



РОССИЙСКАЯ ФЕДЕРАЦИЯ
МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ
ГОУ ВПО ТЮМЕНСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНСТИТУТ ДИСТАНЦИОННОГО ОБРАЗОВАНИЯ
ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ НАУК

И. Н. ГЛУХИХ, Д. В. СЕМИХИН

КОМПЬЮТЕРНЫЕ ПРОДУКТЫ

Учебное пособие



Издательство
Тюменского государственного университета
2008

УДК 004.9(075.8)
ББК 3973.26-018.2я73
Г 554

И. Г. Глухих, Д. В. Семихин. КОМПЬЮТЕРНЫЕ ПРОДУКТЫ: Учебное пособие. Тюмень: Издательство Тюменского государственного университета, 2008. 112 с.

Дан общий обзор представленных на рынке продуктов компьютерной индустрии. Коротко описаны история и классификация персональных компьютеров, рассмотрены компоненты современных вычислительных машин и их характеристики; общие методы и средства для разработки программного обеспечения и наиболее значимые промышленные продукты для решения финансово-экономических задач.

Методические материалы, задания и вопросы теста для итогового контроля предназначены повышать качество полученных знаний и закрепить их.

Адресовано студентам, обучающимся по специальностям «Экономика», «Экономика и управление на предприятии», «Национальная экономика», «Бухгалтерский учет, анализ и аудит» на всех формах обучения, в том числе с применением дистанционных технологий.

Одобрено на заседании кафедры информационных систем ТюмГУ.

Рекомендовано к печати Учебно-методической комиссией Института математики и компьютерных наук ТюмГУ.

Рецензенты: **А. Г. Ивашко**, док. тех. наук, доцент, зав. кафедрой информационных систем ТюмГУ

А. В. Пестова, канд. экон. наук, доцент кафедры менеджмента в отраслях ТЭК ТюмГНГУ

Ответственный
за выпуск:

А. В. Трофимова, зав. отделом учебно-методического обеспечения ИДО ТюмГУ

ISBN 978-5-88081-928-7

© ГОУ ВПО Тюменский государственный университет, 2008

ОГЛАВЛЕНИЕ

ПРЕДИСЛОВИЕ	4
МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ	
РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ	5
Пояснительная записка	5
Содержание дисциплины	6
РЕКОМЕНДАЦИИ ПО САМОСТОЯТЕЛЬНОЙ РАБОТЕ СТУДЕНТА	8
Календарно-тематический план	8
Методические рекомендации по отдельным видам самостоятельной работы	9
ТЕОРЕТИЧЕСКИЕ МАТЕРИАЛЫ	
ВВЕДЕНИЕ	11
РАЗДЕЛ 1. АППАРАТНОЕ ОБЕСПЕЧЕНИЕ ПК	
Глава 1. Классификация компьютеров	18
Глава 2. Корпус и системная плата персонального компьютера	20
Глава 3. Процессоры	25
Глава 4. Оперативная память	29
Глава 5. Накопители	33
Глава 6. Подключение внешних устройств	43
Резюме	48
Вопросы для самопроверки	50
РАЗДЕЛ 2. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ПК	
Глава 1. Определение и классификация программных продуктов	51
Глава 2. Стратегия разработки ПО	53
Глава 3. Технология быстрой разработки приложений — RAD	62
Глава 4. Элементы языка VBA	67
Глава 5. Практика использования VBA — объекты MS Excel	73
Глава 6. Программное обеспечение для решения финансово-экономических задач	75
Резюме	84
Вопросы для самопроверки	85
ЗАКЛЮЧЕНИЕ	86
ПРАКТИКУМ	87
ЗАДАНИЯ ДЛЯ КОНТРОЛЯ	90
Тесты для самоконтроля	90
Ключи к тестам для самоконтроля	97
Задания для контрольных работ	98
Вопросы для подготовки к зачету	103
ГЛОССАРИЙ	104
СПИСОК ЛИТЕРАТУРЫ	112

ПРЕДИСЛОВИЕ

Наблюдаемое во всех сферах человеческой деятельности изменение количества и качества информации обуславливает необходимость ее оперативного и многомерного анализа и, в этой связи, выдвигает новые требования к уровню знаний в области информационных технологий. Для современного специалиста умение результативно использовать прогрессивное техническое и программное обеспечение и информационные технологии (ИТ) в своей работе становится неотъемлемым показателем его квалификации.

Изучение материалов в рамках учебной дисциплины «Компьютерные продукты» нацелено на реализацию государственных требований к уровню подготовки выпускников экономических специальностей.

Данный курс формирует базовые знания, умения и навыки в аспекте применения компьютерных продуктов. В качестве таковых изначально рассматриваются продукты компьютерной индустрии — как техническое, так и программное обеспечение вычислительных систем.

Содержание курса отражает современное (на момент написания учебного пособия) состояние развития информационных технологий и основано на многолетнем опыте преподавания дисциплин компьютерной направленности авторами курса — Игорем Николаевичем Глухих, профессором, доктором технических наук и Дмитрием Витальевичем Семихиным, кандидатом физико-математических наук, доцентом кафедры информационных систем ТюмГУ.

МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Пояснительная записка

Предмет «Компьютерные продукты» входит в состав естественно-научных дисциплин, изучаемых на экономических специальностях ТюмГУ. В процессе знакомства с курсом лекций и решения задач, входящих в практикум, закрепляется и усиливается обучающий эффект от знаний, полученных студентами при изучении «Информатики», «Микроэкономики» и др., а также создается необходимая база для последующего изучения таких дисциплин, как «Информационные технологии в экономике», «Интернет-технологии».

Цель данного курса — дать структурированное представление о том, что предоставляет потребителю современная компьютерная индустрия.

Решаются такие **задачи**, как подготовка специалистов, которые могут использовать компьютерные системы для решения профессиональных задач, участвовать в проектах, связанных с выбором аппаратно-технического обеспечения, разработкой программных продуктов и оказания информационных услуг.

Изучение дисциплины призвано обеспечить приобретение таких **профессиональных компетенций**, как:

- знание архитектуры современного персонального компьютера, его основных компонентов, их характеристик и функциональных возможностей;
- знание этапов, основных процессов и подходов к разработке программных продуктов;
- возможность обоснованного выбора состава аппаратного обеспечения для решения профессиональных задач.

К числу **инструментальных компетенций** можно отнести навыки применения программных средств для решения экономических задач, использование современных технологий разработки приложений в среде Visual Basic for Application для автоматизации работы с офисными программами.

Содержание дисциплины

Тематический план

Шифр раздела, темы	Наименование разделов и тем	Распределение часов			
		всего	лекции	практ.	самост. работа
1	Введение				
1.1	Аппаратное обеспечение ПК				
1.2	Классификация персональных компьютеров				
1.3	Корпус и системная плата персонального компьютера	52	2	2	48
1.4	Процессоры				
1.5	Оперативная память				
1.6	Накопители				
	Подключение внешних устройств				
2	Программное обеспечение ПК				
2.1	Определение и классификация программных продуктов				
2.2	Стратегия разработки ПО				
2.3	Технологии быстрой разработки приложений – RAD	51	2	4	45
2.4	Элементы языка VBA				
2.5	Практика использования VBA – объекты MS Excel				
2.6	Программное обеспечение для решения финансово-экономических задач				
	Всего по дисциплине	103	4	6	93

Содержание лекционного курса

Шифр раздела	Номер лекции	Содержание лекции	Кол-во часов
1	1	Аппаратное обеспечение ПК. История появления персонального компьютера. Понятие компьютерного продукта. Классификации персональных компьютеров и мобильных устройств. Корпус и системная плата персонального компьютера. Процессоры и их характеристики. Назначение, виды и характеристики оперативной памяти; модули оперативной памяти. Происхождение, состав, характеристики и интерфейсы жестких дисков, съемные накопители. Оптические накопители – история, характеристики; перспективы развития. Флэш-память. Шины и платы расширения; порты ввода-вывода; внешние устройства – видеокарты и модемы.	2

Шифр раздела	Номер лекции	Содержание лекции	Кол-во часов
2	2	Программное обеспечение ПК. Понятия «программное обеспечение» и «программный продукт». Классификация и жизненный цикл программного обеспечения. Процессы и модели жизненного цикла. CASE-средства разработки приложений. Описание технологии RAD и ее применения к офисному программированию. Основные понятия языка VBA – переменные, константы и операторы. Основные понятия теории объектно-ориентированного программирования – объект, класс, метод, свойство. Объекты MS Excel. Промышленные программные продукты – назначение и функциональные возможности.	2
		Всего по дисциплине	4

Перечень практических занятий

Шифр раздела	Номера и темы практических занятий	Кол-во часов	Формы контроля
1	Изучение состава и характеристик аппаратной части ПК	2	Тест
2	Применение VBA к решению экономических задач	4	КР
	Всего по дисциплине	6	



РЕКОМЕНДАЦИИ ПО САМОСТОЯТЕЛЬНОЙ РАБОТЕ СТУДЕНТА

Календарно-тематический план

№	Название раздела	Время, отводимое на изучение	Виды учебной работы, рекомендуемое время на выполнение		Форма контроля	Сроки представления заданий на проверку
1	Аппаратное обеспечение ПК	52 часа	Изучение теоретических материалов	46	Тестовые задания	В течение семестра
			Ответы на вопросы для самопроверки	1		
			Выполнение практических заданий	4		
			Промежуточное тестирование	1		
			Всего	52		
2	Программное обеспечение ПК	51 час	Изучение теоретических материалов	44	Контрольная работа, тестовые задания	За месяц до сессии
			Ответы на вопросы для самопроверки	1		
			Выполнение контрольной работы	6		
			Промежуточное тестирование	1		
			Всего	52		

Методические рекомендации по отдельным видам самостоятельной работы

Указания по самостоятельному изучению теоретической части дисциплины

Рекомендуется последовательное освоение материала в том порядке, в котором он изложен. Тем не менее, материал разделов 1 и 2 не перекликается друг с другом и может изучаться независимо.

При изучении раздела 1 следует обратить внимание на то, что в силу динамичности рынка компьютерных продуктов представленная там информация может устареть. Поэтому настоятельно рекомендуется параллельно с изучением теоретических материалов, изложенных в пособии, обратиться к интернет-ресурсам, посвященным обзорам новинок компьютерной индустрии (таким, как **www.ferra.ru**, например).

Выполнение практической работы по первому разделу предполагает загрузку и установку на компьютер обучающегося специализированного программного обеспечения. В силу каких-либо затруднений с получением и использованием указанного в пособии приложения *SiSoftware Sandra* его можно (хотя и крайне нежелательно) заменить другим программным продуктом родственного назначения. Например, для выполнения работы можно использовать утилиту *CPU-Z*, также рассмотренную в пособии.

Материал, изложенный во втором разделе, более статичен, и потому не требует непременно изучения дополнительных материалов. Следует, однако, обратить особое внимание на материал глав 4 и 5, в которых излагаются некоторые аспекты использования программирования на языке VBA. Знакомство с ними будет необходимо при выполнении контрольной работы. Также понадобится знание такого инструмента MS Excel, как *Поиск* решения, который традиционно рассматривается при изучении дисциплин «Компьютерные технологии» и «Информатика».

Указания по выполнению контрольной работы

Указания по выполнению контрольных работ приведены в «Заданиях для контроля».

Указания к промежуточной аттестации с применением балльно-рейтинговой системы оценки знаний студента

Итоговая аттестация по дисциплине — зачет. Она осуществляется на основе результатов выполнения:

- практической работы;
- контрольной работы;
- итогового тестирования.

Качество выполнения каждого рода работ оценивается в баллах. При правильном выполнении работы за нее выставляется максимальный балл.

Для практической работы он составляет 20, для контрольной работы — 40 баллов. Итоговое тестирование можно оценить по следующей шкале:

Процент правильных ответов	Кол-во баллов
Менее 50%	0
50% – 59%	5
60% – 69%	10
70% – 79%	20
80% – 89%	30
90% – 100%	40

Итоговую аттестацию можно считать успешной, если по совокупности всех работ обучающимся набрано более 75 баллов.

ТЕОРЕТИЧЕСКИЕ МАТЕРИАЛЫ

ВВЕДЕНИЕ

Развитие науки и техники XX в. ознаменовалось многими выдающимися достижениями: массовой автомобилизацией, развитием авиации, радио, телевидением, использованием энергии атома, полетом человека в космос, появлением Интернета и многим другим. Среди этих событий появление и последующее стремительное развитие компьютерной индустрии в последние десятилетия XX столетия занимает весьма заметное, если не первое, место.

Темпы роста новой отрасли промышленности были связаны с появлением на рынке принципиально нового продукта — персонального компьютера.

Начало было положено в 70 гг. XX в., когда идея персональной ЭВМ — машины, принадлежащей одному человеку, а не организации — витала в воздухе. Возможно, главным побудительным мотивом к этому была та парадоксальная смесь восхищения и досады, которую вызывали у людей большие вычислительные машины первых поколений. Такие электронные гиганты могли творить чудеса, но из-за своей огромной стоимости и технической сложности работали лишь в строго определенных условиях и были доступны лишь немногим избранным. Взаимодействие с ЭВМ строилось через посредников — операторов. Положение не изменилось и после появления и широкого распространения мини-ЭВМ, которые нельзя отнести к персональным компьютерам из-за их стоимости, — они были по карману только очень богатым персонам.

В настоящее время существует несколько противоречивых мнений, что же следует считать первым персональным компьютером (ПК). Наиболее часто пальму первенства отдают микрокомпьютеру «Альтаир-8800», представленному широкой публике в 1975 г. (рис. 1). Мик-

рокомпьютер продавался по почте в виде набора деталей для самостоятельной сборки крошечной компанией Micro Instrumentation and Telemetry Systems (MITS) из г. Альбукерке, штат Нью-Мексико. Возглавлял компанию отставной офицер ВВС США, энтузиаст электроники Эд Робертс. Название «Альтаир» придумала дочь Робертса (так называлась звезда из популярного телесериала «Звездный поход»).



Рис. 1. Персональный компьютер Altair 8800

Весь комплект для сборки стоил 397 долларов, что было лишь на 17 долларов больше той цены, которую выставяла Intel только за один микропроцессор (MITS покупала чипы у Intel по себестоимости). Заказ на первые 200 комплектов был получен по телефону уже в течение дня — настолько людям хотелось иметь собственную вычислительную машину. «Альтаир-8800» имел ошеломляющий коммерческий успех.

И это притом, что пользователям приходилось самостоятельно паять и тестировать собранные узлы, а если сборка завершалась успешно, они становились программистами и создавали программы для своего ПК на машинном языке, то есть с помощью нулей и единиц. В компьютере не было ни клавиатуры, ни дисплея, ни долговременной памяти. Весь объем оперативной памяти (ОЗУ) составлял 256 Байт. Программы вводились переключением тумблеров на передней панели, а результаты считывались со светодиодных индикаторов.

Конечно, 8080 был настоящим процессором и позволял большее. Altair был оснащен *шиной расширения*, и новые устройства, расширяющие возможности системы, не замедлили появиться. Появились

платы расширения памяти, телетайпный ввод, устройства работы с перфолентой и т. д. Однако писать программы с помощью нулей и единиц могли только большие энтузиасты. Чтобы компьютер оказался действительно полезным многим людям, требовался язык более высокого уровня. Такой язык — Basic был создан в 1975 г. двумя приятелями — Биллом Гейтсом и Полом Алленом (рис. 2), которые, будучи еще студентами (один Вашингтонского, а другой — Гарвардского) университетов, в том же году организовали фирму «Microsoft» (тогда она еще называлась через дефис).

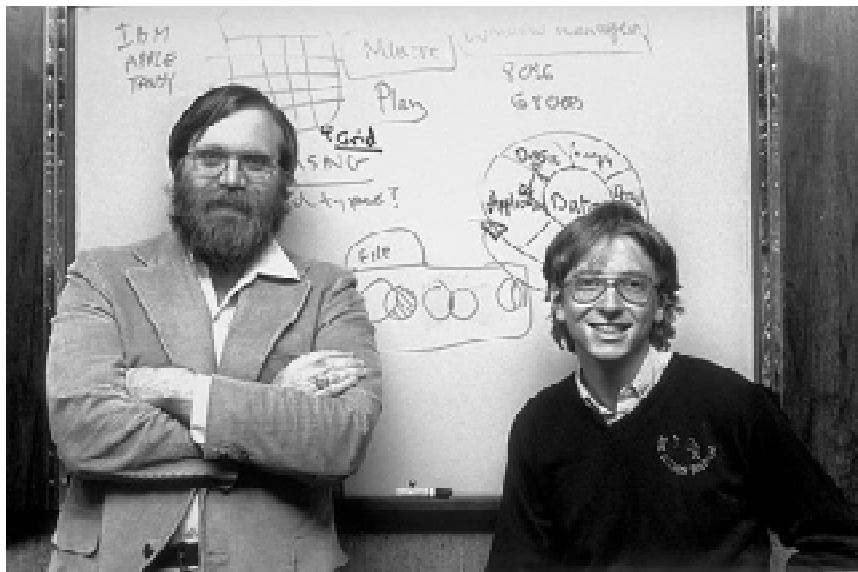


Рис. 2. Основатели Microsoft — Билл Гейтс и Пол Аллен (1982 г.)

В 1976 г. 26-летний инженер Стив Возняк создал принципиально новый микрокомпьютер. Он впервые применил для ввода данных клавиатуру, подобную клавиатуре пишущей машинки, а для отображения информации — обыкновенный телевизор. Символы выводились на его экран в 24 строки по 40 символов в каждой. Компьютер имел 8 кБайт памяти, половину из которых занимал встроенный язык Basic, а половину пользователь мог использовать для введения своих программ. Этот компьютер значительно превосходил Altair-8800 по целому ряду параметров. Возняк предусмотрел для своего нового ком-

пьютера разъем (так называемый «слот») для подсоединения дополнительных устройств. Первым понял и оценил перспективы этого компьютера приятель Стива Возняка — Стив Джобс. Он предложил организовать фирму для его серийного изготовления. 1 апреля 1976 г. они основали компанию Apple и в январе 1977 г. официально зарегистрировали ее. Новый компьютер они назвали Apple-I. В течение 10 месяцев им удалось собрать и продать около 200 экземпляров Apple-I. В силу малости серии, этот компьютер не стал достаточно известным. Популярность продукция фирмы Apple приобрела лишь благодаря ПК Apple-II (рис. 3), массовое производство которого началось в 1977 г. Эта вычислительная машина стала родоначальником популярных сейчас «Макинтошей». Компьютер был выполнен в пластмассовом корпусе, получил графический режим, звук, цвет, расширенную память, 8 разъемов расширения (слотов). Для сохранения программ в нем использовался кассетный магнитофон. Основу первой модели Apple II составлял, как и гораздо менее известного Apple I, микропроцессор 6502 фирмы MOS Technology с тактовой частотой 1 МГц.

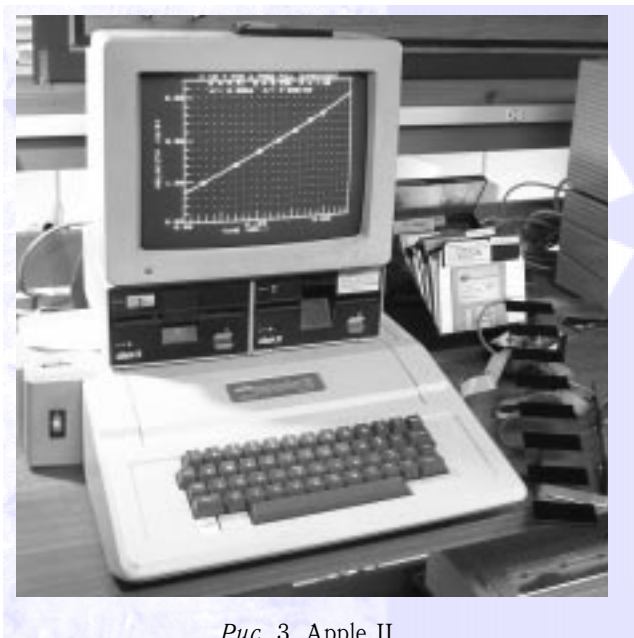


Рис. 3. Apple II

Информация выводилась на цветной или черно-белый телевизор, работающий в стандартной для США системе NTSC. В текстовом режиме отображались 24 строки, по 40 символов в каждой, а в графическом — разрешение составляло 280 на 192 точки (шесть цветов). Основное достоинство Apple II заключалось в возможности расширения его оперативной памяти до 48 кБайт и использования 8 разъемов для подключения дополнительных устройств. Благодаря использованию цветной графики его можно было использовать для самых различных игр.

Что касается Европы, то можно вспомнить ПК ZX-Spectrum на процессоре Z-80, появившийся в 1981 г. Эта конструкция быстро завоевала всю Западную Европу. Считается, что для ZX-Spectrum написано более сотни тысяч программ. Автор конструкции — британец Клайв Синклер, основатель и владелец Sinclair Research Ltd, — получил дворянское звание за вклад в компьютеризацию Соединенного Королевства. ZX-Spectrum получил второе дыхание на рубеже 80-90 гг. В эти годы к процессу компьютеризации подключилась Восточная Европа, включая Россию. Было разработано огромное количество разных вариантов Sinclair — совместимых ПК на местной элементной базе. Через эту машину многие тысячи молодых людей приобщились к вычислительной технике.

В 1981 г. случилось воистину эпохальное событие — на рынок ПК вышла фирма IBM, представив первую модель IBM PC, получившую индекс IBM 5150 (рис. 4). Позже многие назовут этот компьютер «отцом» всех современных ПК.

Выход на арену IBM означал конец «самодеятельности», неформального, любительского стиля работы, характерного для первого этапа развития индустрии ПК. Предложенная фирмой конструкция персонального компьютера имела ряд черт, определивших ее ошеломляющий успех.

Прежде всего — это применение нового, перспективного и весьма дорогого по тем временам процессора Intel 8088. Фирма IBM не пошла на то, чтобы использовать популярные на тот момент восьмибитные процессоры (таков процессор Z-80), которые были уже близки к своему закату. Процессор Intel 8088 имел внутренние регистры, рассчитанные на 16 разрядов двоичного числа (это уже шестнадцатитбитная архитектура).



Рис. 4. Персональный компьютер IBM PC 5150

Во-вторых — это применение принципа открытой архитектуры. Данный принцип означает применение в конструкции при сборке компьютера готовых блоков и устройств, а также стандартизацию способов соединения компьютерных устройств. Сразу же появилось немало сторонних фирм, производящих карты расширения.

