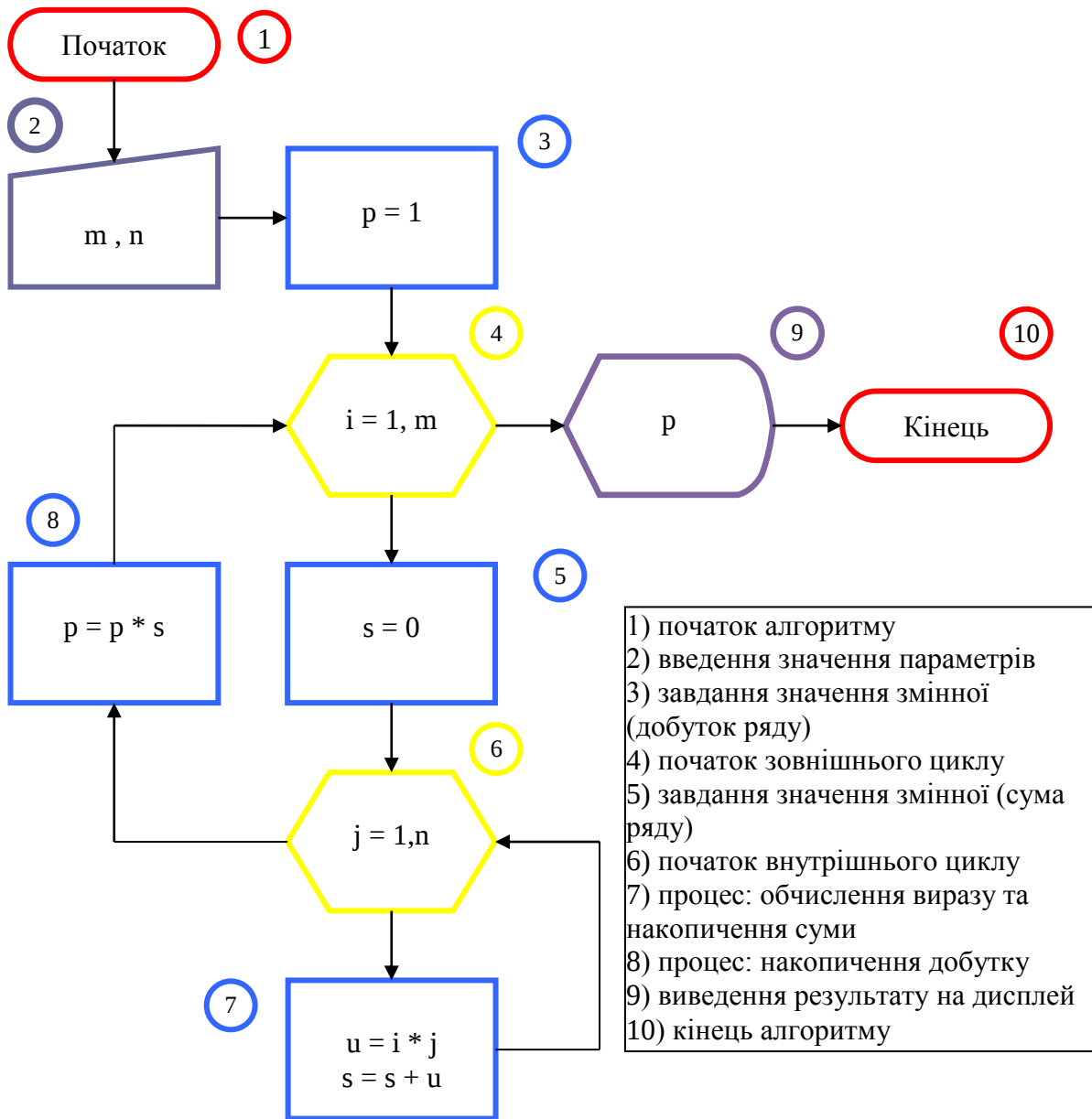
Розв'язок задачі «Подвійна сума / добуток числового ряду» (конструкція For...Next з вкладенням)

Умова

$$\prod_{i=1}^2 \sum_{j=1}^3 ij$$

Стислі відомості про елементи управління, які призначені для створення інтерфейсу користувача наведені у додатку Б.