В-третьих — это то обстоятельство, что IBM никак не закрепила свои авторские права на разработанную архитектуру. Были запатентованы только коды BIOS (Basic Input Output System или «Базовая Система Ввода-Вывода» — с этой программы начинается запуск ПК). Поэтому у других фирм появилась возможность подключиться к производству ПК с такой архитектурой, требовалось лишь написать свои коды BIOS. Так появилось понятие IBM-совместимый ПК. В конце 1990-х гг. IBM PC-совместимые компьютеры составили 90% рынка персональных компьютеров.

Так, благодаря персональным компьютерам, появилась новая отрасль промышленности, темпы развития которой оказались невиданными. Объем производства в этой отрасли очень быстро стал составлять в ценовом выражении десятки миллиардов долларов. При этом будет неверным считать, что компьютерная индустрия — это лишь производство комплектующих для современных ПК. Без программного обеспечения (ПО) даже самая удачная конструкция ПК обречена на забвение. И потому создание программного обеспечения для

современных «персоналок» давно уже перестало быть уделом кустарей-одиночек и встало на промышленные рельсы.

В самом общем смысле, под компьютерными продуктами понимаются аппаратные и программные средства компьютера как объекты производства и коммерческой деятельности.

Так, например, по результатам опроса журнала «Мир ПК» в 2006-м г. лучшими компьютерными продуктами стали: настольные компьютеры HP Compaq dc5100 и антивирусный пакет «Антивирус Касперского 5»; процессоры AMD Athlon X2 4800+ и программа для записи CD-DVD Nero AG Nero Burning ROM 7.x; видеоадаптер Sapphire Radeon X1900XTX 512 Mb и правовая информационно-справочная система «КонсультантПлюс».

В настоящем курсе лекций последовательно рассматриваются оба типа компьютерных продуктов — и аппаратное, и программное обеспечение. В первом разделе рассмотрены классификация и основные компоненты архитектуры персонального компьютера и приведены их основные характеристики, во втором — программные продукты. Даны возможные классификации программного обеспечения, описаны современные подходы к проектированию и разработке ПО. В рамках повествования о технологиях разработки будут изучены методы применения инструментальной среды программирования для создания приложений на примере Visual Basic for Applications. Также приведены оценки некоторых экономических программных продуктов, используемых для автоматизации экономической деятельности предприятия. Для обеспечения усвоения материала ссылки на используемую литературу по тексту не приводятся. Используемые при составлении источники приведены в списке литературы.

РАЗДЕЛ 1. АППАРАТНОЕ ОБЕСПЕЧЕНИЕ ПК

Глава 1. Классификация компьютеров

Быстрый прогресс в развитии микроэлектронных технологий и большое разнообразие создаваемой вычислительной техники затрудняло и затрудняет введение четкой градации между компьютерами. По мере совершенствования структур и технологий производства появляются новые классы компьютеров, границы существующих классов существенно изменяются. Тем не менее, в настоящее время существует несколько устоявшихся способов классифицировать вычислительные машины.

Во-первых, к таковым можно отнести классификацию *по этапам развития* (по поколениям). Идея делить машины на поколения вызвана к жизни тем, что за время короткой истории своего развития компьютерная техника проделала большую эволюцию как в смысле элементной базы (лампы, транзисторы, микросхемы и др.), так и в смысле изменения ее структуры, появления новых возможностей, расширения областей применения и характера использования.

Всего в эволюции компьютеров выделяют пять этапов:

— 1-е поколение (1955-1960 гг.) — ЭВМ на электронных вакуумных лампах;

— 2-е поколение (1960-1965 гг.) — ЭВМ на дискретных полупроводниковых приборах (транзисторах);

— 3-е поколение (1965-1970 гг.) — переход к интегральным микросхемам с малой степенью интеграции (порядка десяти транзисторов на одном кристалле);

— 4-е поколение (1970-1990 гг.) — переход к интегральным микросхемам с большой и сверхбольшой степенью интеграции (порядка 100 тысяч транзисторов на одном кристалле);

— 5-е поколение (1990-...) — переход к архитектуре с множеством параллельно работающих процессоров.

Другая классификация вычислительных машин характеризует область их применения. Также можно встретить название *«по размерам и вычислительной мощности»*. Согласно этой классификации, компьютеры можно разделить на следующие классы:

— суперкомпьютеры — сверхбыстродействующие вычислительные комплексы, использующие десятки тысяч процессоров;

— мэйнфреймы (или большие ЭВМ) — высокопроизводительные компьютеры, содержащие один или несколько центральных и периферийных процессоров с общей памятью, связанных между собой высокоскоростными магистралями передачи данных;

— рабочие станции — это компьютеры, число процессоров у которых варьируется от одного до четырех, с многопользовательскими операционными системами семейства ОС UNIX, обслуживающими вычислительные потребности небольшой группы пользователей;

— персональные компьютеры — это, как правило, однопроцессорные системы на платформе Intel или AMD, работающие под управлением однопользовательских операционных систем (Microsoft Windows и др.).

Современное развитие компьютерных технологий дало жизнь еще одному направлению компьютерной индустрии — на рынке появилось огромное разнообразие портативных персональных компьютеров, зачастую мало уступающих по производительности настольным ПК.

К числу мобильных систем относят следующие классы устройств:

1) ноутбуки (Note Book) — весом в несколько килограмм и размером экрана от 10 до 20 дюймов, под управлением обычной операционной системы (MS Windows и др.) эти компьютеры могут с успехом выполнять роль домашнего, офисного или игрового компьютера;

2) планшетные ПК (Tablet PC) — особый класс ноутбуков, оснащаемых сенсорным экраном и соответствующей операционной системой (MS Windows Tablet Edition, например), позволяющей осуществлять ввод данных прикосновениями к экрану;

3) устройства UMPC (Ultra-Mobile PC) — занимают среднее положение (размер экрана — 7 дюймов) между планшетным и карманным ПК,

работают под управлением Linux или Windows XP Tablet PC Edition 2005 на процессоре с низким потреблением энергии Intel Pentium или VIA C7-M, на частоте около 1 ГГц (рис. 5);

4) карманный персональный компьютер (КПК или Personal Digital Assistant — PDA) — собирательное название класса портативных электронных вычислительных устройств, изначально предложенных к использованию в качестве электронных органайзеров.



Рис. 5. UMPC Samsung Q1 Ultra

Глава 2. Корпус и системная плата персонального компьютера

Компьютер подобен конструктору, позволяющему из разных устройств, выполняющих определенные функции, собрать вычислительную машину. Конструктивно, он по-прежнему напоминает модель IBM PC 1981 г. выпуска. В стандартной конфигурации он состоит из системного блока, клавиатуры и монитора. Производительность и функциональные возможности компьютера определяются характеристиками системного блока. Внутри него располагаются все основные узлы компьютера (рис. 6): материнская плата, блок питания, накопители, разнообразные платы расширения и т. д. Располагающиеся в корпусе устройства выделяют немало тепла, и потому для его отвода и поддержания необходимого температурного режима в системном блоке помещают целую систему вентиляторов («кулеров»). Количество и качество этих вентиляторов (а их не менее трех — для охлаждения центрального процессора, для нагнетания воздуха в корпус и для охлаждения блока питания) определяет шум, производимый включенным компьютером.



Рис. 6. Вид системного блока ПК со снятой крышкой

Изначально, системный блок ставился на стол горизонтально, и этот тип корпуса называется *desktop* (настольный). Корпуса были довольно громоздкие, но со временем, за счет уменьшения площади системной платы удалось сократить их длину. Так появился формат корпуса (и системной платы) *baby-AT* (здесь *AT* следует понимать как *Advanced Technology* — «передовая технология»; термин использовался IBM как название модели персонального компьютера, созданного в 1984 г.), а традиционные корпуса получили название *full-AT*. Позже стали ставить корпус вертикально, слегка изменив расположение отсеков внешних устройств. Так появился тип корпуса *tower* (башня), наиболее популярный ныне. В него можно устанавливать системные платы и карты расширений тех же размеров, что и в *desktop*, но конструктивно он лучше и удобнее за счет наличия жесткого скелета-шасси. Корпуса типа *tower* могут иметь различные размеры.

Корпус *mini-tower* имеет высоту около 35 см, ширину 17-18 см, глубину около 40 см, два отсека формата 5 дюймов, из трех-четырех отсеков формата 3 дюйма на лицевую панель могут выводиться всего два.

Корпус *midi-tower* несколько больше — он имеет высоту около 40 см и по крайней мере три отсека формата 5 дюймов.

Корпус *big-tower* имеет высоту около 60 см и пять-шесть отсеков формата 5 дюймов. Эти корпуса обычно шире (для устойчивости и лучшего охлаждения устройств). Есть и более емкие корпуса, предназначенные для компьютеров-серверов.

В конце 90-х гг. был принят стандарт на конструкцию системной платы и корпуса — *ATX* (*AT eXtended* — расширенный *AT*), отличающийся от стандарта *AT*. В новом стандарте на задней стенке корпуса предусмотрено окно, в которое выходят разъемы периферии, установленной на системной плате, в то время, как в корпусах *AT* было только отверстие для разъема клавиатуры. Системная плата форм-фактора *ATX* без проблем устанавливается только в корпус этого же стандарта.

В 2003 г. фирма Intel выпустила спецификацию интерфейсов *BTX* (*Balanced Technology eXtended*). Цель спецификации — обеспечить стандартизацию интерфейсов и конструктивов настольных компьютеров с учетом электрических, тепловых и механических характеристик компонентов.

Основным функциональным компонентом системного блока ПК является материнская (системная) плата. Она представляет собой лист фольгированного текстолита, на котором размещаются электронные

компоненты и разъемы для подключения дополнительных плат, внешних устройств, блока питания.

Существуют материнские платы, у которых все необходимые для работы компьютера элементы размещены непосредственно на плате. Это так называемые платы All In One. Однако у большей части персональных компьютеров системные платы содержат лишь основные узлы, а элементы связи с периферийными устройствами (платы или карты расширения) устанавливаются в слоты расширения. Кроме слотов расширения на материнской плате имеется гнездо для установки центрального процессора (CPU) — сокет (рис. 7-1), разъем для подключения блока питания (рис. 7-4), несколько разъемов для установки модулей памяти (рис. 7-5), 80-контактные разъемы для подключения накопителей с интерфейсами IDE (рис. 7-6), 7-контактные разъемы интерфейса SATA и некоторые другие.

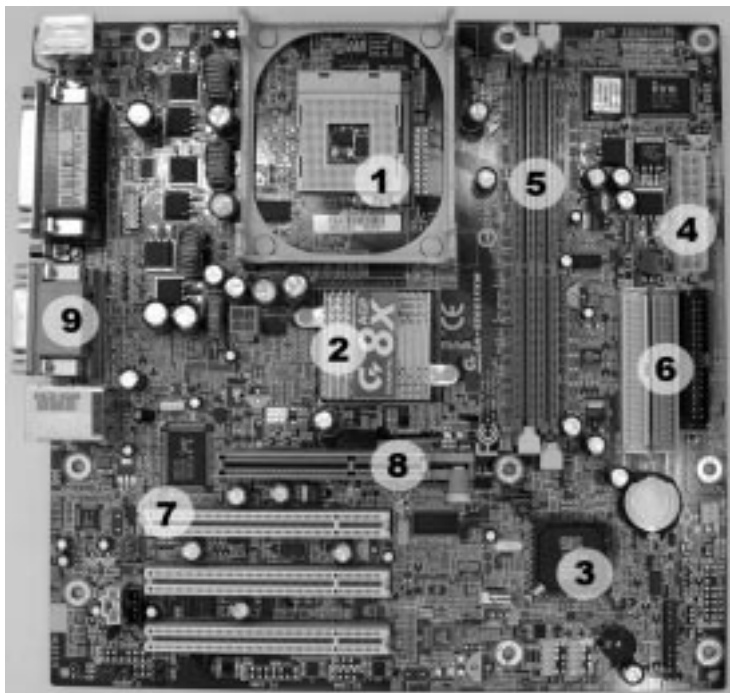


Рис. 7. Системная плата ПК

Также на материнской плате находится микросхема флэш-памяти, предназначенная для хранения системного программного обеспечения — базовой системы ввода-вывода или BIOS (Basic Input Output System). Память часов реального времени и конфигурационных установок системы — RTC CMOS RAM (Real Time Clock CMOS RAM) представляет собой энергонезависимую экономичную микросхему памяти, выполненную по технологии CMOS и предназначенную для хранения показаний часов реального времени, а также сведений о конфигурации системы.

При загрузке операционной системы с помощью программы Setup можно обратиться к этой памяти и внести в нее необходимые изменения. Микросхема памяти постоянно находится во включенном состоянии благодаря питанию от встроенного в материнскую плату аккумулятора. Эта память должна хранить сведения о настройке компьютера (его конфигурации).

Как уже было сказано, по краю материнской платы традиционно располагаются разъемы для подключения внешних устройств (рис. 7–9). К разъемам стандарта PS/2 подключаются клавиатура и мышь. Традиционное подключение принтера выполняется через 25-контактный разъем параллельного порта. Через разъем стандартного последовательного или COM-порта могут подключаться такие низкоскоростные устройства, как модемы, программаторы и др. Для подключения к компьютеру большого количества разнотипных периферийных устройств используются разъемы USB-интерфейса. На платах со встроенным звуком и видео дополнительно присутствуют разъемы для подключения аудиоустройств и видеомонитора.

Материнские платы различаются, в первую очередь, по типу поддерживаемых процессоров, или по типу сокета (socket). Новые виды процессорных разъемов появляются и исчезают вместе с появлением определенной линейки процессоров и прекращением их производства. В настоящее время Intel выпускает процессоры под сокет 775 (хотя еще можно встретить системные платы и процессоры под сокет 478), а фирма AMD — под сокет AM2 (его предшественник — сокет 939).

Другая важная характеристика материнской платы — *чипсет*, на основе которого она производится. Раньше компьютер имел до 2-х сотен микросхем на материнской плате. Современные компьютеры содержат две основные большие микросхемы чипсета:

— Контроллер-концентратор памяти (MCH) или Северный мост (North Bridge), который обеспечивает работу процессора с памятью и с видеоподсистемой (рис. 7-2);

— контроллер-концентратор ввода-вывода (ICH) или Южный мост (South Bridge), обеспечивающий работу с внешними устройствами (рис. 7-3).

Таким образом, *чипсет* — это набор микросхем материнской платы для обеспечения работы процессора с памятью и внешними устройствами.

Чипсет определяет типы шинных интерфейсов с дополнительными устройствами. К этим интерфейсам в современном компьютере можно отнести шины PCI (рис. 7-7) и PCI Express и поддерживаемую еще на некоторых платах шину AGP (рис. 7-8). Встроенные в плату контроллеры портов ввода вывода и сетевые карты определяют возможности подключения внешних устройств и сетевых подключений. SATA и IDE-контроллеры определяют типы подключаемых дисковых накопителей. Широкие возможности для подключения внешних устройств дает поддержка материнскими платами интерфейсов USB и FireWire, позволяющих подключать несколько устройств к одному порту компьютера (о них речь пойдет ниже).



Глава 3. Процессоры

Наиболее значимым элементом ПК является *центральное процессорное устройство* (CPU — Central Processing Unit), или просто процессор. Считается, что именно он определяет основные функциональные возможности современного персонального компьютера, его класс, стоимость, престиж (хотя это не совсем так — на производительность компьютера влияют и другие его элементы: системная плата, память, устройства ввода-вывода и т. д.).

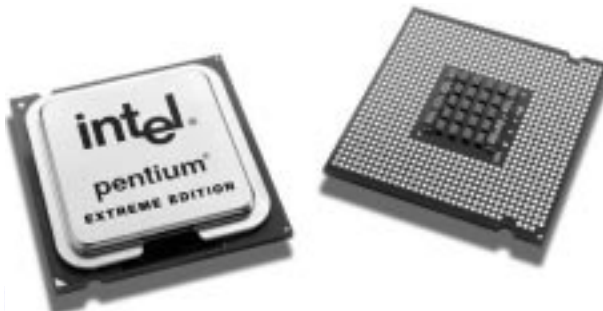


Рис. 8. Процессоры Intel

Технически процессор реализован как сверхбольшая интегральная микросхема, структура которой постоянно усложняется, а количество функциональных элементов (транзисторов), размещаемых на ней, постоянно возрастает (от 30 тыс. в процессоре Intel 8086 до 5 млн. в процессоре Pentium II, а в двухядерном процессоре Athlon 64X2 их уже более 200 млн.).

Основными характеристиками процессора являются:

- тактовая частота;
- разрядность;
- объем кэш-памяти;
- количество ядер.

Тактовая частота задает ритм работы компьютера. Чем выше тактовая частота, тем меньше длительность выполнения команды. Единица измерения частоты — МГц (Мегагерц — миллион тактов в секунду) или ГГц (Гигагерц).

Под *разрядностью* процессора в самом общем смысле понимается количество разрядов у чисел, с которыми способен работать процессор. Первый микропроцессор, реализованный в виде отдельной

микросхемы — Intel 4004 имел разрядность 4-бит. Это значит, что модулю, выполняющему вычисления с целыми числами, были доступны значения лишь от -7 и до 8 (это связано со способом представления целых чисел). Впрочем, уже очень скоро появился на свет 8-битный 8008 (от -127 до 128), также скоро сменил его 16-битный 8086 (-32768 .. 32767), а уже в 1985 г. был представлен 80386 процессор, впервые реализовавший 32-битный режим работы. Последние разработки Intel (на основе *архитектуры EM64T* — Xeon, Core 2 и др.) и AMD (на основе *архитектуры AMD64* или *x86-64* — Athlon 64, Athlon 64 FX и др.) имеют разрядность 64 бит.

Кэш представляет собой быструю оперативную память относительно небольшого размера, размещенную, как правило, на кристалле процессора и предназначенную для временного хранения команд и данных при обмене с относительно медленной оперативной памятью. Кэши принято обозначать символом L с числовым индексом, характеризующим его быстродействие и удаленность (физическую) от исполнительных модулей процессора. Так, характеризуя процессор, обычно говорят о кэше первого (раздельно — L1 данных и L1 команд) и второго (L2) уровня.

Основным средством повышения производительности долгое время считалось повышение тактовой частоты. Начав с частоты немногим более 1 ГГц, тактовая частота процессоров Intel Pentium IV сначала преодолела рубеж в 2 ГГц, затем — в 3 ГГц, а сейчас упорно подходит к отметке 4 ГГц. Но с ростом тактовой частоты росла и потребляемая мощность процессоров и их тепловыделение (не считая ряда других технических проблем).

Невозможность дальнейшего эффективного увеличения тактовой частоты подстегнула ведущих производителей процессоров (компания AMD и Intel) к переходу на производство двуядерных, а в дальнейшем — *многоядерных* процессоров. Двуядерные процессоры, появившиеся в 2005-м г., содержали на одном полупроводниковом кристалле два равноценных процессорных ядра, то есть образовывали своего рода двухпроцессорную систему на одном полупроводниковом кристалле. Тогда на смену одноядерным Intel Pentium 4 и Celeron стали появляться сначала двуядерные продолжатели торговой марки Pentium (Pentium Extreme Edition, Pentium D и др.), а с января 2006 г. — представители новой торговой марки Intel Core (а потом и Core 2). И, наконец, как результат естественного развития

технологии, зимой 2007 г. был представлен четырехядерный процессор Intel Core 2 Quad. Практически синхронно с Intel, ее основной конкурент (компания AMD) также сначала дополнил линейку своих одноядерных процессоров (Athlon 64, Opteron и др.) двухядерными Athlon 64X2 (или Athlon X2), Turion 64 X2, Opteron. И так же, как и Intel, в 2007 г. эта компания анонсировала свою версию четырехядерного процессора семейства Opteron.

Как уже было упомянуто выше, считается, что именно характеристики процессора определяют производительность персонального компьютера, и поэтому практически любой продавец ПК, описывая его, публикует значения этих характеристик. Выяснить основные параметры процессора можно при помощи целого ряда специализированных тестирующих программ, таких, например, как CPU-Z (ее можно найти на сайте разработчика — www.cpuid.com), SiSoftware Sandra (<http://www.sisoftware.net>) и многих других. Так, например, в окне CPU-Z (рис. 9), видно, что процессор Intel Core 2 Extreme QX6700:

- использует сокет 775;
- имеет разрядность — 64-х битная (поскольку упомянуты инструкции архитектуры EM64T);
- работает на тактовой частоте — 2667 МГц;
- количество ядер (cores) в его составе — 4;
- имеет кэш L1 — по 32 кБ на ядро (как для данных, так и для команд);
- имеет кэш L2 — по 4МБ на каждую пару ядер.

Из всего сказанного, правда, не совсем понятно, насколько сильно описанные параметры процессора влияют на его производительность. Для оценки этого влияния можно использовать результаты измерения вычислительной мощности процессора в эталонных величинах — в MIPS (миллион операций в секунду — величина, которая показывает, сколько миллионов инструкций в секунду выполняет процессор в некотором тесте) и в MFLOPS (миллион элементарных арифметических операций над числами с плавающей точкой, выполненных в секунду). Арифметические тесты производительности процессора есть, например в *SiSoftware Sandra*. Альтернативный способ заключается в измерении производительности ПК в некотором синтетическом (комплексном) тесте. Такой метод подразумевает использование таких программ, как PCMark 05, например, которые оценивают производительность процессора (или ПК в целом) в неко-

торых условных единицах (баллах). Результаты такого тестирования можно посмотреть в табл. 1.

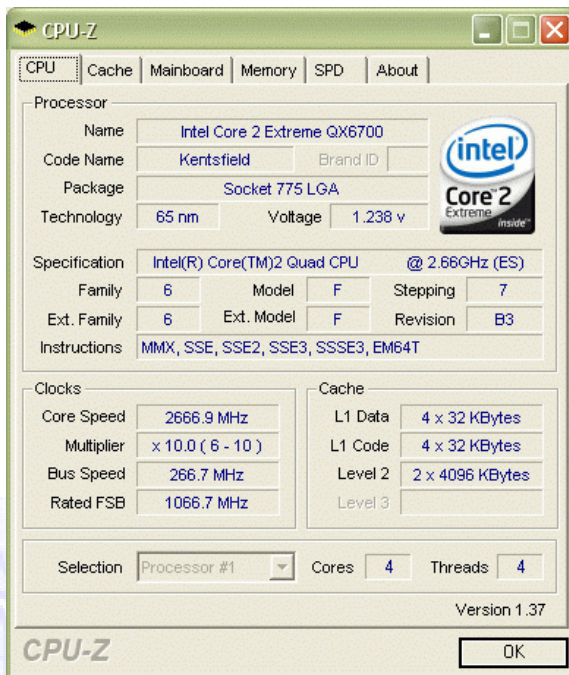


Рис. 9. Характеристики процессора в окне программы CPU-Z

Таблица 1

Сравнительная оценка производительности процессоров

Тип процессора	Сокет	Кол-во ядер	Тактовая частота, МГц	L2, кБ	PCMark, баллы
Intel Core 2 Extreme QX6850	775	4	3000	8192	9697
Intel Core 2 Extreme QX6700	775	4	2666	8192	8450
Intel Core 2 Duo E6700	775	2	2666	4096	6670
AMD Athlon 64 X2 5000+	AM2	2	2600	1024	5339
Intel Core 2 Duo E4400	775	2	2000	2048	5049
AMD Athlon 64 X2 EE 3600+	AM2	2	2000	512	4091
Intel Pentium 4 541	775	1	3200	1024	4074
Intel Celeron S 440	775	1	2000	512	3646

Глава 4. Оперативная память

Память, наряду с процессором и устройствами ввода-вывода является одним из основных компонентов компьютера. Она применяется практически во всех подсистемах ПК, выступая в качестве сверхоперативной кэш-памяти (упомянутой ранее), оперативной памяти или внешней памяти (памяти на внешних носителях).

Основная, или *оперативная, память* — это память, предназначенная для временного хранения данных и команд, необходимых процессору для выполнения им операций. Оперативная память передает процессору команды и данные непосредственно либо через кэш-память. Каждая ячейка оперативной памяти имеет свой индивидуальный адрес. Ее другое название — ОЗУ (оперативное запоминающее устройство), а в английском варианте — RAM (Random Access Memory — память с произвольным доступом). Произвольность доступа подразумевает возможность операций записи в любую ячейку ОЗУ в произвольном порядке.

ОЗУ — энергозависимая память, то есть при отключении питания информация, хранящаяся в ней, теряется. Основу оперативной памяти составляют микросхемы динамической памяти (*DRAM — Dynamic RAM*). Принцип работы DRAM заключается в том, что каждый бит такой памяти представляется в виде наличия или отсутствия заряда на конденсаторах, образованных элементами полупроводниковых микросхем и являющихся ячейками памяти. Длительное отсутствие обращений к ячейке памяти, приводит к снижению заряда и, как следствие, потере информации. Поэтому для сохранности данных требуется регулярная подзарядка конденсатора, что возможно при периодическом выполнении операций чтения/записи — динамический режим работы памяти.

В ходе развития DRAM появились различные модификации этой памяти. Так, для синхронизации работы памяти с тактами работы внешней шины данных была разработана *синхронная оперативная память (Synchronous DRAM или SDRAM)*.

Позже появилась технология, позволяющая за один такт синхронизации данные передавать дважды — *DDR (Double Data Rate — удвоенная скорость передачи данных)*. Дальнейшее развитие этой технологии привело к появлению памяти DDR2 и DDR3, которые, основываясь на технологии DDR SDRAM, за счет технических изменений показывают более высокое быстродействие.

Быстродействие микросхем DRAM определяется рабочей частотой шины, на которой они работают, и интервалами времени (или так называемыми *таймингами*), характеризующими задержки при операциях с памятью.

Из числа временных характеристик работы памяти, наибольшее значение имеет параметр CAS Latency (t_{CL}), определяемый как минимальное время между подачей команды на чтение (CAS) и началом передачи данных (задержка чтения). Измеряется он в тактах, а выяснить его значение можно, используя те же специализированные программы, что и в случае изучения характеристик процессора (CPU-Z, например — рис. 10).

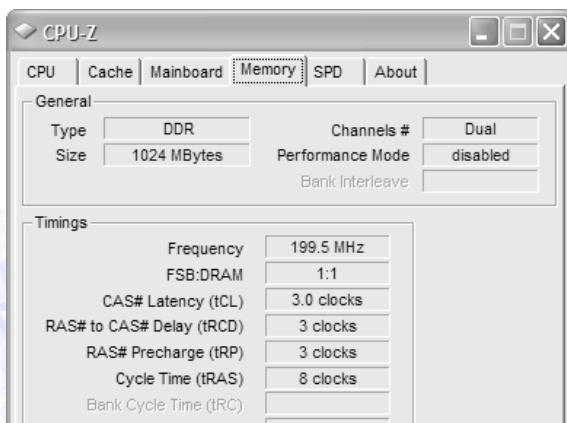


Рис. 10. Характеристики памяти DDR SDRAM в окне CPU-Z

В современных компьютерах в системном ОЗУ используются не отдельные чипы памяти, а изготавливаемые на их основе модули памяти. Модули представляют собой небольшие текстолитовые платы с печатным монтажом и установленными на них микросхемами памяти. Устанавливаются модули в специальные слоты на материнской плате. Посредством шин они связываются с центральным процессором и другими устройствами.

В процессе эволюции компьютера использовались модули разных типов памяти: SIMM, DIMM и RIMM.

Модули с однорядным расположением выводов имеют название *SIMM (Single Inline Memory Module)*. Модули с двухрядным распо-

ложением выводов — *DIMM (Dual Inline Memory Module)*. Модули технологии Rambus — *RIMM (Rambus Inline Memory Module)*. В настоящее время широкое распространение получили только модули типа DIMM для настольных компьютеров и *SO-DIMM (Small Outline DIMM)* для мобильных систем (рис. 11).

Стандартные модули SDRAM DIMM имеют 168 контактов, модули DDR SDRAM DIMM (обычно называются просто DDR DIMM) — 184 вывода, DDR2 DIMM — 240 выводов, модули SO DIMM — 144 и 200 выводов. Модули имеют конструктивные особенности, которые предотвращают неправильную установку в разъемы.

Память SDRAM работала на частотах шины компьютера 66, 100 и 133 МГц и отвечала спецификациям PC66, PC100 и PC133 соответственно. Модули памяти, изготавливаемые по технологии DDR SDRAM, имеют обозначение, в котором указывается не тактовая частота, а пропускная способность шины. Так, модуль, соответствующий спецификации PC3200 (по другому его маркируют как DDR 400), может работать с пропускной способностью шины 3200 МБ/с. Здесь 400 — это так называемая «эффективная» частота передачи данных в МГц, равная удвоенной частоте шины (200 МГц).

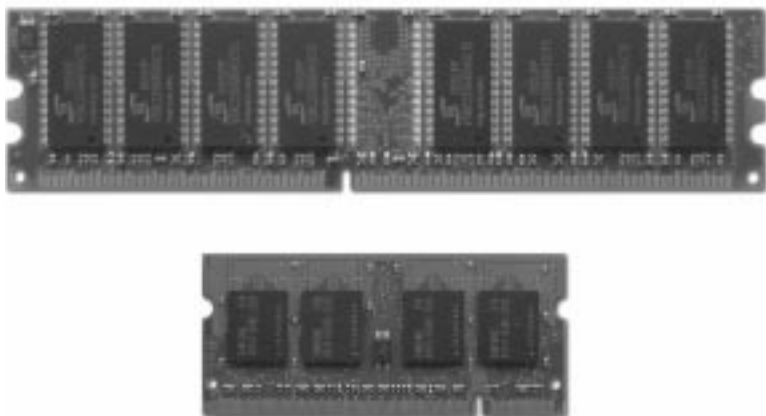


Рис. 10. Модули ОЗУ — DDR DIMM (сверху) и SO DIMM (снизу)

Пропускная способность и частота шины взаимосвязаны. Модули DIMM имеют разрядность 64 бит (8 Байт), а значит, именно столько информации будет передаваться за каждый такт синхроимпульсов.

И при эффективной частоте DDR, равной 400 МГц, пропускная способность будет $400 \times 8 = 3200$ МБ/с.

Модули DDR DIMM выпускались в спецификациях PC1600 (частота шины — 100 МГц, пропускная способность $100 \times 2 \times 8 = 1600$), PC2100 (133 МГц), PC2700 (166 МГц) и PC3200 (200 МГц).

Модули памяти DDR 2 работают на частотах от 200 до 400 МГц (PC3200 — PC6400), DDR 3 — на частотах от 400 МГц и выше.



Глава 5. Накопители

Накопители на жестких магнитных дисках

Накопитель на жестких магнитных дисках, жесткий диск, винчестер или HDD (*Hard Disk Drive*), — энергонезависимое, перезаписываемое компьютерное запоминающее устройство. Является основным накопителем данных практически во всех современных компьютерах.

Само название «винчестер» появилось в 1973 г., когда фирма IBM разработала ЭВМ 3340. Тогда впервые в одном неразъемном корпусе были объединены считывающие головки и пластины диска по 30 МБ с маркировкой «30/30», что напоминало пользователям калибр двустволки «Винчестер 30/30». В Европе и США название «винчестер» вышло из употребления в 1990-х гг., в российском же компьютерном сленге название «винчестер» сохранилось, в том числе и в сокращенном варианте — «винт».

В корпусе типичного HDD находится шпиндель с закрепленными на нем круглыми пластинами из алюминия или стекла (рис. 12), блок считывающих головок с устройством позиционирования (поворотным сервоприводом или актуатором), размещенный под пластинами электродвигатель и блок электроники. Пластины покрыты слоем ферромагнитного материала (двуокиси хрома или сплава кобальта). Считывающие головки в рабочем режиме не касаются поверхности пластин благодаря прослойке набегающего потока воздуха, образуемого у поверхности при быстром вращении. Расстояние между головкой и диском составляет несколько нанометров (в современных дисках 5-10 нм), а в отсутствие вращения дисков, головки находятся у шпинделя в безопасной зоне, где исключен их контакт с поверхностью дисков. При записи информации головка, проходя над поверхностью вращающегося диска, намагничивает миллиарды *горизонтально ориентированных* дискретных областей — доменов. Каждая из этих областей является логическим нулем или единицей, в зависимости от намагниченности. Начиная с 2005 г., на рынке появились устройства использующие технологию *перпендикулярной записи*. Изменение ориентации доменов позволило использовать более сильные магнитные поля и снизить площадь материала, необходимой для записи 1 бита.



Рис. 12. Жесткий диск со снятой крышкой

Из перспективных технологий сейчас можно назвать технологию *HAMR* (Heat Assisted Magnetic Recording). Предложенная компанией Seagate технология предполагает использование лазера совместно с магнитной головкой для точечного разогрева магнитного слоя. В разогретом слое намного проще переориентировать домены, а после остывания их ориентация станет стабильнее.

«Жесткими дисками будущего» сейчас называют *твердотельные диски (Solid State Disks — SSD)*, работающие по тому же принципу, что и флеш-карты. Перспективность этой технологии связана с высокой износостойкостью таких дисков — в них нет движущихся частей, к тому же эти устройства абсолютно бесшумны и экономичны. Однако пока емкость SSD мала, а цена — очень высока. Нынешние модели имеют объем до 64 ГБ, выпуск моделей большей емкости еще только планируется.

Когда возникает потребность описать параметры жесткого диска, как правило, используют следующий набор характеристик, которые можно было бы назвать общими:

- емкость накопителя;
- интерфейс;
- форм-фактор;
- скорость вращения шпинделя.

Емкость (англ. *capacity*) — количество данных, которые могут храниться накопителем. Емкость современных носителей достигла 1000 ГБ (или 1 терабайт). При оценке винчестера по этому параметру необходимо помнить, что в отличие от принятой в информатике системы приставок, обозначающих кратную 1024 (кило = $1024 = 2^{10}$, мега = $1\,048\,576 = 2^{20}$ и т. д.), производителями при обозначении емкости жестких дисков используются кратные 1000 величины. Так, например, реальная емкость жесткого диска, маркированного как «200 ГБ», составляет 186,2 ГБ.

Интерфейс — в общем смысле это система связи и сопряжения узлов компьютера между собой.

До недавнего времени использовались два основных типа интерфейсов для подключения накопителей:

— *ATA (Advanced Technology Attachment* — Усовершенствованная технология подключения) — параллельный интерфейс подключения накопителей (жестких дисков и оптических приводов) к компьютеру был недорогим и не очень быстрым решением, применяемым для настольных компьютеров;

— *SCSI (Small Computer System Interface* — Интерфейс малых вычислительных систем) — скоростное и дорогостоящее решение, применяемое преимущественно для серверов.

Разные версии ATA известны под синонимами IDE, EIDE, UDMA, ATAPI; с появлением SATA также получил название *PATA (Parallel ATA)*.

Первоначальная версия стандарта ATA была разработана в 1986 г. фирмой Western Digital и по маркетинговым соображениям получила название IDE (*Integrated Drive Electronics* — «Электроника, встроенная в привод»). Оно подчеркивало важное нововведение: контроллер привода располагается в нем самом, а не в виде отдельной платы расширения. Такой подход упрощал подключение винчестеров к компьютеру и обеспечивал высокий уровень совместимости устройств разных фирм.

Подключение устройств к системной плате осуществлялось с помощью широкого кабеля (или «шлейфа»), к которому можно подключать два винчестера. Для корректной адресации устройств один из винчестеров должен быть установлен в режим *Master* (ведущий), другой — в режим *Slave* (ведомый). Режимы работы дисководов задавались с помощью переключателей, расположенных около интерфейсного разъема винчестера.

Пропускная способность ATA в различных стандартах составляла от 33 МБ/с (ATA33) до 133 МБ/с (ATA133). В то время как для SCSI тот же параметр составлял от 160 МБ/с до 320 МБ/с.

Интерфейс Serial ATA представляет собой эволюционное развитие интерфейса ATA. Это переход от параллельной шинной архитектуры к последовательной.

Пропускная способность интерфейса SATA увеличилась до 150 МБ/с (для SATA2 — 300 МБ/с, SATA3 — 600 МБ/с). В интерфейсе SATA применяется 7-проводный кабель с разъемами для подключения к системной плате с одной стороны и к винчестеру — с другой (рис. 13). Это намного удобнее при установке и эксплуатации, занимает меньше места, внутри корпуса улучшается циркуляция воздуха. И самое главное — длина соединительного кабеля может достигать одного метра. Для интерфейса IDE ATA длина шлейфа ограничивалась 46 см.

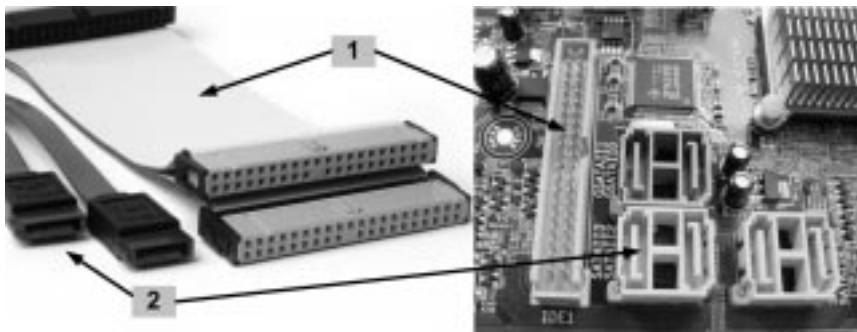


Рис. 13. Интерфейсы IDE (1) и SATA (2)

В SATA отпала необходимость установки перемычек ведущий/ведомый, каждое устройство подключается к своему каналу и никоим образом не влияет на остальные. Изменился и разъем питания, хотя некоторые производители еще снабжают свои накопители дублирующим разъемом питания старого образца, строго предупредив «ни в коем случае не подключать оба питающих разъема одновременно — во избежание выхода устройства из строя».

Физический размер (форм-фактор) — характеристика основана на диаметрах используемых в жестком диске пластин. Почти все современные накопители для персональных компьютеров и серверов

используют типовые размеры 3.5 и 2.5 дюйма. Последние чаще применяются в ноутбуках. Другие распространенные форматы — 1.8 дюйма, 1.3 дюйма и 0.85 дюйма используются в субноутбуках и других ультрамобильных устройствах.

Скорость вращения шпинделя (spindle speed) — количество оборотов шпинделя в минуту (об/мин или rpm). От этого параметра в значительной степени зависят время доступа и скорость передачи данных. В настоящее время выпускаются винчестеры со следующими стандартными скоростями вращения: 4200, 5400 и 7200 об/мин (ноутбуки), 7200 и 10000 об/мин (персональные компьютеры), 10000 и 15000 об/мин (серверы и рабочие станции).

К числу менее распространенных характеристик можно отнести:

- потребление энергии — важный фактор для мобильных устройств;

- уровень шума — шум, который производит механика накопителя при его работе;

- скорость передачи данных — параметр, зависящий как от используемого интерфейса и скорости вращения шпинделя, так и от прочих индивидуальных свойств HDD.

Переносные винчестеры.

Накопители на жестких дисках позволяют хранить огромные объемы информации, имея при этом высокую скорость записи и чтения данных и достаточно небольшие размеры в современных исполнениях. Единственным недостатком является невозможность использования жестких дисков в качестве съемного носителя данных.

Попыткой решения этой проблемы являются системы, превращающие винчестер во внешнее портативное устройство, подключаемое по необходимости к различным компьютерам. Такой накопитель представляет собой обычный винчестер IDE или SATA, размещаемый в специальном корпусе.

В качестве интерфейса подключения к компьютеру могут использоваться USB или FireWire (ее еще называют IEEE 1394 — последовательная шина для подключения внешних высокоскоростных устройств). Степень портативности внешних жестких дисков существенно зависит от форм-фактора используемого HDD. Громоздкие внешние носители на основе обычных 3.5-дюймовых винчестеров требуют наличия независимого блока питания (рис. 14) и дополнительного подключения к электрической сети, в то время как компактные 2.5-дюймовые винчестеры, как правило, питаются непосредственно от шины USB.

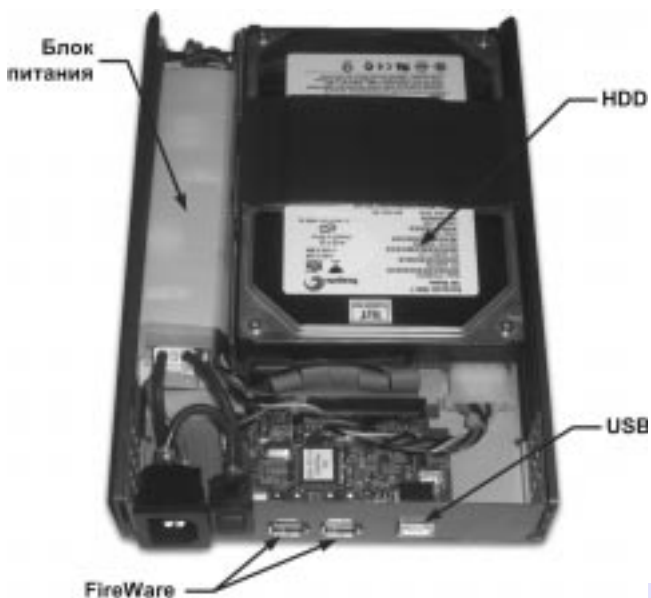


Рис. 14. Внешний HDD со снятой крышкой

Оптические накопители.

Первые оптические диски были разработаны в 1982 г. фирмами Sony и Philips и предназначались для записи звука. Позже появились компакт-диски для хранения данных, для считывания которых применялись приводы CD-ROM (*Compact Disk Read Only Memory* — компакт-диск, предназначенный только для чтения). И те, и другие имеют диаметр 120 мм и толщину 1.2 мм. Информация записывается вдоль спиральной дорожки от центра диска и считывается при помощи инфракрасного лазера с длиной волны 780 нм. Скорость чтения/записи CD указывается кратной 150 кБ/с. Например, 48-скоростной привод обеспечивает максимальную скорость чтения (или записи) CD дисков, равную $48 \times 150 = 7200$ кБ/с (7,03 МБ/с).

Оптические диски CD-ROM, CD-R (Recordable — допускающие однократную запись) и CD-RW (*ReWriteble* — перезаписываемые) имеют прозрачную поликарбонатную (пластиковую) основу, над которой расположен хранящий информацию слой, сверху закрытый лаком. Устройство хранящего слоя может быть различным.

У печатных (штампованных на заводе) компакт-дисков этот слой покрыт светоотражающим составом. Чередующиеся ямки (*pits*) и ровные участки (*lands*) формируют различную интенсивность отраженного луча, которая регистрируется фотоприемником.

Диски CD-R и CD-RW предназначены для записи в домашних условиях. В таких дисках отражающая способность питов и промежутков между ними должны имитироваться другим способом. Это достигается добавлением красителя между золотой (алюминиевой) поверхностью и слоем поликарбоната. В изначальном состоянии уровень красителя прозрачен и позволяет лучу лазера свободно проходить через него и отражаться от золотого (алюминиевого) покрытия. Во время записи лазер попадает на краситель и нагревает его, разрушая химические связи, и образует темные, непрозрачные пятна. При чтении фотодетектор замечает разницу между прожженными пятнами и нетронутыми областями. Это различие интерпретируется так же, как и разница между выемками и ровными поверхностями на обычных компакт дисках. Такие диски в просторечии называются «болванками» и записываются на специальных пишущих приводах для компакт-дисков. Процесс записи называется «прожигом» (от англ. *to burn*) или «нарезкой» диска.

Стандартный объем оптических дисков (CD-ROM, CD-R и CD-RW) — 700 МБ — очень скоро оказался недостаточным для хранения мультимедийной информации. Это обусловило появление оптических дисков нового поколения — DVD. Название DVD сначала понималось как *Digital Video Disk* (Цифровой видеодиск), но сейчас эта аббревиатура чаще понимается как *Digital Versatile Disk* (Универсальный цифровой диск). Эти носители имеют те же размеры, что и обычные CD — 120 мм в диаметре и толщину 1.2 мм, позволяют хранить больший объем информации за счет использования лазера с меньшей длиной волны (635–650 нм). Кроме того, диск DVD может иметь одну или две рабочие стороны и один или два рабочих слоя на каждой стороне.

От их количества зависит емкость диска (из-за чего они получили также названия DVD-5, -9, -10, -18, по принципу округления емкости диска в Гб до ближайшего сверху целого числа — таблица 2). В двухслойной пластине «ямки» расположены в двух плоскостях (рис. 15), нижний слой сверху покрыт полупрозрачной полуотражающей пленкой, а верхний — отражающей. Какой из слоев считывается, опреде-

ляется фокусировкой луча. Сочетания количества слоев и сторон определяется маркировкой DVD:

- SS — Single Sided (односторонние);
- DS — Dual Sided (двухсторонние);
- SL — Single Layer (однослойные);
- DL — Dual Layer (двухслойные).