Принципова блок-схема алгоритму



Програмний код алгоритму

```

Dim i As Integer, m As Integer = tbxIn.Text, n As Integer = tbxJn.Text
Dim j As Integer, u As Double, s As Double, p As Double = 1
With tbxInf1
    .ScrollBars = ScrollBars.Both
    .Multiline = True
    .Font = New System.Drawing.Font("Times New Roman", 14)
    .Text = "Тест внутрішнього цикла:"
End With
With tbxInf2
    .Text = "Тест зовнішнього цикла:"
    .ScrollBars = ScrollBars.Both
    .Multiline = True
  
```

.Font = New System.Drawing.Font("Times New Roman", 14)

End With

В наступному блоці реалізується циклічний алгоритм із вкладенням. У внутрішньому циклі (за змінної j) організовано обчислення членів ряду U , накопичення їх суми, та виведення результату в текстове поле. У зовнішньому циклі (за змінної i) організовано накопичення добутку суми членів ряду, та виведення результату в текстове поле:

```
For i = 1 To m : s = 0
```

```
  For j = 1 To n
```

```
    u = i * j : s = s + u
```

```
    tbxInf1.Text = tbxInf1.Text & vbNewLine & "U(" & j & ")=" & u
```

```
  Next
```

```
  tbxInf1.Text = tbxInf1.Text & vbNewLine & "_____"
```

```
  tbxInf1.Text = tbxInf1.Text & vbNewLine & " ЧАСТИЧНАЯ СУММА: S(" & i & "
```

```
  p = p * s
```

```
  tbxInf2.Text = tbxInf2.Text & vbNewLine & "S(" & i & ")=" & s
```

```
Next
```

```
tbxInf2.Text = tbxInf2.Text & vbNewLine & "_____"
```

```
tbxInf2.Text = tbxInf2.Text & vbNewLine & "P( ИТОГОВОЕ )=" & p
```

```
If cbxFont.Checked = True Then
```

```
  If FontDialog1.ShowDialog = Windows.Forms.DialogResult.OK Then _
```

```
    tbxInf1.Font = FontDialog1.Font
```

```
    tbxInf2.Font = tbxInf1.Font
```

```
End If
```

Тестування алгоритму

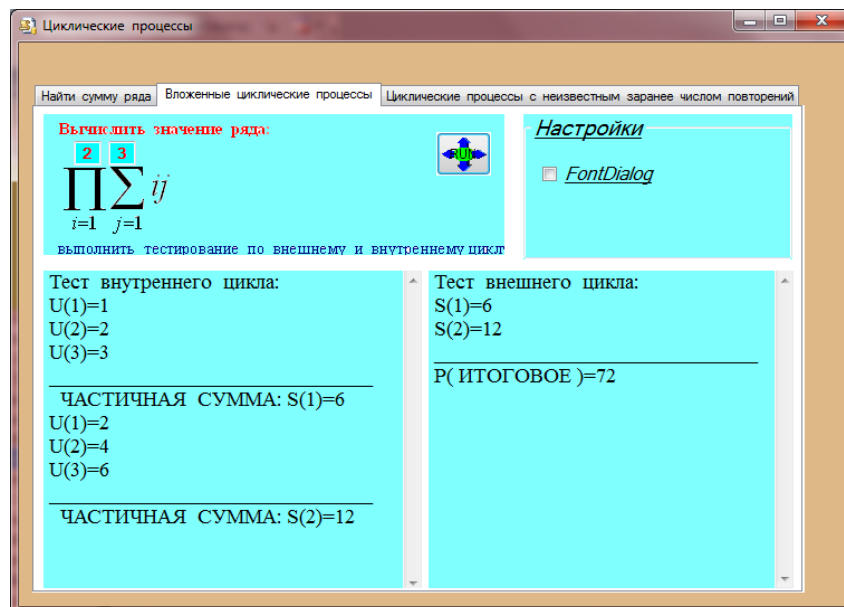


Рисунок 2.2 – Форма задачі «Пошук добутку суми кінцевого ряду»