Рис. 15. Структура двухслойного диска DVD с лучами лазера, сфокусированными на верхнем (А) и нижнем (Б) слоях

Таблица 2

Характеристики DVD носителей

Тип DVD	Маркировка	Емкость DVD-дисков, ГБ
1-сторонние 1-слойные (DVD-5)	SS/SL	4,7
1-сторонние 2-слойные (DVD-9)	SS/DL	8,5
2-сторонние 1-слойные (DVD-10)	DS/SL	9,4
2-сторонние 2-слойные (DVD-18)	DS/DL	17,1

Первые записываемые диски DVD появились в 1997 г., именно тогда был разработан стандарт записи DVD-R(W) как официальная спецификация записываемых и перезаписываемых дисков. Однако цена лицензии на эту технологию была слишком высока, и поэтому несколько производителей пишущих приводов и носителей для записи объединились в «*DVD+RW Alliance*», который и разработал в середине 2002 г. стандарт DVD+R(W), стоимость лицензии на который была ниже. Поначалу «болванки» (чистые диски для записи) DVD+R(W) были дороже, чем «болванки» DVD-R(W), но теперь цены сравнялись.

И хотя стандарты записи «+» и «-» лишь частично совместимы, практически все современные приводы для DVD могут читать и записывать оба формата дисков.

В настоящее время уже доступно следующее поколение оптических дисков. Целью разработки новых технологий записи является возможность хранения информации в формате *HDTV (High Definition*

TeleVision — телевидение высокой четкости), которое уже доступно в США и Японии. Девять компаний Hitachi, LG, Matsushita (Panasonic), Pioneer, Philips, Samsung, Sharp, Sony и Thomson создали цифровые видеодиски нового поколения — Blu-Ray (буква «e» в слове blue пропущена по маркетинговым соображениям). Конкурирующая разработка — диски HD DVD (*High Definition DVD*) от компании Toshiba и NEC, поддержанная Microsoft и Intel. К настоящему времени, правда Toshiba — разработчик формата и единственный производитель проигрывателей HD DVD, выпустила официальное заявление, в котором говорится об отказе от дальнейшего развития и продвижения этого формата. Обе технологии подразумевают использование синего лазера с длиной волны 405 нанометров. Емкость оптических дисков Blu-ray составляет 25 ГБ на один слой и 100 ГБ на четыре слоя.

Флэш-память

Флэш-память — разновидность твердотельной полупроводниковой энергонезависимой перезаписываемой памяти. Бытует мнение, что название *Flash* применительно к типу памяти переводится как «вспышка». На самом деле это не совсем так.

Одна из версий его появления говорит о том, что впервые в 1989–1990 гг. компания Toshiba употребила слово *Flash* в контексте «быстрый, мгновенный» при описании своих новых микросхем. Вообще, изобретателем считается Intel, представившая флэш-память в 1988 г.

Наиболее популярная разновидность флэш-памяти — устройства хранения с интерфейсом USB, который есть во всех современных компьютерах (рис. 16). Конструкция этих устройств состоит из двух основных компонентов: микросхемы флэш-памяти (на рисунке она справа) и контроллера (слева).



Рис. 16. USB-флэш со вскрытым корпусом

«Флэшки» заменили устаревшие и постепенно выходящие из употребления дискеты.

Количество операций чтения у флэш-памяти неограниченно, но осуществить запись в нее можно лишь ограниченное число раз (порядка 10 тысяч). Несмотря на наличие этого предела, 10 тысяч циклов перезаписи — это намного больше, чем способна выдержать дискета или CD-RW. А с учетом того, что современные USB-флэш позволяют хранить информацию едва ли не большего объема чем DVD, то становится понятно, что за этими устройствами хранения большое будущее.



Глава 6. Подключение внешних устройств

Шины и интерфейсы

С самого начала развития персональных компьютеров наличие шин расширения позволяло наращивать возможности вычислительной машины путем подключения дополнительных (или периферийных) устройств через соответствующие разъемы.

Шины разделяют на внутренние (*local bus*) и внешние (*external bus*). Первые разработаны для подключения внутренних устройств, таких, как видеоадаптеры, звуковые платы и т. д., а вторые предназначены для подключения внешних устройств — принтеров, сканеров, внешних модемов и проч. При использовании внутренней или локальной шины (PCI или PCI-E, например) подключить к компьютеру дополнительное устройство можно через специальную плату расширения, имеющую контактную группу соответствующего ей стандарта. Такая плата (ее называют также картой) вставляется в интерфейсный разъем (слот) системной платы. Тем самым обеспечивается включение размещенного на карте устройства в состав компьютерной аппаратуры. Как правило, подключение устройств к внешним шинам возможно через интерфейсные разъемы или *порты ввода-вывода* на корпусе ПК. Здесь порт ввода-вывода — это стандартный контроллер, для которого разрабатываются внешние устройства.

Процессор с периферийными устройствами обменивается байтами (1 Байт = 8 бит), словами (то есть по два байта — 16 бит), двойными словами (32 бита) данных. Для того чтобы передавать группу битов, существует два подхода к организации интерфейса.

— Параллельный интерфейс — для каждого бита передаваемой группы имеется своя сигнальная линия (обычно с двоичным представлением), и все биты группы передаются одновременно за единицу времени, то есть продвигаются по шине параллельно. Примеры: параллельный порт подключения принтера (LPT-порт, 8 бит), интерфейс ATA/IDE (16 бит), SCSI (8 или 16 бит), шина PCI (32 или 64 бита).

— Последовательный интерфейс — используется лишь одна сигнальная линия, и биты группы передаются друг за другом по очереди; на каждый из них отводится свой квант времени (битовый интервал). Примеры: последовательный коммуникационный порт (COM-порт), последовательные универсальные шины USB и FireWire,

последовательные интерфейсы для подключения устройств хранения данных — Serial ATA, SAS (последовательный вариант SCSI).

Среди внутренних шин распространены шины расширения PCI или PCI-x. Соответственно и платы расширения должны иметь контактные группы, отвечающие приемным гнездам слотов этих шин.

В любом случае подобный способ подключения предполагает «вскрытие» системного блока и манипуляции с разъемами системной платы. У портативных компьютеров слот шины расширения (PC Card) выведен наружу, и подключение к такой шине не требует «вскрытия» компьютера.

Для связи ПК с внешним миром могут служить коммуникационные или последовательный порт (COM-порт). Последний может быть использован для связи двух ПК, подключения внешнего модема, мыши и для некоторых других целей. Внутренний модем (подключаемый через шину расширения) воспринимается системой как еще один COM-порт.

Постепенно уходящий в прошлое параллельный порт LPT может быть использован для подключения принтера (хотя для этой цели все чаще применяется USB-интерфейс). К шине USB можно подключить до 127 устройств, она допускает «горячее» включение и отключение того или иного устройства, при этом операционная система сама загружает или отгружает соответствующие драйверы. Интерфейс USB давно стал обязательным для любого современного ПК и нередко материнские платы оснащаются вместо устаревших COM и LPT портов несколькими дополнительными разъемами этого универсального интерфейса. Теперь многие периферийные компьютерные устройства (сканеры, принтеры, цифровые фотокамеры и даже мониторы) оборудуют разъемами и устройствами, обеспечивающими для связи с системным блоком интерфейс USB.

Другая последовательная шина, которая имеет большие перспективы — это FireWire (буквальный перевод с английского: огненный провод). Официальное название этого устройства — высокопроизводительная последовательная шина IEEE 1394. Она обеспечивает скорость обмена 100 мегабит в секунду и более. Эта шина позволяет связать периферийные устройства не только с ПК, но и между собой. Шина обеспечивает одновременное соединение многих устройств (до 63) без применения дополнительной аппаратуры. Это могут быть

цифровые фото- и видеокамеры, видеокамеры для конференций, видеомагнитофоны, сканеры. На базе FireWire можно строить небольшие локальные сети. Для оснащения ПК шиной FireWire нужна соответствующая карта расширения.

Типичная современная системная плата помимо портов ввода-вывода, разъемов универсальных шин, разъемов для подключения мыши и клавиатуры (PS/2), может также иметь интерфейсные разъемы интегрированных в материнскую плату устройств (рис. 17).

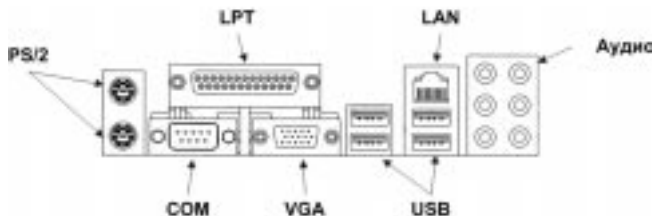


Рис. 17. Разъемы материнской платы для подключения дополнительных устройств

К их числу могут относиться, например, гнездо для подключения сетевого кабеля, монитора, разъемы для подключения аудиоустройств.

Видеокарты

Одной из наиболее употребительных плат расширения, подключаемых к внутренним шинам персонального компьютера, являются так называемые *видеокарты*. Видеокарта — устройство персонального компьютера, предназначенное для приема информации от процессора, обработки ее и управления выводом изображения на экран видеомонитора. В общей производительности компьютера при обработке мультимедийных приложений роль видеокарты может стать ключевой.

В качестве системного интерфейса долгое время использовалась шина AGP, соединявшая видеокарту с центральным процессором и оперативной памятью. Дальнейшего повышения производительности видеосистем удалось достичь с появлением более скоростной и перспективной шины PCI-E (или *PCI-Express*).

Принцип работы видеокарты можно представить следующим образом. Центральный процессор компьютера формирует изображение (кадр) в виде массива данных, записываемых в видеопамять и непосредственно в часть памяти, называемую кадровым буфером. После

этого процессор видеокарты последовательно, бит за битом, строка за строкой, считывает содержимое кадрового буфера и формирует выходной сигнал, который передается на монитор.

Важным параметром видеокарты является установленный на ней объем памяти. Типовые значения величины видеопамати в настоящее время — от 128 МБ и выше.

Не меньшее значение на производительность видеосистемы оказывает установленный на видеокарту чипсет, который изначально состоял из нескольких микросхем. Сейчас же эти микросхемы объединены в одну — видеопроцессор (в некоторых последних модификациях используется два видеопроцессора). Видеопроцессор отвечает за вывод изображения из видеопамати, регенерацию ее содержимого, формирование сигналов развертки для монитора и обработку запросов центрального процессора. Для исключения конфликтов при обращении к памяти со стороны видеопроцессора и центрального процессора используется отдельный буфер, который в свободное от обращений CPU время заполняется данными из видеопамати. Если конфликта избежать не удастся, задерживается обращение центрального процессора к видеопамати, что снижает производительность системы.

Видеопроцессор связан с видеопаматью внутренней шиной. Ее ширина или разрядность является важным скоростным параметром. В настоящее время разрядность этой шины для большинства видеокарт составляет 128 или 256 бит. Частота видеопроцессора и видеопамати на современных картах составляет 400–700 МГц, что требует охлаждения с помощью радиатора и вентилятора и блока питания для компьютера повышенной мощности.

На видеокarte обычно размещаются один или несколько разъемов для подключения внешних устройств. Через выходной разъем *TV-out* подключается телевизор или видеокамера. Эти разъемы бывают 7- и 4-штырьковые, типа *S-Video*, «тюльпан» и композитные. Композитный выход имеет всего один сигнальный провод и подключается к антенному входу телеприемника, такой сигнал дает изображение среднего качества. Для подключения мониторов на видеокартах могут присутствовать два типа разъемов: стандартный 15-штырьковый, для подключения аналоговых мониторов, и типа *DVI* — для подключения цифровых мониторов (рис. 18).



Рис. 18. Видеокарта PCI-E; разъемы: VGA (1), DVI (2), S-Video (3)

Модемы

Модем может рассматриваться в качестве примера устройства, подключаемого как к разъемам системного блока (внешний модем), так в слоты материнской платы (внутренний модем). Модем (аббревиатура, составленная из слов модулятор-демодулятор) — это устройство, применяющееся в системах связи и, как следует из названия, выполняет функцию модуляции и демодуляции. Модулятор осуществляет модуляцию, то есть изменяет характеристики несущего сигнала в соответствии с изменениями входного информационного сигнала, демодулятор осуществляет обратный процесс.

Кроме модуляции-демодуляции модемы могут выполнять сжатие и декомпрессию пересылаемой информации, а также заниматься поиском и исправлением ошибок, возникающих в процессе передачи данных по линиям связи.

Одной из основных характеристик модема является скорость передачи данных, которая измеряется в битах в секунду.

Внешние модемы подключаются к COM или USB-порту и обычно имеют внешний блок питания (существуют USB-модемы, питающиеся от USB). *Внутренние* — устанавливаются внутрь компьютера в слот PCI, PCMCIA и др. Встречаются, правда, и *встроенные*, интегрированные в материнскую плату компьютера.

В зависимости от используемой среды передачи сигнала модемы можно разделить на проводные и беспроводные. К числу

проводных модемов относятся, в первую очередь, обычные аналоговые (или Dial-Up модемы), которые используют обычные коммутированные телефонные линии. Максимальная теоретическая скорость передачи данных без компрессии для таких устройств равна 56 кб/с, что по современным меркам является совершенно недостаточным для комфортной навигации по Интернет. Альтернативой медленным аналоговым являются ISDN-модемы для цифровых коммутируемых телефонных линий и цифровые модемы семейства xDSL, которые для организации выделенных (некоммутируемых) линий используют обычную телефонную сеть и позволяют передавать данные по сети одновременно с голосом. Самой популярной разновидностью этого семейства являются так называемые ADSL-модемы (рис. 19). ADSL расшифровывается как Asymmetric Digital Subscriber Line (асимметричная цифровая абонентская линия). Асимметричность состоит в увеличении скорости передачи в одном направлении за счет снижения этой скорости в другом. При передаче данных из сети абоненту скорость может достигать 8 Мб/с, а при передаче данных от модема (исходящий трафик) она может составлять порядка 1 Мб/с.

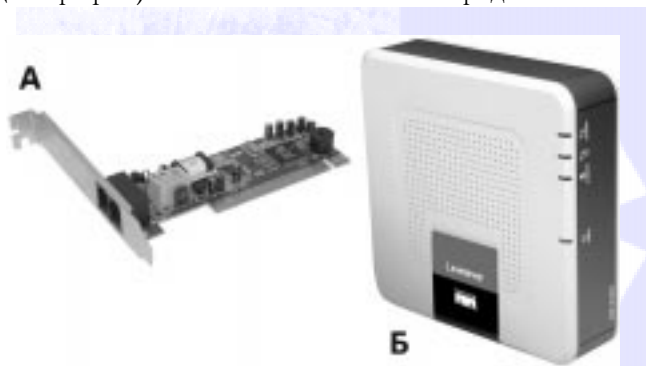


Рис. 19. Внутренний аналоговый PCI-модем (А) и внешний ADSL-модем (Б)

РЕЗЮМЕ

В самом общем смысле под *компьютерными продуктами* понимаются аппаратные и программные средства компьютера как объекты производства и коммерческой деятельности.

Говоря о классификации компьютеров, можно выделить группы, подразделяемые по этапам развития (поколениям), а также по размерам и вычислительной мощности.

Аппаратные средства или компьютерное «железо» можно условно разделить на компоненты, размещенные в системном блоке и внешние устройства. К числу первых можно отнести системную плату персонального компьютера, микропроцессор, оперативную память, дисковые накопители информации (оптические и на жестких магнитных дисках). Производительность компьютера напрямую зависит от характеристик этих устройств. В частности, производительность системной платы зависит от используемого на ней *чипсета* (двух основных микросхем, размещенных на ней); производительность процессора определяется тактовой частотой, числом ядер, разрядностью и величиной кэш-памяти; эффективность применения оперативной памяти зависит от ее объема и скорости передачи данных, измеряемой в МБ/с.

Разнообразные *внешние* устройства для подключения к компьютеру используют интерфейсные разъемы шин расширения, таких, как USB, FireWare, PS/2 и проч. Помимо обязательных клавиатуры и мыши, к числу наиболее популярных устройств, используемых для расширения функциональности персонального компьютера, следует отнести:

— видеoadаптеры (или видеокарты), которые обеспечивают производительность системы в игровых программах и при работе с графическими пакетами;

— модемы (МОДуляторы-ДЕМОдуляторы), открывающие возможность выхода в сеть Интернет.

Вопросы для самопроверки:

1. Что понимается под термином «компьютерные продукты»?
2. Какие компьютеры можно считать первыми персональными?
3. Какие факторы обеспечили популярность архитектуры IBM PC?
4. По каким признакам и на какие классы можно подразделить компьютеры?

5. Какие типы корпусов персональных компьютеров являются наиболее популярными?
6. Какие элементы компьютера располагаются на материнской плате?
7. Какие характеристики процессора можно использовать для оценки его производительности?
8. Каков принцип работы оперативной памяти?
9. Из каких элементов состоит типичный HDD?
10. Какую технологию используют твердотельные диски?
11. Какой объем информации может хранить диск DVD?
12. Что означает ADSL?



РАЗДЕЛ 2. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ПК

Глава 1. Определение и классификация программных продуктов

Понятие «*программное обеспечение*» (*software*) вошло в жизнь с развитием компьютерной индустрии. Без программного обеспечения компьютер — всего лишь электронное устройство, которое не управляется и потому не может приносить пользы. Но это понятие несколько шире, чем комплекс (набор, совокупность) программ. Кроме собственно программ на носителях информации (дискетах или компакт-дисках) оно включает упаковку, эксплуатационную документацию и лицензионное соглашение, когда речь идет о программном продукте, который тиражируется.

Программный продукт — это совокупность отдельных программных средств, их документации, гарантии качества, рекламных материалов, мер по обучению пользователей, распространению и сопродажению готового программного обеспечения.

Основная классификация программного обеспечения, родившаяся еще в 60-е гг., делит его по *сфере применения* на два класса: системное и прикладное. Впервые такое разделение было проведено, видимо, в знаменитом проекте IBM System/360. Согласно ему прикладные программы предназначены для решения определенных задач и получения конкретных результатов, а системные программы — для обеспечения работы прикладных программ. К классу системного ПО традиционно относят операционные системы и разнообразные утилиты по обслуживанию системы. В классе прикладного ПО можно выделить группы офисных пакетов программ, разнообразные программы для обработки графической информации, программы для Интернета и т. д. Промежуточное место занимает группа инструментальных средств разработки программ, таких, как, например, Visual C++, Delphi и проч.

В последнее время стали употребляться выражения «вертикальные» и «горизонтальные» приложения. Эти термины заимствованы из модной экономической фразеологии и в обыденной жизни заменяются неформальными определениями «заказное» и «коробочное» приложение. Их можно считать примером классификации по *источнику финансирования*. В первом случае подразумевается приложение, которое сделано по

конкретному индивидуальному заказу, а во втором — приложение, рассчитанное на массовый спрос, которое можно купить в магазине. Разработка горизонтальных приложений финансируется производителем в расчете на последующую продажу, а разработка вертикальных приложений с самого начала работ оплачивается заказчиком. Из сказанного видно, что вертикальные приложения обладают одним важным преимуществом — при их разработке можно более полно и точно учесть ожидания пользователей, и этого оказывается достаточно для оправдания разработки вертикальных приложений. Однако свои преимущества имеют и тиражируемые коробочные приложения. Они:

- используются многими заказчиками, которые могут обмениваться между собой едиными форматами данных, формами представления информации;

- прошли неоднократную апробацию, проверку другими заказчиками;

- как правило, менее дороги, чем заказные продукты;

- могут быть внедрены в более короткие сроки по проверенным технологиям внедрения.

Так же важной характеристикой ПО является его масштаб. *Масштаб* приложения — это качественная оценка размера приложения с точки зрения его использования. Обычно различают следующие градации размеров приложений:

- настольные (desktop);

- групповые (groupware);

- масштаба предприятия (enterprise).

Это разделение чисто качественное, точных количественных характеристик, отличающих приложения разных классов, указать нельзя.

Настольное приложение, как правило, имеет одного пользователя, может выполняться на одном компьютере и работать автономно, без сети.

Групповое приложение имеет нескольких пользователей, которые могут работать с ним порознь или одновременно, но все принадлежат к одной группе (или категории) пользователей. Как правило, групповые приложения используют локальную сеть или иной способ связи компьютеров.

Приложение *масштаба предприятия* имеет значительное количество пользователей, принадлежащих к различным категориям, выполняет несколько различных связанных между собой функций и работает в локальной или глобальной сети.

Глава 2. Стратегия разработки ПО

Самым существенным и неотъемлемым свойством программных систем является их сложность. Благодаря уникальности и несхожести своих составных частей программные системы принципиально отличаются от технических систем (например, компьютеров), в которых преобладают повторяющиеся элементы.

Сами компьютеры сложнее, чем большинство продуктов человеческой деятельности. Количество их возможных состояний очень велико, поэтому их так трудно понимать, описывать и тестировать. У программных систем количество возможных состояний на порядок величин превышает количество состояний компьютеров.

Потребность контролировать процесс разработки ПО, прогнозировать и гарантировать стоимость разработки, сроки и качество результатов привела в конце 70-х гг. к необходимости перехода от кустарных к индустриальным способам создания ПО и появлению совокупности инженерных методов и средств создания ПО, объединенных общим названием «*программная инженерия*» (software engineering). Впервые этот термин был использован как тема конференции, проводившейся под эгидой НАТО в 1968 г. Спустя семь лет, в 1975 г., в Вашингтоне была проведена первая международная конференция, посвященная программной инженерии. Тогда же появилось первое издание, посвященное программной инженерии, — IEEE Transactions on Software Engineering.

В процессе становления и развития программной инженерии можно выделить два этапа: 70-е и 80-е гг. — систематизация и стандартизация процессов создания ПО и 90-е гг. — начало перехода к сборочному, индустриальному способу создания ПО.

В основе программной инженерии лежит одна фундаментальная идея: проектирование ПО является формальным процессом, который можно изучать и совершенствовать. Для успешной реализации проекта должны быть построены полные и непротиворечивые модели архитектуры ПО.

Под *моделью* понимается полное описание системы ПО с определенной точки зрения. Модели представляют собой средства для визуализации, описания, проектирования и документирования архитектуры системы. Модели строятся для того, чтобы понять и осмыслить структуру и поведение будущей системы, облегчить управле-

ние процессом ее создания и уменьшить возможный риск, а также документировать принимаемые проектные решения. В 70–80-х гг. при разработке ПО достаточно широко применялись структурные методы, базирующиеся на строгих формализованных методах описания ПО и принимаемых технических решений. В настоящее время такое же распространение получают объектно-ориентированные методы. Эти методы основаны на использовании наглядных графических моделей, для описания архитектуры ПО в различных аспектах используются схемы и диаграммы. Наглядность и строгость методов структурного и объектно-ориентированного анализа позволяют разработчикам и будущим пользователям системы с самого начала неформально участвовать в ее создании, обсуждать и закреплять понимание основных технических решений. Перечисленные проблемы породили потребность в программно-технологических средствах специального класса — CASE-средствах, о которых будет рассказано ниже.

Жизненный цикл

Понятие *жизненного цикла* программного обеспечения (ЖЦ ПО) является одним из базовых в программной инженерии. Жизненный цикл программного обеспечения определяется как период времени, который начинается с момента принятия решения о необходимости создания ПО и заканчивается в момент его полного изъятия из эксплуатации.

Основным нормативным документом, регламентирующим состав процессов ЖЦ ПО, является международный стандарт ISO/IEC 12207: 1995 «Information Technology — Software Life Cycle Processes» (*ISO — International Organization for Standardization — Международная организация по стандартизации, IEC — International Electrotechnical Commission — Международная комиссия по электротехнике*). Структура ЖЦ по стандарту ISO/IEC 12207 базируется на трех группах процессов:

- основные процессы ЖЦ ПО (приобретение, поставка, разработка, эксплуатация, сопровождение);
- вспомогательные процессы, обеспечивающие выполнение основных процессов (документирование, управление конфигурацией, обеспечение качества, верификация, аттестация, оценка, аудит, решение проблем);

— организационные процессы (управление проектами, создание инфраструктуры проекта, определение, оценка и улучшение самого ЖЦ, обучение).

Некоторые процессы ЖЦ разберем подробнее.

Разработка включает в себя все работы по созданию ПО и его компонент в соответствии с заданными требованиями, включая оформление проектной и эксплуатационной документации, подготовку материалов, необходимых для проверки работоспособности и соответствующего качества программных продуктов, материалов, необходимых для организации обучения персонала и т. д. Разработка ПО включает в себя, как правило, анализ, проектирование и реализацию (программирование).

Эксплуатация включает в себя работы по внедрению компонентов ПО в эксплуатацию, в том числе конфигурирование базы данных и рабочих мест пользователей, обеспечение эксплуатационной документацией, проведение обучения персонала и т. д., и непосредственно эксплуатацию, в том числе локализацию проблем и устранение причин их возникновения, модификацию ПО в рамках установленного регламента, подготовку предложений по совершенствованию, развитию и модернизации системы.

Процесс *сопровождения* предусматривает действия и задачи, выполняемые сопровождающей организацией (службой сопровождения). Данный процесс активизируется при изменениях (модификациях) программного продукта и соответствующей документации, вызванных возникшими проблемами или потребностями в модернизации либо адаптации ПО. В соответствии со стандартом IEEE-90 под сопровождением понимается внесение изменений в ПО в целях исправления ошибок, повышения производительности или адаптации к изменившимся условиям работы или требованиям.

Среди вспомогательных процессов особо выделим процесс *верификации* — определения того, отвечает ли текущее состояние разработки, достигнутое на данном этапе, требованиям этого этапа. Проверка позволяет оценить соответствие параметров разработки исходным требованиям. Проверка частично совпадает с тестированием, которое связано с идентификацией различий между действительными и ожидаемыми результатами и оценкой соответствия характеристик ПО исходным требованиям.

Управление конфигурацией позволяет организовывать, систематически учитывать и контролировать внесение изменений в ПО на всех стадиях ЖЦ. Общие принципы и рекомендации конфигурационного учета, планирования и управления конфигурациями ПО отражены в проекте стандарта ISO 12207-2.

Процесс *аудита* представляет собой определение соответствия требованиям, планам и условиям договора. Аудит может выполняться двумя любыми сторонами, участвующими в договоре, когда одна сторона проверяет другую.

Каждый процесс характеризуется определенными задачами и методами их решения, исходными данными, полученными на предыдущем этапе, и результатами. Результатами анализа, в частности, являются функциональные модели, информационные модели и соответствующие им диаграммы. ЖЦ ПО носит итерационный характер: результаты очередного этапа часто вызывают изменения в проектных решениях, выработанных на более ранних этапах.

Модели жизненного цикла

Стандарт ISO/IEC 12207 не предлагает конкретную модель ЖЦ и методы разработки ПО (под моделью ЖЦ понимается структура, определяющая последовательность выполнения и взаимосвязи процессов, действий и задач, выполняемых на протяжении ЖЦ). Стандарт ISO/IEC 12207 описывает структуру процессов ЖЦ ПО, но не конкретизирует в деталях, как реализовать или выполнить действия и задачи, включенные в эти процессы.

К настоящему времени наибольшее распространение получили следующие основные модели ЖЦ:

- каскадная модель (70–85 гг.);
- спиральная модель (86–90 гг.).

В каскадной модели основной характеристикой является разбиение всей разработки на этапы, причем переход с одного этапа на следующий происходит только после того, как будет полностью завершена работа на текущем (рис. 20). Каждый этап завершается выпуском полного комплекта документации, достаточной для того, чтобы разработка могла быть продолжена другой командой разработчиков.

Положительные стороны применения каскадного подхода заключаются в следующем:

- на каждом этапе формируется законченный набор проектной документации, отвечающий критериям полноты и согласованности;



Рис. 20. Каскадная схема разработки ПО

— выполняемые в логичной последовательности этапы работ позволяют планировать сроки завершения всех работ и соответствующие затраты.

Однако в процессе использования этого подхода обнаружился ряд недостатков, вызванных, прежде всего тем, что реальный процесс создания ПО никогда полностью не укладывался в такую жесткую схему. В процессе создания ПО постоянно возникала потребность в возврате к предыдущим этапам и уточнении или пересмотре ранее принятых решений. В результате реальный процесс создания ПО принимал вид, представленный на рис. 21.



Рис. 21. Реальный процесс разработки ПО по каскадной схеме

Основным недостатком каскадного подхода является существенное запаздывание с получением результатов. Согласование результатов с пользователями производится только в точках, планируемых после завершения каждого этапа работ, требования к ПО «заморожены» в виде технического задания на все время его создания.

Таким образом, пользователи могут внести свои замечания только после того, как работа над системой будет полностью завершена. В случае неточного изложения требований или их изменения в течение длительного периода создания ПО, пользователи получают систему, не удовлетворяющую их потребностям. Модели (как функциональные, так и информационные) автоматизируемого объекта могут устареть одновременно с их утверждением.

Для преодоления перечисленных проблем была предложена спиральная модель ЖЦ (рис. 22), делающая упор на начальные этапы ЖЦ: анализ и проектирование. На этих этапах реализуемость технических решений проверяется путем создания прототипов. Каждый виток спирали соответствует созданию фрагмента или версии ПО, на нем уточняются цели и характеристики проекта, определяется его качество и планируются работы следующего витка спирали.



Рис. 22. Спиральная модель ЖЦ

Таким образом, углубляются и последовательно конкретизируются детали проекта, и в результате выбирается обоснованный вариант, который доводится до реализации.

Разработка итерациями отражает объективно существующий спиральный цикл создания системы. Неполное завершение работ на каждом этапе позволяет переходить на следующий этап, не дожидаясь полного завершения работы на текущем. При итерационном способе разработки недостающую работу можно будет выполнить на следующей итерации. Главная же задача — как можно быстрее показать пользователям системы работоспособный продукт, тем самым активизируя процесс уточнения и дополнения требований. В силу этого будущий пользователь программного продукта в данной модели становится полноправным участником процессов разработки ПО (в первую очередь на этапах анализа и проектирования). Это предъявляет соответствующие требования и к заказчикам ПО (в том числе специалистам экономического профиля) — они должны владеть основными сведениями в области инженерии программирования, которые бы позволили им продуктивно общаться с разработчиками.

Основная проблема спирального цикла — определение момента перехода на следующий этап. Для ее решения необходимо ввести временные ограничения на каждый из этапов жизненного цикла. Переход осуществляется в соответствии с планом, даже если не вся запланированная работа закончена. План составляется на основе статистических данных, полученных в предыдущих проектах, и личного опыта разработчиков.

CASE-средства разработки приложений

Термин CASE (Computer Aided System Engineering) имеет весьма широкое толкование. В рамках программной инженерии CASE-средства представляют собой основную технологию, используемую для создания и эксплуатации систем ПО. Под *CASE-средством* понимается программное средство, поддерживающее процессы жизненного цикла ПО, включая анализ требований к системе, проектирование прикладного ПО и баз данных, генерацию кода, тестирование, документирование, обеспечение качества, управление конфигурацией ПО и управление проектом, а также другие процессы. CASE-средства вместе с системным ПО и техническими средствами образуют среду разработки ПО.

Наиболее трудоемкими стадиями разработки ПО являются стадии формирования требований и проектирования, в процессе которых

CASE-средства обеспечивают качество принимаемых технических решений и подготовку проектной документации. При этом большую роль играют методы визуального представления информации. Это предполагает построение разнообразных графических моделей (диаграмм), использование многообразной цветовой палитры, сквозную проверку синтаксических правил.

В разряд CASE-средств попадают как относительно дешевые системы для персональных компьютеров с весьма ограниченными возможностями, так и дорогостоящие системы для неоднородных вычислительных платформ и операционных сред. Так, современный рынок программных средств насчитывает около 300 различных CASE-средств, наиболее мощные из которых, так или иначе, используются практически всеми ведущими западными фирмами.

CASE-средствам присущи следующие основные особенности:

- наличие мощных графических средств для описания и документирования системы, обеспечивающих удобный интерфейс с разработчиком и развивающих его творческие возможности;

- интеграция отдельных компонентов CASE-средств, обеспечивающая управляемость процессом разработки ПО;

- использование специальным образом организованного хранилища проектных метаданных (репозитория). Интегрированное CASE-средство (комплекс средств, поддерживающих полный ЖЦ ПО) содержит следующие компоненты:

- а) репозиторий, являющийся основой CASE-средства. Он должен обеспечивать хранение версий проекта и его отдельных компонентов, синхронизацию поступления информации от различных разработчиков при групповой разработке, контроль метаданных на полноту и непротиворечивость;

- б) графические средства анализа и проектирования, обеспечивающие создание и редактирование комплекса взаимосвязанных диаграмм, образующих модели деятельности организации и системы ПО;

- в) средства разработки приложений, включая языки 4GL (Fourth Generation Language — язык 4-го поколения) и генераторы кодов;

- г) средства управления требованиями;

- д) средства управления конфигурацией ПО;

- е) средства документирования;

- ж) средства тестирования;
- з) средства управления проектом.

Основные функции средств организации и поддержки репозитория — хранение, доступ, обновление, анализ и визуализация всей информации по проекту ПО. Содержимое репозитория включает не только информационные объекты различных типов, но и отношения между их компонентами, а также правила использования или обработки этих компонентов. Репозиторий может хранить свыше 100 типов объектов, примерами которых являются диаграммы, определения экранов и меню, проекты отчетов, описания данных, исходные коды и проч.



Глава 3. Технология быстрой разработки приложений — RAD

Давайте рассмотрим наиболее популярные на сегодняшний день подходы к разработке программного обеспечения и некоторые примеры их использования.

Описание технологии

В современной инженерии программирования существуют различные технологические методы и приемы разработки программных продуктов. Выбор той или иной технологии разработки обуславливает разные временные и трудовые ресурсы, а также различную квалификацию разработчиков. Потенциальному потребителю программного продукта нужно иметь представление об особенностях и возможностях различных технологий. Наиболее функционально развитой на сегодня является технология быстрой разработки приложений — RAD. Подход RAD (Rapid Application Development) предусматривает наличие следующих составляющих:

- развитых средств визуального программирования, основанных на использовании готовых фрагментов программ;
- небольших групп разработчиков (от 3 до 7 человек), выполняющих работы по проектированию отдельных подсистем ПО. Это обусловлено требованием максимальной управляемости коллектива;
- короткого, но тщательно проработанного производственного графика (до 3 месяцев);
- повторяющегося цикла, при котором разработчики по мере того, как приложение начинает обретать форму, запрашивают и реализуют в продукте требования, полученные в результате взаимодействия с заказчиком.

Команда разработчиков должна представлять собой группу профессионалов, имеющих опыт в проектировании, программировании и тестировании ПО, способных хорошо взаимодействовать с конечными пользователями (менеджерами проекта, экономистами) и трансформировать их предложения в рабочие прототипы. В основе технологии RAD лежит использование современных средств визуального программирования, в которых высокая скорость разработки приложений связана с возможностью создавать и настраивать различные интерфейсные элементы без написания кода программы.

В настоящее время для разработки программного обеспечения активно используется целый спектр инструментальных сред. Это продукты компании Microsoft — Visual C++ и Visual Basic, популярный продукт компании Borland — среда программирования Delphi, построенная на основе Pascal и др. В зависимости от решаемой задачи разработчик вправе выбирать тот или иной инструмент.

Применение к офисному программированию

Корпорация Microsoft — крупнейший производитель программного обеспечения — создала и активно продвигает в качестве удобного инструмента разработки офисных приложений интегрированный в пакет MS Office язык визуального программирования VBA (Visual Basic for Applications). Для специалистов он интересен прежде всего тем, что позволяет естественным образом автоматизировать решение своих повседневных задач.

Чтобы начать знакомство со средой программирования VBA, необходимо выполнить следующие действия: запустить приложение MS Office (здесь и далее — MS Excel); в меню программы выбрать команду *Сервис — Макрос — Редактор Visual Basic* (или комбинацию Alt+F11). В результате откроется интегрированная среда разработки VBA.

Она имеет стандартный для всех Windows-приложений вид: строка меню, панель инструментов и два окна — *Project* и *Properties*.

Рассмотрим для начала окно Project. В нем отображается список форм и модулей, входящих в создаваемый проект. Модуль, упрощенно говоря, — это лист (не путать с рабочим листом), в котором набирается код программы. Двойным щелчком на значке модуля в окне Project можно открыть соответствующий модуль. В VBA каждый рабочий лист, как и рабочая книга, имеет собственный модуль. Для того чтобы добавить стандартный модуль в проект, необходимо в меню редактора выбрать *Insert — Module* (Вставка модуля).

Рассмотрим пример создания простейшей программы для вывода диалогового окна с надписью «Hello, world!». Для этого необходимо в модуле ввести следующий код:

```
Sub Hello ()  
MsgBox "Hello, world!"
```

После нажатия Enter текст автоматически дополнится еще одной строкой (End Sub), и программа готова. Запустить ее можно, выбрав в меню *Run — Sub/UserForm* (или нажать F5). Результат работы и окно редактора представлены на рисунке 23.

В только что созданной программе первая и третья строки — это первая и последняя инструкции *процедуры*, написанной на языке VBA. Процедура — это минимальная семантически законченная программная конструкция, допускающая выполнение. Процедура имеет следующий синтаксис:

```
Sub name (arglist)
    [statements]
End Sub
```

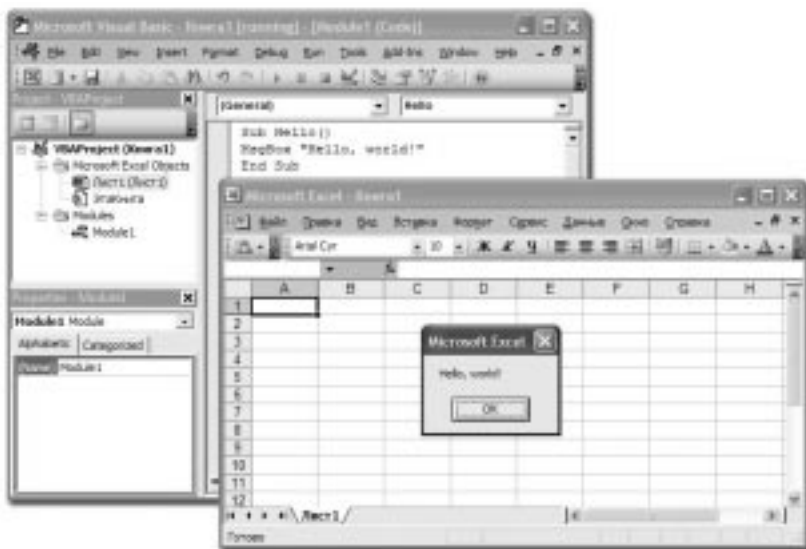


Рис. 23. Результат работы программы с окном редактора VBA

Здесь *name* — имя процедуры; *arglist* — список аргументов, который может быть и пустым; *statements* — это последовательность операторов (инструкций). Оператор *MsgBox* имеет следующий синтаксис:

```
MsgBox Text
```

В результате его действия на экране отображается диалоговое окно с кнопкой ОК и сообщением, текст которого задается параметром *Text*.

Созданную таким образом процедуру можно запустить не только из инструментальной среды VBA, но и с помощью кнопки панели инструментов Excel. Например, чтобы создать такую кнопку, необходимо выполнить следующее:

1. В MS Excel, используя команду Сервис — *Настройка* или команду Вид — *Панели инструментов* — *Настройка*, вызвать окно диалога *Настройка* (рис. 24).

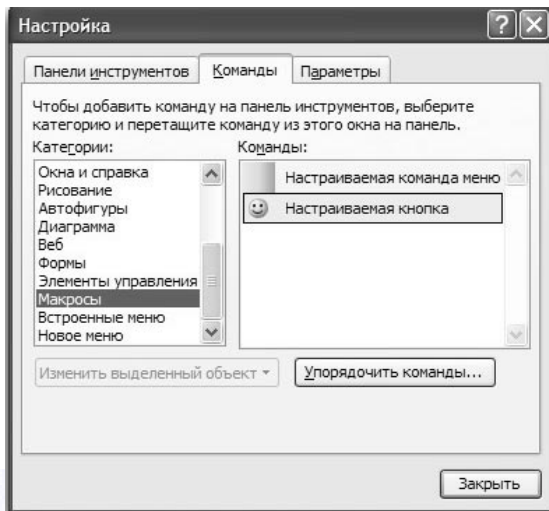


Рис. 24. Окно настройки панелей инструментов

2. Выбрать вкладку *Команды*, затем нажать на категории *Макросы*.
3. Выбрать *Настраиваемая кнопка* и с помощью мыши перетащить ее на предварительно открытую панель инструментов. Это приведет к установке кнопки на встроенной панели инструментов. При первом нажатии на нее на экране появится окно, предлагающее указать имя макроса (процедуры) из списка, который будет вызываться по нажатию на нее. В нашем примере выбираем *Hello* и нажимаем *OK*.

Другой пример использования программирования в MS Office — создание пользовательских функций. Функция отличается от процедуры тем, что помимо выполнения операторов ею возвращается некоторое значение. Собственно процедуру (*sub*) называют также процедура-подпрограмма, а функцию (*function*) — процедура-функция. Чтобы в существующий модуль с примером *Hello* добавить функцию, необходимо в меню VBA выбрать *Insert* — *Procedure*, вписать имя и указать тип создаваемой *процедуры* — функция, а после нажать *OK*. В код модуля будет добавлено описание функции.

Синтаксис описания функции:

```
Function name (arglist) [as type]
    [statements]
End Function
```

Назначение элементов описания функции аналогично описанию процедуры, но есть и отличия. Так, после указания имени функции и описания аргументов, следует указание типа возвращаемого результата (блок *as type*). Также в коде функции должен присутствовать оператор

```
name = expression,
```

где *expression* — выражение, определяющее значение функции (рис. 25).

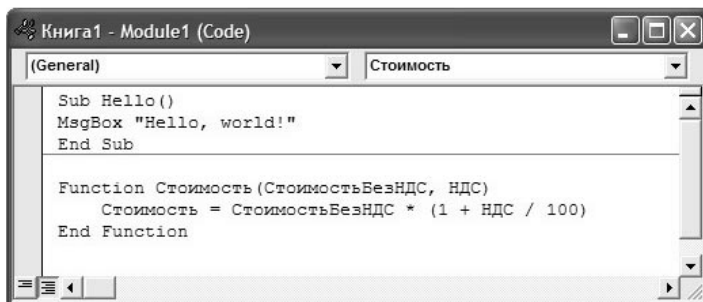


Рис. 25. Окно модуля с текстом примеров

Для закрепления сказанного можно построить простую пользовательскую функцию для расчета стоимости товара по его стоимости без НДС и значению ставки НДС. Для этого необходимо занести в код модуля следующий текст:

```
Function Стоимость (СтоимостьБезНДС, НДС)
    Стоимость = СтоимостьБезНДС * (1+НДС/100)
End Function
```

Описанная таким образом функция попадает в категорию *Определенные пользователем* списка *Категория окна Мастер функций*. А значит, предполагая, что в ячейке A2 хранится стоимость без НДС, а в ячейке B2 — НДС, можно рассчитать стоимость товара, занеся в ячейку C2, например, следующую формулу:

```
= Стоимость (A2;B2)
```

Глава 4. Элементы языка VBA

Как правило, списка аргументов процедуры недостаточно для вычислений. Тогда описание подпрограммы дополняют описанием переменных. *Переменной* называется имя, определяющее область памяти для хранения величины, которая может изменяться во время работы программы. В каждом из языков программирования определены правила, определяющие, какие имена можно давать переменным, процедурам и константам. В VBA имена должны удовлетворять следующим правилам:

- первым символом имени обязательно должна быть буква;
- нельзя использовать символы «.», «!», «@», «&», «\$», «#» и пробел;
- имя может содержать буквы, цифры и знак подчеркивания;
- имя не может содержать более 255 символов;
- имя не должно совпадать с ключевыми словами VBA;
- имена не должны повторяться на одном уровне определения, то есть в одной процедуре не могут быть объявлены две переменные с одним и тем же именем.

Перед использованием переменной в программе ее необходимо сначала объявить, то есть задать ее тип и область видимости (область использования переменной). Тип переменной определяет множество допустимых значений, которые может принимать переменная указанного типа, а также объем, который она занимает в памяти, и операции, применимые к ней.

Синтаксис объявления переменных имеет вид:

```
Dim VariableName [As] [Type],
```

где Dim — зарезервированное слово, с которого начинается описание переменных;

VariableName — имя переменной;

As Type — указание *типа* переменной (например, As Integer или As String).

VBA не требует обязательного объявления переменных. В случае если переменная не была объявлена, ей автоматически присваивается тип Variant. Этот тип является универсальным и может содержать данные различных подтипов: Integer, Long, String и т. п. Так, в примере с функцией *Стоимость* было пропущено указание типа как самой функции, так и ее аргументов. Но следует помнить, что ис-

пользование универсального типа затрудняет отладку программы. Чтобы избежать этого, рекомендуется всегда объявлять переменные и присваивать им необходимый тип. Задание переменной типа, отличного от Variant, позволяет также экономить память, так как переменные соответствующих подтипов типа Variant занимают больший объем памяти.

Таблица 3

Стандартные типы переменных VBA

Тип данных	Описание
Byte	Байт
Boolean	Логический
Integer	Целый
Long	Длинный целый
Single	С плавающей точкой обычной точности
Double	С плавающей точкой двойной точности
Currency	Денежный
Decimal	Масштабируемое целое
Date	Даты и время
Object	Ссылка на объект
String	Строка
Variant	Хранит переменную любого типа

Ниже приведен пример блока процедуры с описанием переменных.

```
Sub Test ()
Dim Count As Integer, Cash As Long
Dim Results As String
```

Для изменения типа переменной можно использовать имеющиеся в VBA функции преобразования типа. Так, функция *CStr(expression)* возвращает значение выражения (*expression*) в строковом виде. А функции *CDBl(expression)* и *CSng(expression)* могут преобразовать строковую запись числа в переменную типа Double и Single, соответственно. Приведем пример использования этих функций.

```
Dim sTXRT As String
Dim dTXRT As Double
sTXRT = "13,0"
dTXRT = CDbl(sTXRT)
```

В этом примере переменной sTXRT присваивается строковое значение «13,0», которое потом после преобразования функцией CDbl к числовому вещественному типу присваивается переменной dTXRT.

Массив (array) — это упорядоченная группа однотипных переменных, которые объединены одним общим именем. Их можно считать одной из разновидностей переменных. Отличие состоит в том, что в массивах можно хранить не одно, а несколько значений. Доступ к заданному элементу массива осуществляется с помощью индекса. Синтаксис объявления массива отличается от синтаксиса объявления переменной тем, что здесь требуется указать также размерность массива и границы изменения индексов. Ниже представлен пример объявления массива:

```
Dim B(3,3) As Single  
Dim A(11) As Integer
```

Первая инструкция объявляет двумерный массив 3x3, состоящий из действительных чисел. Вторая инструкция объявляет одномерный массив из 12 целых чисел, причем по умолчанию первый элемент массива будет A(0), а последний — A(11). Массив в программе определяется поэлементно. Например, так можно поэлементно определить массив B(2,2):

```
Dim B(2,2) As Single  
B(1,1)=2: B(1,2)=4  
B(2,1)=5: B(2,2)=1
```

Для повышения наглядности при написании программ вместо использования какого-либо постоянного значения часто используют константы. Применение констант облегчает восприятие текста программы, а также значительно упрощает отладку программы.

В VBA существуют константы двух типов:

— *встроенные константы*. Список этих констант можно посмотреть в окне просмотра объектов. Их количество определяется используемым приложением. Например, к таким константам относятся True, False, Null;

— *пользовательские константы* объявляются с помощью инструкции Const.

В момент объявления пользовательским константам присваиваются значения. Приведем пример объявления и использования пользовательской константы:

```
Sub TestSub()  
Const sngЧислоПи As Single = 3.14  
Dim sngРадиусКруга As Single  
Dim sngПлощадьКруга As Single
```

```

sngРадиусКруга = 5
sngПлощадьКруга = зпдЧислоПи * sngРадиусКруга
End Sub

```

В программах VBA можно использовать стандартный набор операций над данными, записываемых при помощи *операторов*. Например, оператор «+» выполняет операцию сложения двух чисел или выражений, являющихся операндами. Большинство операторов VBA требуют наличия двух операндов. Все простые операторы можно разделить на несколько групп:

— *арифметические операторы*. Они используются для выполнения основных арифметических операций над операндами, представляющими числовые выражения. Определены следующие арифметические операторы: сложение (+), вычитание (-), умножение (*), деление (/), возведение в степень (^) и целая часть от деления двух операндов (\);

— *логические операторы* предназначены для выполнения логических операций, таких, как логическое сложение (or), логическое умножение (and) и логическое отрицание (not);

— *операторы сравнения* (<, >, =, Like).

Особо нужно выделить так называемые структурные операторы (или операторы управления). Первый из них — оператор выбора (условного перехода). Оператор выбора задает выполнение определенных групп инструкций в зависимости от значения выражения.

Синтаксис:

```
If condition Then [statements] [Else elsestatements]
```

Если условие *condition* принимает значение True, то выполняются инструкции *statements*, иначе — инструкции *elsestatements*. Ветвь Else является необязательной. Например, если скидка (скажем 5%) применяется только к суммам больше 1000, то в VBA это можно записать так:

```
If Money>1000 Then Discount = 0.05 Else Discount = 0
```

Допускается также использование блочной формы синтаксиса, которая часто упрощает восприятие оператора условного перехода:

```

If condition Then
    [statements]
[Else
    [elsestatements]
End If

```

Дерево условий может оказаться гораздо более сложным, чем просто проверка одного условия. В этом случае используется оператор If...Then...ElseIf, который позволяет проверять множественные условия. Следующий пример показывает порядок проверки условий.

Предположим, что менеджеру по продажам необходимо разработать функцию, позволяющую рассчитать комиссионные. Процент комиссионных зависит от объема продаж и начисляется по следующему правилу:

от 0 до 999 — 8%;
от 1000 до 1999 — 10%;
от 2000 до 3999 — 12%;
свыше 4000 — 14%.

Построим для расчета комиссионных следующую функцию:

```
Function Комиссионные(Продажи)
  If Продажи<=999 Then
    Комиссионные = Продажи*0.08
  ElseIf Продажи<=1999 Then
    Комиссионные = Продажи*0.1
  ElseIf Продажи<=3999 Then
    Комиссионные = Продажи*0.12
  Else
    Комиссионные = Продажи*0.14
  End If
End Function
```

Вторая группа структурных операторов — циклы. Эти операторы предназначены для многократного выполнения группы инструкций. В VBA для организации циклов используются: For, ...Next, While...Wend, Do...Loop и For Each...Next.

Оператор For...Next

Этот оператор повторяет выполнение группы инструкций указанное число раз. Синтаксис:

```
For counter = start To end [Step step]
  [statements]
[Exit For]
  [statements]
Next [counter]
```

Оператор повторяет выполнение группы инструкций *statements*, пока счетчик *counter* изменяется от начального *start* до конечного

end значения с указанным шагом *step*. Если шаг не задан, он полагается равным 1.

Оператор For Each...Next

Этот оператор повторяет выполнение группы инструкций *statements* для каждого элемента *element* массива *group*. Синтаксис:

```
For Each element In group
  [statements]
[Exit For]
  [statements]
Next [element]
```

Оператор While...Wend

Оператор *While...Wend* выполняет последовательность конструкций *statements*, пока заданное условие *condition* имеет значение *True*. Синтаксис:

```
While condition
  [statements]
Wend
```

Оператор повтора *While...Wend* в отличие от *For...Next* работает не заданное число раз, а пока выполняется условие *condition*.

Оператор Do...Loop

Оператор *Do...Loop* повторяет выполнение набора инструкций *statements*, пока условие *condition* имеет значение *True* (случай *While*) или пока оно не примет значение *True* (случай *Until*). Синтаксис:

```
Do
  [statements]
Loop [While | Until condition]
```


Глава 5. Практика использования VBA — объекты MS Excel

VBA — пример визуального объектно-ориентированного языка программирования высокого уровня, в котором все визуальные составляющие, такие, как формы, элементы управления, меню и панели инструментов являются объектами со своими свойствами и методами.

Объект — это объединение данных с кодом, предназначенным для обработки. Каждый *объект* является экземпляром какого-либо *класса*. Класс задает свойства и поведение объектов класса. Например, каждый рабочий лист MS Excel является объектом класса *WorkSheet*. Таким образом, класс — это логическая конструкция, а объект — это его физическое воплощение. Класс содержит данные и функции, которые обрабатывают данные.

Объект сам по себе не представляет большого значения. Намного существеннее то, какие действия можно совершать над объектом и какими свойствами он обладает. *Метод* как раз и представляет собой действие, выполняемое над объектом. Синтаксис применения метода:

```
Object.Method
```

Например, метод *Delete*, примененный к рабочему листу, удаляет его из книги.

```
Worksheets(1).Delete
```

Свойство представляет собой атрибут объекта, определяющий характеристики объекта, такие, как размер, цвет, положение на экране или состояние объекта, например, доступность для пользователя или видимость. Чтобы изменить характеристики объекта, надо просто изменить значения его свойств. Синтаксис установки значения свойства:

```
Object.Property = PropertyValue
```

В следующем фрагменте кода в ячейку A1 вводится текст *Пример* посредством установки значения свойства *Value* объекта *Range*(«A1»):

```
Range("A1").Value = "Пример"
```

Значения свойств можно не только устанавливать, но и считать. В следующем примере выводится сообщение о том, введено ли в ячейку A1 слово *Пример*.

```
If Range("A1").Value = "Пример" Then  
MsgBox "В ячейку A1 введено слово Пример"
```

```
Else
```

```
    MsgBox "В ячейку A1 введено какое-то другое слово"
```

```
End If
```

Особое место среди свойств занимают те, которые сами возвращают объект. Например, свойство *Rows* объекта *Range* (диапазон ячеек) возвращает объект *Rows*, через свойства которого можно получить доступ к ячейкам указанного диапазона. Так, конструкция

```
Rcount = Range("A1:A5").Rows.Count
```

присвоит переменной *Rcount* значение 5 — количество ячеек в диапазоне A1:A5. Используем эту конструкцию для расчета суммы чисел из ячеек диапазона с именем «Данные». Напомним: чтобы присвоить имя диапазону ячеек, необходимо, выделив ячейки, в меню выбрать *Вставка — Имя — Присвоить* и в открывшемся окне занести имя диапазона (в данном примере — «Данные»). Результат расчета будет помещен в ячейку с именем «Результат».

Текст макроса будет выглядеть следующим образом:

```
Sub Test()
```

```
Dim S As Single
```

```
Dim i As Byte
```

```
For i = 1 To Range(«Данные»).Rows.Count
```

```
    S = S + Range(«Данные»)(i).Value
```

```
Next i
```

```
Range(«Результат»).Value = S
```

```
End Sub
```

В приведенном примере после объявления переменных следует оператор цикла *For...Next*, который выполнит операцию $S = S + \text{Range}(\text{«Данные»})(i).Value$ столько раз, сколько ячеек содержит диапазон «Данные». Оператор $\text{Range}(\text{«Результат»}).Value = S$ запишет результат в ячейку рабочего листа.

Глава 6. Программное обеспечение для решения финансово-экономических задач

В данной главе будут рассмотрены программные продукты, предназначенные для комплексной автоматизации деятельности предприятия, — корпоративные информационные системы, представленные на российском рынке.

В начале 90-х гг. зародился отечественный рынок экономических программ. Эти программы, выйдя из стен вычислительных центров и отделов АСУ, попали непосредственно на рабочий стол экономиста, бухгалтера или руководителя и стали рассматриваться как продукт, товар. И многие пользователи уже понимают разницу между надежной, мощной, устойчивой в финансовом отношении фирмой-разработчиком и разработчиком-одиночкой, между фирменным отработанным «коробочным» продуктом с хорошей документацией, гарантированной поддержкой в самых дальних регионах и постоянно дорабатываемой кустарной узкопрофильной разработкой местного значения. Не претендуя на полноту, рассмотрим, тем не менее, программные продукты двух крупнейших производителей ПО в России.

1С:Предприятие 8

Система программ «1С:Предприятие 8» (рис. 26) предназначена для решения широкого спектра задач автоматизации управления и учета на современном предприятии.

Руководитель может выбрать прикладное решение, которое соответствует актуальным потребностям предприятия. По мере изменения требований бизнеса и законодательства прикладные решения гибко перенастраиваются на новые условия. Решения 1С позволяют легко увеличивать число пользователей. Прикладные решения «1С:Предприятие 8» отличаются быстрыми темпами внедрения, в том числе поэтапного.

В зависимости от вида деятельности, отрасли, размера и структуры бизнеса задачи управления и учета на предприятиях могут существенно отличаться. Система программ 1С позволяет построить индивидуальную информационную систему предприятия, учитывающую все эти особенности. Внедрение и сопровождение прикладных решений могут выполнить партнеры 1С или ИТ-специалисты самого предприятия.

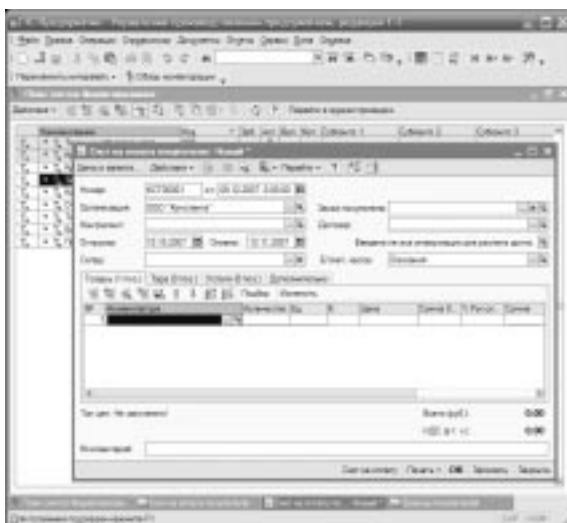


Рис. 26. Внешний вид приложения 1С:Предприятие 8

Все эти преимущества обеспечивает прогрессивная архитектура «1С:Предприятия 8»: единая технологическая платформа и прикладные решения на ее основе.

Отличительные качества такого подхода: гибкость решений, их высокая масштабируемость, возможность быстрого поэтапного внедрения.

Высокая степень адаптации «1С:Предприятия 8» к условиям конкретного предприятия сочетается со всеми преимуществами массового проверенного продукта: централизованным совершенствованием технологий, оперативным внесением изменений, связанных с законодательной базой, мощной методологической и технологической поддержкой известной фирмы.

На платформе «1С:Предприятие 8» разработаны прикладные решения различных задач:

— «1С:Управление производственным предприятием 8» и «1С:Управление торговлей 8», отраслевые и специализированные решения для промышленности, торговли, энергетики, строительства, транспорта, связи, страховых организаций, некоммерческих организаций — комплексные интегрированные системы, разработанные для автоматизации управления и учета предприятий разного масштаба.

— «1С:Бухгалтерия 8», «1С:Зарплата и Управление Персоналом 8» предназначены для автоматизации учета на предприятиях самого широкого круга: от маленьких фирм до крупных холдингов.

Продукт «1С:Управление производственным предприятием 8» отличается прежде всего своей широкой функциональностью и является комплексным прикладным решением, охватывающим основные контуры управления и учета на производственном предприятии. Решение позволяет организовать информационную систему, соответствующую корпоративным, российским и международным стандартам и обеспечивающую следующую финансово-хозяйственную деятельность предприятия:

— управление производством (планирование производства, управление затратами и расчет себестоимости, управление данными об изделиях);

— управление основными средствами и планирование ремонтов;

— управление финансами (бюджетирование, управление денежными средствами, управление взаиморасчетами, бухгалтерский и налоговый учет, учет по МСФО, формирование консолидированной отчетности);

— управление складом (запасами);

— управление продажами;

— управление закупками;

— управление отношениями с покупателями и поставщиками;

— управление персоналом, включая расчет заработной платы;

— мониторинг и анализ показателей деятельности предприятия.

«1С: Управление производственным предприятием 8» создает единое информационное пространство для отображения финансово-хозяйственной деятельности локального предприятия, холдинга или сетевой структуры, позволяя вести сквозной учет по всем структурным единицам. В то же время четко разграничивается доступ к хранимым сведениям, а также возможности тех или иных действий в зависимости от статуса работников.

«1С:Управление торговлей 8» — современный инструмент повышения эффективности торгового бизнеса.

«1С:Управление торговлей 8» в комплексе решает задачи управленческого и оперативного учета, анализа и планирования, автоматизирует торговые, финансовые и складские операции; обеспечивает современный уровень управления предприятием.

- управление продажами;
- управление поставками;
- управление складскими запасами;
- управление заказами;
- управление отношениями с клиентами (CRM);
- управление товарооборотом предприятия;
- планирование продаж и закупок;
- анализ цен и управление ценовой политикой;
- мониторинг и анализ показателей торговой деятельности.

«1С:Бухгалтерия 8» — универсальная программа для автоматизации бухгалтерского и налогового учета, включая подготовку обязательной отчетности.

Весь учет в одной программе — «1С:Бухгалтерия 8» предназначен для ведения учета различных видов деятельности в любых коммерческих организациях. В одной программе поддерживаются все системы налогообложения: общая, упрощенная, система налогообложения в виде единого налога на вмененный доход.

— учет хозяйственной деятельности нескольких организаций в единой информационной базе;

- учет «от документа» и типовые операции;
- валютные операции;
- партионный учет (ФИФО, ЛИФО, по средней себестоимости);
- складской учет;
- учет торговых операций;
- учет операций с денежными средствами;
- учет основных средств и нематериальных активов;
- учет производства;
- возможность сдачи отчетности в электронном виде;
- возможность обновления форм отчетности через Интернет;
- новая версия самой популярной бухгалтерской программы.

«1С:Зарплата и Управление Персоналом 8» — автоматизация задачи расчета заработной платы персонала и управления кадровой политикой фирмы. Характеристики продукта:

— ведение учета нескольких организаций в единой информационной базе;

- планирование потребностей в персонале;
- решение задач обеспечения бизнеса кадрами — подбор, анкетирование и оценка;

- управление компетенциями, обучением, аттестациями работников;
- управление финансовой мотивацией персонала;
- эффективное планирование занятости персонала;
- учет кадров и анализ кадрового состава;
- исчисление регламентированных законодательством налогов и взносов с фонда оплаты труда; подготовка регламентированной отчетности.

Галактика ERP

«Галактика» — многопользовательская сетевая система управления деятельностью предприятия (корпорации) решает задачи оперативного управления, бухгалтерского учета, планирования, контроля и анализа (рис. 27). Основные характеристики программного продукта:

- использование технологий интерактивной аналитической обработки данных (OLAP) с возможностью детализации уровней аналитики;

- соответствие функциональности, технологичности и степени интеграции системы современным концепциям ERP (*Enterprise Resource Planning* — «планирование ресурсов предприятия»), CSRP (*Custom Synchronized Resource Planning* — «планирование ресурсов, синхронизированное с покупателем»), SEM (*Strategic Enterprise Management* — «стратегическое управление предприятием»), а также стандартам открытых систем;

- сочетание высокой степени интеграции с гибкостью — система строится в виде набора взаимосвязанных, но относительно независимых компонентов, которые приобретаются в составе и порядке, удобном конкретному клиенту;

- обеспечение настройки Галактики на конкретную отрасль, регион, особенности бизнеса. Независимость прикладных свойств системы от программно-аппаратной платформы, гарантированная работа системы в различных операционных средах (Windows 95, 98, 2000, Windows NT, Windows XP и т. д.) и с различными СУБД — Oracle, MS SQL, Pervasive SQL (Btrieve).

Система «Галактика» обеспечивает функционирование всех подразделений территориально-рассредоточенных компаний в едином информационном пространстве.

В части оперативного управления предприятием система «Галактика» обеспечивает решение следующих задач:

- управление договорными обязательствами;
- управление закупками и материально-техническое снабжение;
- управление продажами, сбыт и реализация продукции;
- учет наличия и движения материальных ценностей (склад);
- управление продажами через торговый зал, использование интеллектуальных кассовых аппаратов;
- контроль расчетов с поставщиками и получателями;
- управление бюджетами, платежный календарь, финансовый анализ;
- управление производством и др.

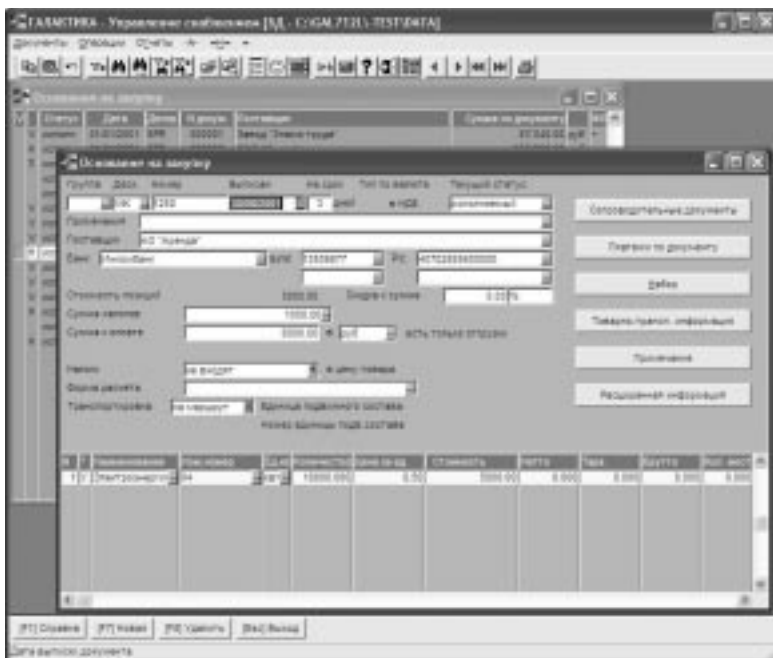


Рис. 27. Внешний вид приложения

Многие учетные задачи бухгалтерского и материального учета, налогового планирования и др. решаются без дополнительных затрат путем обработки уже введенных для целей оперативного учета документов.

Система имеет модульную структуру; модули, в свою очередь, объединены в функциональные контуры (рис. 28).

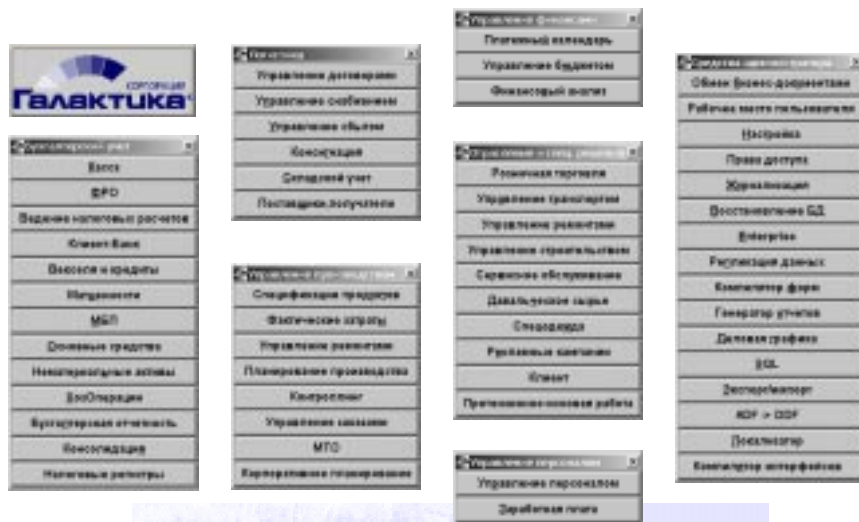


Рис. 28. Контуры и модули системы

В каждом модуле присутствуют функции, предназначенные, с одной стороны, для использования как непосредственными исполнителями, так и управленцами различного уровня, а с другой — для решения задач, относящихся к различным видам управленческой деятельности.

В рамках системы решаются такие задачи планирования, как финансовое планирование, построение бюджета, хозяйственное планирование, планирование маркетинговых кампаний, техническая подготовка и планирование производства, планирование себестоимости продукции, оценка необходимых ресурсов, баланс мощностей. Решаются и многочисленные задачи контроля: контроль выполнения планов, проектов, контроль исполнения бюджета, контроль взаимных финансовых обязательств, контроль использования ресурсов, контроль неликвидов, сверхнормативных запасов, дефицитных позиций, контроль качества, отработка возвратов и рекламаций, контроль исполнительской дисциплины.

Система обеспечивает реализацию ряда необходимых предприятию задач анализа:

- выполнения планов и использования ресурсов;
- финансовой и хозяйственной деятельности;
- эффективности использования оборотных средств;
- фактических затрат (себестоимости) и отклонение их от плановых значений;
- эффективности маркетинговых мероприятий и рекламных кампаний;
- факторов, определяющих качество и причины рекламаций и возвратов.

Среди других важных особенностей системы «Галактика» стоит выделить следующие:

- возможность ведения учета как по отечественным стандартам, так и в соответствии со стандартами GAAP;
- гибкая структура справочников и каталогов, поддерживающих иерархические структуры;
- поддержка импорта-экспорта документов и данных в формате DBF, TXT, XML;
- поддержка технологии автоматической идентификации товаров по штрих-коду, проведение инвентаризации, автоматическое формирование спецификации документов с помощью сканеров штрих-кода и др.

Допустимо как изолированное использование отдельных модулей, так и их произвольные комбинации, в зависимости от производственно-экономической необходимости.

Функциональный состав системы Галактика позволяет для любого предприятия определить набор компонентов, обеспечивающий решение задач управления хозяйственной деятельностью в трех глобальных разрезах: по видам ресурсов, по масштабам решаемых задач (уровню управления), видам управленческой деятельности.

При четко налаженной технологической схеме эксплуатации системы Галактика каждый работник выполняет определенные для него инструкции действия, получая информацию в объеме, необходимом и достаточном для осуществления своих должностных обязанностей.

Несколько слов о *настройке* системы. Система «Галактика» тиражно-заказной продукт. Этап настройки является обязательным при вводе системы в эксплуатацию, модуль Настройка входит в любой

комплект поставки. В процессе настройки выполняется первичное заполнение основных каталогов, классификаторов и справочников, которые составляют единую информационную базу и используются всеми модулями системы. Дальнейшее пополнение классификаторов происходит в процессе оперативной работы.

Параметры общесистемной настройки позволяют выбрать из поставляемых с системой алгоритмов выполнения различных функций те, которые максимально соответствуют специфике данного предприятия. Параметры настройки пользователя определяют особенности работы конкретного пользователя. Часть из них, касающаяся прав доступа к документам и возможностей их редактирования, устанавливается только администратором системы.

Важным элементом подготовки системы к работе является настройка аналитик. Аналитики используются как традиционным образом для задания разрезов по счетам бухгалтерского учета и статьям бюджетов, так и для создания шаблонов описаний материальных ценностей с последующим уточнением свойств конкретной материальной ценности, отображения дополнительных полей в документах и др.

Кроме того, большинство модулей имеет собственный блок настройки. В него, в частности, включены функции настройки универсальных иерархических отчетов («Управление снабжением», «Управление сбытом», «Складской учет»), типовых хозяйственных операций («Хозоперации»), формата входящих и исходящих документов («Обмен бизнес-документами») и т. д. Наибольшие возможности по настройке системы предоставляются в модулях контура *Управления производством*, которые позволяют конструировать внешний вид документов и создать алгоритмы расчетов с учетом специфики конкретного предприятия.

К настроечным функциям можно отнести также разграничение прав доступа пользователей и настройку корпоративного межофисного обмена данными, если у организации есть удаленные подразделения. Эти функции выполняются с помощью специальных модулей «Права доступа» и «Репликация данных», входящих в контур *Системного администрирования*.

РЕЗЮМЕ

Понятие «*программное обеспечение*» (*software*) вошло в жизнь с развитием компьютерной индустрии. *Программный продукт* — это совокупность отдельных программных средств, их документации, гарантии качества, рекламных материалов, мер по обучению пользователей, распространению и сопровождению готового программного обеспечения (ПО).

Наиболее известны классификации программных продуктов по сфере применения, по источнику финансирования и масштабу.

Инженерные методы и средства индустриального создания ПО, появившиеся в 70-х гг. прошлого столетия, носят общее название «*программная инженерия*». Понятие *жизненного цикла* программного обеспечения (ЖЦ ПО) является одним из базовых в программной инженерии. Жизненный цикл программного обеспечения определяется как период времени, который начинается с момента принятия решения о необходимости создания ПО и заканчивается в момент его полного изъятия из эксплуатации.

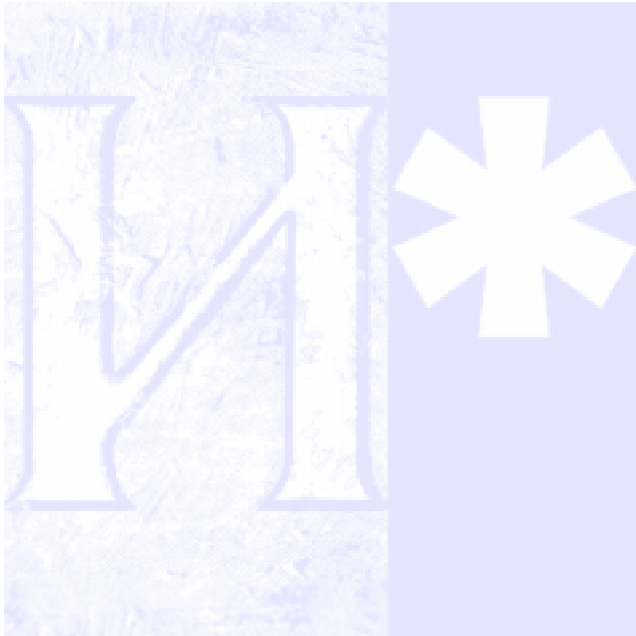
В данном курсе рассмотрены две наиболее известные модели жизненного цикла — каскадная и спиральная (итерационная). Основное отличие между ними — в выборе момента времени для перехода с этапа на этап. В каскадной модели этот момент соответствует завершению работ на текущем этапе, в спиральной модели — когда вышли отпущенные сроки.

Наиболее функционально развитой на сегодня технологией разработки приложений является технология RAD (Rapid Application Development). Один из примеров реализации этой технологии — среда программирования VBA (Visual Basic for Application), интегрированная в пакет офисных программ MS Office. Использование VBA позволяет расширить возможности компонент MS Office путем создания пользовательских процедур (макросов) и функций.

Рынок отечественных программных продуктов экономического профиля широк и разнообразен. В рамках курса были рассмотрены два из них — продукты компании «IC» и «Галактика». Не смотря на существенные различия между ними, можно указать на ряд общих черт. Оба продукта относятся к числу систем *комплексной* автоматизации деятельности предприятия, оба имеют модульную структуру, оба реализуют наиболее современные стандарты в управлении предприятием.

Вопросы для самопроверки:

1. Какие существуют классификации программного обеспечения?
2. Какие годы можно назвать ключевыми в развитии программной инженерии?
3. Что такое жизненный цикл программного обеспечения, какие процессы в него входят?
4. Какие модели жизненного цикла имеют сейчас наибольшее распространение?
5. Что отличает реальную каскадную модель жизненного цикла от идеальной?
6. Что такое CASE-средства разработки приложений?
7. Как можно использовать пользовательские функции и процедуры, написанные на языке VBA?
8. Какие операторы VBA называют структурными?



ЗАКЛЮЧЕНИЕ

В настоящее время компьютерная промышленность является, пожалуй, наиболее динамично развивающейся. На фоне стремительно совершенствующихся технологий наблюдается рост производственных мощностей, все больше людей оказываются втянутыми в гонку за мегагерцами и мегабайтами. В торговых салонах конфигурация компьютера в среднем за полгода перемещается с верхней строчки прайс-листа на нижнюю.

На рынке программных продуктов постоянно появляются новые разработки экономического назначения или модернизируются существующие. Они отличаются как своими возможностями, так и сложностью, и исполнением. В настоящее время разработчики экономических программных продуктов все больше обращают внимание на расширение функциональных возможностей своих систем в области интеллектуальной обработки данных, поддержки принятия управленческих решений, мультимедиа, визуализации данных, методов общения человека с компьютером и других наукоемких технологий. Возрастают сложность и темпы программных проектов. В связи с этим все больше внимания уделяется технологиям индустриальной разработки, основанным на CASE-системах, технологиях RAD, использовании готовых компонентов, представленных выше.

В общем, компьютерная индустрия переживает времена огромных перемен. Успеть за всем этим вряд ли получится, особенно, если профессионально заниматься другими задачами. Однако знание общих принципов хранения и обработки информации, принципов построения и функционирования компьютерных систем будет не только полезно для специалистов экономического профиля, но и предоставит возможность сориентироваться в этом бесконечно изменчивом мире компьютерных новинок.

ПРАКТИКУМ

В данной работе при помощи специализированного программного обеспечения проводится исследование характеристик составных частей персонального компьютера; предлагается оценить влияние частных свойств каждой подсистемы на интегральную оценку производительности ПК.

Задание: С использованием сервисной программы SiSoftware Sandra составить описание аппаратной части двух персональных компьютеров, применяя инструмент «Создание отчета»; сравнить результаты и на основе этого сравнения сделать выводы, какие характеристики ПК влияют на его производительность.

Порядок выполнения работы:

1. Необходимо загрузить файл с установкой программы SiSoftware Sandra (предпочтительно бесплатную версию Lite — объем дистрибутива порядка 14 МБ) с сайта разработчика — <http://www.sisoftware.net>.

2. Запустив инсталляцию и следуя указаниям на экране, установить программу на компьютер (операцию установки необходимо выполнить на обоих компьютерах, которые будут участвовать в исследовании).

3. После установки приступить к последовательному тестированию персональных компьютеров. Для этого, запустив программу, обратиться к вкладке «Инструменты».

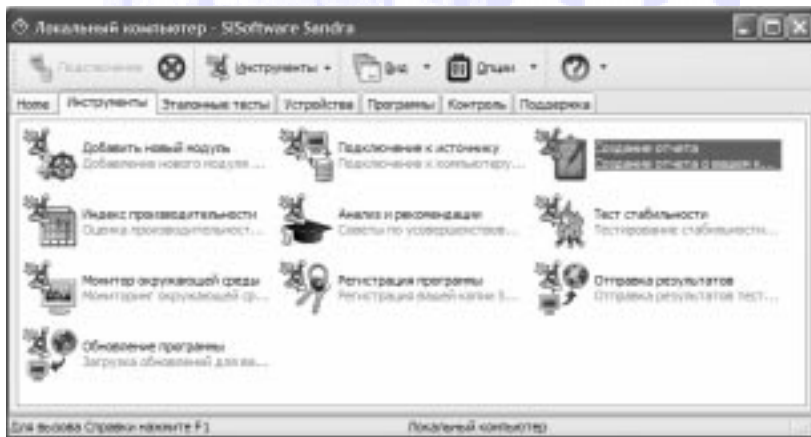


Рис. 29. Начало работы в сервисной программе SiSoftware Sandra

Запустить инструмент «Создание отчета» и последовательно пройти шаги мастера.

На первом шаге мастера следует указать «Выбрать параметры и составить отчет» как способ создания отчета. На втором шаге поставить флажок только рядом с модулем «Информация о системе» — его возможностей хватает для выполнения данной практической работы. На третьем шаге — установить флажки согласно рисунку 30.

На шагах 4, 5 убрать все флажки, а шестой шаг пройти, ничего не меняя. На седьмом шаге выбрать вариант «Сохранить на диск». На шаге 8, указав формат отчета как «Формат Web» или «Стандартный», выбрать в поле *кодировка* значение ANSI. На последнем шаге можно изменить наименование файла и папку для его сохранения (по умолчанию он создается в папке Мои документы).

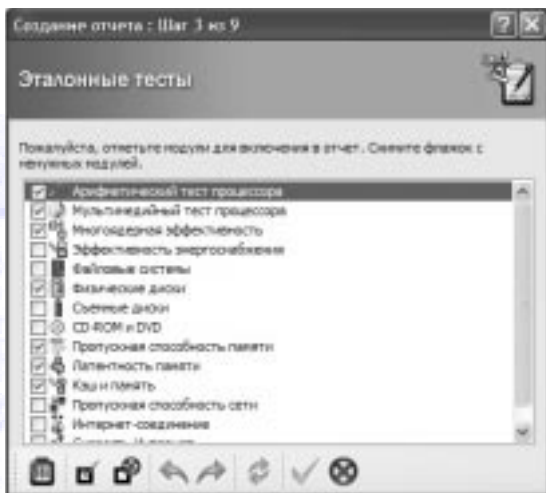


Рис. 30. Выбор эталонных тестов для тестирования

4. Подготовить отчет по практической работе. Отчет, выполненный в текстовом редакторе MS Word, должен включать в себя все полученные при помощи SiSoftware Sandra результаты.

Здесь необходимо помнить, что отчет содержит как результаты эталонных тестов производительности устройств (рис. 31), так и значения параметров комплектующих персонального компьютера.

В конце отчета необходимо сделать выводы о превосходстве одного персонального компьютера над другими, ответив на следующие вопросы:

— во сколько раз выше производительность одного ПК по отношению к другому (привести результаты для MIPS и MFLOPS);

— во сколько раз выше числовые значения характеристик комплектующих, входящих в состав ПК (привести результаты, по меньшей мере, для пяти различных параметров).



Рис. 31. Пример анализа отчета

ЗАДАНИЯ ДЛЯ КОНТРОЛЯ

Тесты для самоконтроля

1. Какая фирма является разработчиком персонального компьютера «Альтаир-8800», который по мнению американцев является первым персональным компьютером:

- a) Intel;
- b) Apple Computers;
- c) MITS;
- d) IBM.

2. Процессор какой фирмы использовался в компьютере «Альтаир-8800»:

- a) IBM;
- b) Intel;
- c) Apple Computers;
- d) AMD.

3. Для ввода данных в этом компьютере использовались:

- a) перфокарты;
- b) клавиатура;
- c) дигитайзер;
- d) тумблеры.

4. Первый европейский компьютер назывался:

- a) ZX-Spectrum;
- b) Macintosh;
- c) Alpha;
- d) Sinclair.

5. Этап развития компьютеров, характеризующийся переходом к интегральным микросхемам с малой степенью интеграции (порядка десяти транзисторов на одном кристалле), относят:

- a) к 1-му поколению;
- b) 2-му поколению;
- c) 3-му поколению;
- d) 4-му поколению.

6. Оперативная память DDR2, работающая на частоте шины 400 МГц соответствует спецификации:

- a) PC 3200;
- b) PC 4200;
- c) PC 5300;
- d) PC 6400.

7. Какие из перечисленных ниже устройств не размещены на системной плате:

- a) процессор;
- b) микросхема BIOS;
- c) слоты;
- d) винчестер.

8. Размер поддерживаемой процессором памяти называется:

- a) внутренней разрядностью данных;
- b) степенью интеграции чипа;
- c) адресацией памяти;
- d) тактовой частотой.

9. Аббревиатура, используемая для обозначения модулей памяти с двухсторонними контактами, выглядит как:

- a) DIMM;
- b) SCSI;
- c) CPU;
- d) DDR RAM.

10. Интерфейс высокопроизводительных винчестеров, аббревиатура которого понимается как «системный интерфейс малых компьютеров»:

- a) IDE;
- b) SCSI;
- c) UDMA;
- d) ATA/33.

11. Что не относится к характеристикам винчестера:

- a) объем диска;
- b) скорость считывания;
- c) интерфейс;
- d) точность позиционирования считывающей головки.

12. Какое из приведенных чисел не является характерным значением скорости вращения дисков винчестера:

- a) 3200;
- b) 4500;
- c) 7200;
- d) 10000.

13. Оптические диски, использующие синий лазер с длиной волны 405 нм, называются:

- a) CD-RW;
- b) DVD-R(RW);
- c) DVD+R(RW);
- d) HD DVD.

14. DVD-9, вмещающий информацию объемом до 8,5 ГБ, относится к типу:

- a) SS/SL;
- b) SS/DL;
- c) DS/SL;
- d) DS/DL.

15. К числу последовательных интерфейсов не относится:

- a) USB;
- b) SCSI;
- c) SATA;
- d) FireWire.

16. В качестве интерфейса для подключения жесткого диска (внешнего или внутреннего) не может использоваться:

- a) USB;
- b) PCI-E;
- c) SATA;
- d) EIDE.

17. К числу локальных шин можно отнести:

- a) IDE;
- b) SCSI;
- c) PCI-E;
- d) FireWire.

18. Модем, использующий абонентскую линию для пересылки цифровой информации одновременно с традиционной телефонной связью, относится:

- a) к кабельным модемам;
- b) ADSL-модемам;
- c) GPRS-модемам;
- d) факс-модемам.

19. Компонент, не входящий в определение программного продукта:

- a) программные средства;
- b) гарантии качества;
- c) меры по обучению пользователей;
- d) технические средства.

20. К какому классу в классификации программного обеспечения по сфере применения относят инструментальные средства:

- a) системному ПО;
- b) инструментальному ПО;
- c) занимает промежуточное место;
- d) не участвует в данной классификации.

21. По источнику финансирования программное обеспечение ПК делится на два класса:

- a) вертикальное и горизонтальное;
- b) локальное и глобальное;
- c) системное и прикладное;
- d) настольное и групповое.

22. Полное описание системы с определенной точки зрения называется:

- a) жизненным циклом;
- b) программной инженерией;
- c) моделью;
- d) документированием.

23. Жизненный цикл — период времени, который начинается с момента принятия решения о необходимости создания ПО и заканчивается в момент:

- a) полного изъятия ПО из эксплуатации;
- b) начала эксплуатации ПО;
- c) прекращения моделирования ПО;
- d) приобретения ПО.

24. Процесс жизненного цикла, включающий в себя работы по внедрению компонентов программного обеспечения в эксплуатацию, в том числе конфигурирование базы данных и рабочих мест пользователей, обеспечение эксплуатационной документацией, проведение обучения персонала и т. д. называется:

- a) верификацией;
- b) эксплуатацией;
- c) сопровождением;
- d) управлением конфигурацией.

25. Процесс ЖЦ ПО, позволяющий организовывать, систематически учитывать и контролировать внесение изменений в ПО на всех стадиях ЖЦ называется:

- a) эксплуатацией;
- b) сопровождением;
- c) управлением конфигурацией;
- d) сопровождением.

26. Разбиение разработки на этапы, где переход с одного этапа на следующий происходит только после того, как будет полностью завершена работа на предыдущем, характеризует:

- a) спиральную модель ЖЦ ПО;
- b) структуру процессов жизненного цикла ПО;
- c) стандарт ISO/IEC 12207;
- d) каскадную модель ЖЦ ПО.

27. Правильная последовательность начальных этапов разработки ПО в каскадной модели жизненного цикла:

- a) формирование требований, проектирование, реализация;

- b) проектирование, реализация, формирование требований;
- c) проектирование, тестирование, реализация, эксплуатация;
- d) тестирование, реализация, сопровождение, формирование требований.

28. В названии технологии *Rapid Application Development (RAD)* первое слово означает:

- a) технологичный;
- b) быстрый;
- c) компьютерный;
- d) структурированный.

29. Компоненты, не входящие в состав CASE-средств:

- a) графические средства анализа и проектирования;
- b) средства разработки приложений;
- c) полнофункциональная СУБД;
- d) средства управления проектом.

30. Неверное утверждение:

- a) VBA — язык визуального программирования;
- b) VBA нельзя запустить из Word;
- c) VBA интегрирован в MS Office;
- d) VBA разработан в Microsoft.

31. Для того, чтобы запустить редактор VBA, можно:

- a) в приложении MS Office нажать Alt-F10;
- b) в MS Excel выбрать Сервис — Макрос — Редактор Visual Basic;
- c) выбрать команду Пуск — Программы — Visual Basic for Application;
- d) в приложении MS Office выбрать Вид — Редактор VBA.

32. В VBA минимальная семантически законченная программная конструкция, допускающая выполнение, называется:

- a) переменной;
- b) процедурой;
- c) функцией;
- d) оператором.

33. Процедура *MsgBox Text* выполняет следующее действие:

- a) отображает диалоговое окно с сообщением Text;
- b) считывает с клавиатуры значение переменной Text;
- c) преобразует в текстовый формат числовую переменную;
- d) запускает почтовую программу.

34. Для создания пользовательской кнопки в MS Excel можно использовать команды меню:

- a) Сервис — Параметры;
- b) Вид — Панели инструментов — Настройка;
- c) Файл — Свойства;
- d) Сервис — Надстройка.

35. В какую категорию мастера функций попадает созданная в редакторе VBA функция:

- a) математические;
- b) информационные;
- c) логические;
- d) определенные пользователем.

36. Какие из указанных символов нельзя использовать в имени переменной:

- a) *;
- b) -;
- c) ~;
- d) @.

37. Какой из приведенных параметров не определяется выбором типа переменной:

- a) область допустимых значений;
- b) объем в памяти, занятой переменной;
- c) операции, применимые к переменной;
- d) наименование переменной.

38. Какой тип данных описывает переменную логического типа:

- a) Boolean;
- b) Decimal;
- c) Variant;
- d) Byte.

39. Функции CDbI, CSng служат:

- a) для вывода сообщений на экран;
- b) преобразования типа переменной;
- c) округления числовых переменных;
- d) считывания числовых данных с клавиатуры.

40. Строка, правильно описывающая переменную числового типа:

- a) Var Count: Integer;
- b) Sub Count As Integer;
- c) Dim A as integer;
- d) Dim B: Long.

41. Упорядоченная группа однотипных переменных называется:

- a) константой;
- b) оператором;
- c) функцией;
- d) массивом.

42. Оператор для вычисления целой части от деления двух операндов:

- a) ::; b) ^; c) \; d) /.

43. Какой из приведенных операторов не предназначен для многократного выполнения группы инструкций:

- a) Оператор If...Then; c) Оператор While...Wend;
b) Оператор For...Next; d) Оператор Do...Loop.

44. Какой из приведенных ниже операторов является оператором цикла, позволяющим выполнять циклические вычисления, пока некоторое заданное условие не примет значение True:

- a) While...Wend; c) If ... Then ... Else;
b) Do ... Loop ... Until; d) Do ... Loop ... While.

45. В VBA объединение данных с кодом, предназначенным для обработки, называется:

- a) объектом; c) свойством;
b) классом; d) методом.

46. Атрибут объекта, определяющий характеристики объекта, такие, как размер, цвет, положение на экране или состояние объекта, например, доступность для пользователя или видимость называется:

- a) объектом; c) свойством;
b) классом; d) методом.

47. Вариант, в котором в ячейку рабочего листа заносится текст «Пример»:

- a) Range(«A1»).Value = «Пример»;
b) Range(«A1»).Value = «Пример»;
c) Range(«A1»).Valume = «Пример»;
d) Range(«A1») = «Пример».

48. Из приведенных ниже решений от компании IC не относятся к числу отраслевых:

- a) «IC:Зарплата и Управление Персоналом 8»;
b) «IC:Управление торговлей 8»;
c) «IC:Предприятие 8. Полиграфия»;
d) «IC:Бухгалтерия 8».

49. Отраслевые и специальные решения от компании «Галактика», объединенные в одноименный контур, не содержат такого модуля, как:

- a) управление снабжением;
- b) розничная торговля;
- c) давальческое сырье;
- d) управление транспортом.

50. В контур Логистика программного комплекса «Галактика» не входит модуль:

- a) консигнация;
- b) хозоперации;
- c) складской учет;
- d) поставщики, получатели.

Ключи к тестам для самоконтроля

Вопрос	Правильный ответ	Вопрос	Правильный ответ
1	c	26	d
2	b	27	a
3	d	28	b
4	a	29	c
5	c	30	b
6	d	31	b
7	d	32	b
8	c	33	a
9	a	34	b
10	b	35	d
11	d	36	d
12	a	37	d
13	d	38	a
14	b	39	b
15	b	40	c
16	b	41	d
17	c	42	c
18	b	43	a
19	d	44	b
20	c	45	a
21	a	46	c
22	c	47	a
23	a	48	a
24	b	49	a
25	c	50	b

Задания для контрольных работ

Указания по выполнению контрольной работы

В данной работе рассматривается следующая упрощенная модель: государство стремится увеличить сумму налоговых поступлений в бюджет; величина налоговых отчислений зависит от величины налоговой ставки и от величины прибыли предприятия; величина прибыли, в свою очередь, зависит от рентабельности предприятия. Возникает вопрос, как при столь сложных связях установить величину налоговой ставки, чтобы получить максимум отчислений в бюджет.

Фактически, в данной работе две части. Первая часть — создать пользовательскую функцию в VBA, поскольку, как будет видно из дальнейшего, стандартными средствами MS Excel эту задачу решить намного сложнее. На основе некоторых исходных данных построить таблицу значений отчислений в бюджет, а уже по данной таблице построить график. На графике для различных значений рентабельности можно будет определить (приблизительно) оптимальную величину налоговой ставки. Вторая часть — для *другого* набора исходных данных при помощи поиска решения получить точное значение оптимальной ставки налога.

Задание:

Исследовать зависимость поступлений в бюджет от величины налоговой ставки и от рентабельности предприятия за определенный период. Найти и обосновать оптимальную величину налога сначала визуально — по диаграмме, затем при помощи инструмента *Поиск решения* (надстройка MS Excel).

Порядок выполнения работы

1. Получить исходные данные. Как видно из сказанного выше, в данной модели на величину поступления в бюджет оказывают влияние следующие параметры:

- ставка налога на прибыль, %;
- начальный капитал предприятия, руб.;
- рентабельность предприятия, %;
- период, в течение которого аккумулируются налоговые отчисления.

На этом этапе для всех вариантов заданий расчет провести для значений налоговой ставки от 10% до 80% с шагом в 10%. Для исследо-

ваний взять период в 10 лет. Начальный капитал и диапазон значений рентабельности (его нужно пройти с шагом в 5%) взять из табл. 4.

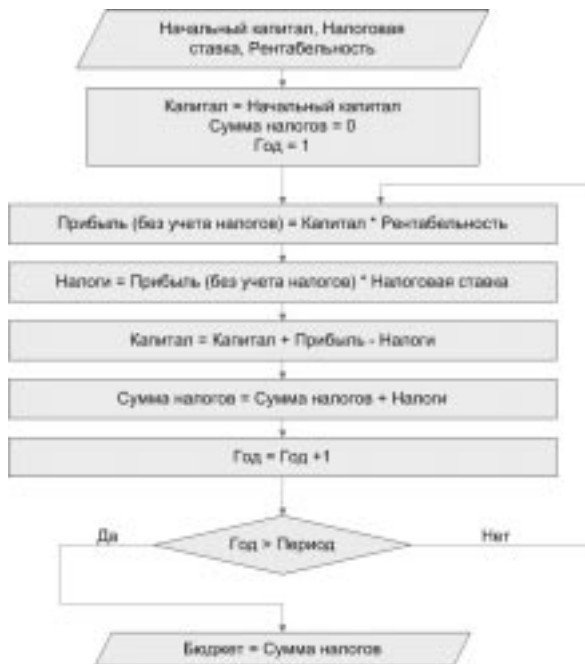
Таблица 4

Исходные данные для расчета

№ варианта	Последняя цифра номера зачетной книжки	Начальный капитал	Диапазон значений рентабельности
1	1, 2	100	10%...30%
2	3, 4	200	30%...50%
3	5, 6	300	50%...70%
4	7, 8	400	70%...90%
5	9, 0	500	90%...110%

2. Создать пользовательскую функцию для расчета поступлений в бюджет.

Алгоритм вычислений, производимых этой функцией, можно описать так:



Итеративно, на каждый год исследуемого периода, по рентабельности и текущей величине капитала предприятия рассчитывается прибыль; по ставке налога на прибыль и величине прибыли рассчитывается величина налоговых отчислений; налоговые отчисления суммируются в течение всего периода и по окончании вычислений эта сумма выводится как значение функции (рис. 32).

Рис. 32. Алгоритм вычислений налоговых отчислений

Чтобы создать такую функцию, необходимо запустить MS Excel (в последствии файл следует сохранить как «test.xls»); запустить редактор VBA и добавить новый модуль, как было показано во второй части курса; в окне модуля напечатать текст программы, согласно рисунку 33.

```

Function Выдвез(НачКап, Рентабельность, Ставка, Период)
    Капитал = НачКап
    Сумма = 0
    For Год = 1 To Период
        Прибыль = Капитал * Рентабельность
        Налог = Прибыль * Ставка
        Капитал = Капитал + Прибыль - Налог
        Сумма = Сумма + Налог
    Next Год
    Выдвез = Сумма
End Function
    
```

Рис. 33. Внешний вид программного кода пользовательской функции

После этого редактор VBA можно закрыть и вернуться в MS Excel.

3. На рабочем листе оформить таблицу подобно тому, как это сделано на рисунке 34. Пустые ячейки заполнить формулой, используя пользовательскую функцию, созданную в предыдущем пункте. Желательно использовать смешанные адреса ячеек, чтобы формулу было возможно скопировать в смежные ячейки, а не вводить заново.

	A	B	C	D	E	F	G
1	Начальный капитал		100				
2							
3	ставка	РЕНТ-%	10%	15%	20%	25%	30%
4		10%					
5		20%					
6		30%					
7		40%					
8		50%					
9		60%					
10		70%					
11		80%					
12							
13							

Рис. 34. Образец заполнения рабочего листа

4. По полученным данным построить диаграмму. Пример результата для диапазона рентабельности от 60% до 100% представлен на рисунке 35.

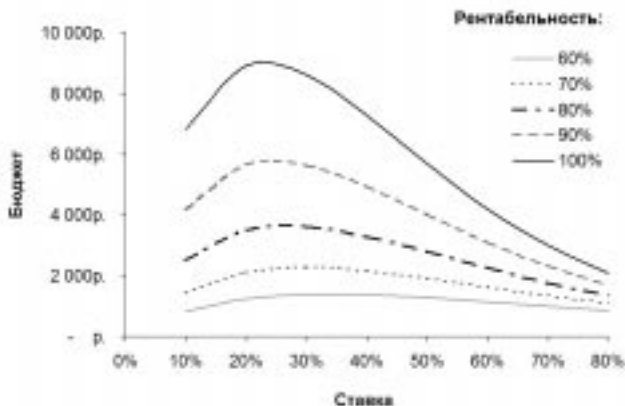


Рис. 35. Зависимость поступлений в бюджет от величины налога и рентабельности

5. Провести анализ результатов по полученным графикам. Ответить на следующие вопросы: как влияет на отчисления в бюджет увеличение рентабельности? Какие фирмы (с высокой или низкой рентабельностью) целесообразнее облагать высоким налогом?

6. Используя поиск решения, для различных значений рентабельности найти оптимальное значение налоговой ставки. Для этого на новом листе рабочей книги «test.xls» MS Excel построить таблицу зависимости поступления в бюджет, как показано ниже (табл. 5).

Таблица 5

Зависимость поступлений в бюджет от рентабельности (при оптимальной ставке налога)

Начальный капитал	Рентабельность	Ставка налога, %	Бюджет
100	30%		
100	60%		
100	90%		
100	120%		

Значения начального капитала и рентабельности взять те, что приведены в таблице 5. В графе *Ставка налога* можно предварительно занести некоторые начальные приближения. Именно эти ячейки

впоследствии необходимо указать как изменяемые (в диалоговом окне поиска решения). Графу *Бюджет* заполнить формулой с одноименной функцией. Ячейки этого столбца будут являться целевыми для поиска решений. Необходимо отдавать себе отчет, что поиск решения в этом задании необходимо запускать несколько раз — для каждой строки исходных данных. При условии правильного выполнения задания, ячейки столбца *Ставка налога* будут содержать оптимальные значения для каждого значения рентабельности.

Оформление отчета

Отчет представляется в виде файла формата MS Word и должен содержать название работы, цель, распечатку исходных данных, таблиц, диаграмм и текстов функций, созданных на VBA.



Вопросы для подготовки к зачету

1. История появления персонального компьютера.
2. Причины успеха IBM PC.
3. Классификации компьютеров.
4. Компоненты системной платы.
5. Чипсет.
6. Характеристики микропроцессоров.
7. Эталонные тесты и единицы измерения производительности.
8. Оперативная память — RAM, DRAM, DDR DRAM.
9. Производительность памяти.
10. Технологии хранения информации на жестких дисках.
11. Интерфейсы для подключения накопителей.
12. Накопители на оптических дисках.
13. Шины и интерфейсы для подключения внешних устройств.
14. Характеристики видеоадаптеров.
15. Определение и классификация программных продуктов.
16. Программная инженерия.
17. Жизненный цикл, процессы и модели жизненного цикла программного обеспечения.
18. Составляющие технологии RAD.
19. Процедуры и функции в VBA.
20. Операторы VBA.
21. Объекты MS Excel.
22. Прикладные решения от 1С.
23. Функциональные контуры системы «Галактика».

ГЛОССАРИЙ

А

Алгоритм — совокупность действий со строго определенными правилами выполнения.

Алгоритмизация — составление алгоритмов для решения поставленных задач.

Алфавит языка программирования — совокупность допустимых символов языка.

Архитектура клиент-сервер — способ организации взаимодействия программ или компонент многокомпонентной программы. А именно, имеется программа или компонента программы, называемая сервером, и одна или несколько других программ или компонент, называемых клиентами. Клиент имеет возможность асинхронно для сервера инициировать выполнение процедур сервера и получать результаты их выполнения.

Б

База данных (БД) — поименованная, целостная, единая система данных, организованная по определенным правилам, которые предусматривают общие принципы описания, хранения и обработки данных.

Базовое приложение (для офисного приложения) — то офисное средство разработки, которое использовано для разработки данного приложения.

Быстрая разработка приложений — обобщающее название для совокупности характеристик методов и средства разработки приложений, которые обеспечивают наиболее быстрое создание прототипа приложения.

В

Вертикальное приложение — приложение, которое сделано по конкретному индивидуальному заказу.

Веха — одномоментное идентифицируемое событие в процессе разработки, сопровождающееся появлением и фиксацией некоторого отчуждаемого материала (документа, программы, и др.).

Визуальный язык программирования — язык программирования, при использовании которого взаимодействие пользователя с сис-

темой программирования при составлении и отладке программ реализуется диалоговыми средствами графического интерфейса пользователя. См. *графический интерфейс пользователя*.

Влияющие ячейки — ячейки рабочего листа, значения которых используются формулой в данной ячейке.

Внедренный цикл — оператор цикла, который находится внутри тела другого оператора цикла.

Внедрение приложения — процесс перевода офисного приложения из состояния разработки в состояние эксплуатации.

Вычислительная сеть — совокупность ЭВМ, объединенных сетью связи, позволяющей компьютерам проводить обмен информацией.

Г

Глобальная сеть — сеть, в которой объединены компьютеры в различных странах, на различных континентах.

Горизонтальное приложение — приложение, которое рассчитано на массовый спрос и которое можно купить в магазине.

Графический интерфейс пользователя — интерфейс пользователя, основанный на средствах машинной графики.

Групповое приложение — приложение, с которым работает группа пользователей.

Д

Деловая графика — графические образы, часто используемые в офисных приложениях.

Диалоговое окно — один из основных элементов организации интерфейса приложения.

Ж

Жизненный цикл программного обеспечения — период, который начинается с момента принятия решения о необходимости создания ПО и заканчивается в момент его полного изъятия из эксплуатации.

З

Заказчик — субъект (отдельное лицо или организация), заказывающий разработку офисного приложения.

И

Идентификатор — последовательность символов, первый символ которой обязан быть буквой.

Интеграция (систем или элементов в системе) — восстановление и (или) повышение качественного уровня взаимосвязей между элементами системы, а также процесс создания из нескольких разнородных систем единой системы, с целью исключения (до технической необходимости минимума) функциональной и структурной избыточности и повышения общей эффективности функционирования.

Интерфейс пользователя — способ и средства взаимодействия пользователя с приложением.

Информационные процессы — процессы сбора, обработки, накопления, хранения, поиска и распространения информации.

К

Класс — множество объектов, связанных общностью структуры и поведения. Любой объект является экземпляром класса.

Компьютерные продукты — аппаратные и программные средства компьютера как объекты производства и коммерческой деятельности.

Константа — элемент данных, не меняющий своего значения

Контроллер — устройство управления.

Конфигурация ПО — совокупность его функциональных и физических характеристик, установленных в технической документации и реализованных в ПО.

Коэффициент расширения — коэффициент, который определяет, как в среднем относится количество исполнимых машинных команд после перевода программы на машинный язык к количеству операторов в исходной программе. Если этот коэффициент близок к 1, то язык считают языком низкого уровня, а если существенно больше 1, то язык считают языком высокого уровня. Например, VBA — это язык высокого уровня, а ассемблер — низкого.

Кэш — сверхоперативное запоминающее устройство; реализует быстродействующую буферную память между процессором и основной памятью.

Л

Локальная вычислительная сеть — сеть, объединяющая компьютеры в комнате или соседних помещениях.

М

Магнитные диски — средство хранения информации в ЭВМ.

Массив — совокупность однотипных индексированных переменных.

Масштаб приложения — качественная оценка размера приложения с точки зрения его использования. См. *настольное приложение*, *групповое приложение*, *приложение масштаба предприятия*.

Метод объекта — процедура или функция, которая позволяет работать с данными объекта.

Метод проектирования ПО — организованная совокупность процессов создания ряда моделей, которые описывают различные аспекты разрабатываемой системы с использованием четко определенной нотации.

Модель — абстрактное описание свойств, структуры, связей и закономерностей функционирования реальной или вымышленной системы.

Модель (ПО) — полное описание системы ПО с определенной точки зрения.

Модель ЖЦ ПО — структура, определяющая последовательность выполнения и взаимосвязи процессов, действий и задач на протяжении ЖЦ.

Модуль — часть программы, оформленная в виде, допускающей ее независимую трансляцию.

Модульное программирование — способ разработки программ, при котором программа разбивается на относительно независимые составные части — программные модули.

Мультимедийные средства — интерактивные средства, позволяющие одновременно проводить операции с неподвижными изображениями, видеофильмами, анимированными графическими образами, текстом, речевым и звуковым сопровождением. Мультимедиа являются обобщением гипертекстовых систем.

Н

Настольное приложение — автономное приложение для одного пользователя.

О

Объект — термин объектно-ориентированного программирования, которым обозначается программный объект, являющийся классом или экземпляром класса.

Объектно-ориентированное программирование (ООП) — наиболее популярная в настоящее время парадигма программирования, являющаяся развитием структурного программирования. Центральной идеей ООП является инкапсуляция, т. е. структурирование программы на модули особого вида, объединяющего данные и процедуры их обработки, причем внутренние данные модуля не могут быть обработаны иначе, как предусмотренными для этого процедурами.

Операнд — элемент данных, участвующий в операции.

Оператор — допустимое в языке программирования высокого уровня предложение, задающее целостное законченное действие или описание.

Операционная система — главная управляющая программа (комплекс программ) на ЭВМ.

Открытая архитектура компьютера IBM PC — возможность расширять конструкцию компьютера путем подключения дополнительных блоков (карт расширения) чрез специальное устройство подключения, называемое шиной расширения.

Отладка программы — специальный этап в разработке программы, состоящий в выявлении и устранении программных ошибок, факт существования которых уже установлен.

Офисное приложение — прикладное вертикальное приложение масштаба предприятия, предназначенное для ввода, вывода, хранения и обработки документов и не связанное критически с реальным временем.

Офисное средство разработки — приложение, которое одновременно является как офисным приложением, так и средством разработки офисных приложений. Например, все приложения пакета Microsoft Office 2000 являются офисными средствами разработки.

П

Переменная — элемент данных, которому присвоено имя.

Персональные компьютеры — малогабаритные вычислительные машины, предназначенные для персонального использования.

Предметная область — конкретная область человеческой деятельности, для автоматизации которой используется офисное приложение. Например, ведение документов отдела кадров предприятия.

Приведение типов — автоматическое преобразование значения одного типа в эквивалентное значение другого типа.

Прикладная программа — программа, предназначенная для решения определенных задач и получения конкретных результатов. Например, программа расчета заработной платы.

Приложение масштаба предприятия — приложение, с которым работает несколько категорий пользователей локальной или глобальной сети.

Приложение реального времени — приложение, в котором точный учет хода астрономического (реального) времени является критически важным для выполнения основных функций приложения.

Программная инженерия — 1. Совокупность инженерных методов и средств создания ПО. 2. Дисциплина, изучающая применение строгого систематического количественного (т. е. инженерного) подхода к разработке, эксплуатации и сопровождению ПО.

Программное обеспечение (программный продукт) — это совокупность отдельных программных средств, их документации, гарантии качества, рекламных материалов, мер по обучению пользователей, распространению и сопровождению готового программного обеспечения.

Продолжающаяся разработка — процесс разработки приложения, включающий в себя как разработку в обычном смысле, так и модификацию программы (и другие действия) в процессе эксплуатации.

Проект — набор программных модулей, образующих приложение или его компонент.

Процедура — минимальная семантически законченная программная конструкция, допускающая выполнение.

Процесс разработки приложения — определенная последовательность сменяющих друг друга видов деятельности, нацеленная на создание приложения.

Процесс (ЖЦ ПО) — совокупность взаимосвязанных действий, преобразующих некоторые входные данные в выходные.

Процесс создания ПО — совокупность упорядоченных во времени, взаимосвязанных и объединенных в стадии работ, выполнение которых необходимо и достаточно для создания ПО, соответствующего заданным требованиям.

Р

Разработка ПО — комплекс работ по созданию ПО и его компонентов в соответствии с заданными требованиями, включая оформление проектной и эксплуатационной документации, под-

готовку материалов, требуемых для проверки работоспособности и соответствующего качества программных продуктов, материалов, необходимых для организации обучения персонала, и т. д.

Разработчик — субъект (отдельное лицо или организация), выполняющий разработку офисного приложения по заказу. См. *офисное приложение*.

Реакция на событие — процедура, которая выполняется при возникновении события.

Репозиторий — база данных, предназначенная для хранения проектных метаданных (версий проекта и его отдельных компонентов), синхронизации поступления информации от различных разработчиков при групповой разработке, контроля метаданных на полноту и непротиворечивость.

С

Свойство объекта — данные, хранящиеся в объекте. См. *класс, объектно-ориентированное программирование*.

Системная программа — программа, предназначенная для обеспечения работы прикладных программ. Например, операционная система.

Сопровождение ПО — внесение изменений в ПО в целях исправления ошибок, повышения производительности или адаптации к изменившимся условиям работы или требованиям.

Средство разработки — приложение, используемое при разработке других приложений.

Стадия ЖЦ ПО — часть процесса создания ПО, ограниченная определенными временными рамками и заканчивающаяся выпуском конкретного продукта (моделей ПО, программных компонентов, документации), определяемого заданными для данной стадии требованиями.

Структурное программирование — исторически первая оформленная парадигма программирования.

Т

Тело процедуры — последовательность операторов в процедуре.

Тестирование — процесс исполнения программы в целях обнаружения ошибки.

Технология проектирования ПО — совокупность технологических операций проектирования в их последовательности и взаимосвязи, приводящая к разработке проекта ПО.

У

Уровень языка программирования — качественная характеристика языка программирования.

Ф

Фаза — часть процесса, во время которой выполняются определенные функции с целью достижения определенной вехи.

Форма — типовой документ, в котором меняются только некоторые поля, а основная часть остается неизменной.

Функция рабочего листа — функция, которую можно использовать в формулах на рабочем листе.

Ш

Шина — совокупность проводов, спроектированных таким образом, чтобы выполнялось соглашение о правилах передачи информации между чипами, подключенными к шине.

Э

Электронная таблица — структура данных, внешне похожая на таблицу, в ячейках которой могут храниться значения и формулы. Используется для организации вычислений и другой обработки данных на персональном компьютере. Например, Excel 2000.

Элемент управления — средство графического интерфейса пользователя, предназначенное для ввода/вывода информации и управления работой программы. Воздействие пользователя на элемент управления вызывает событие в программе.

Я

Язык программирования VBA (Visual Basic for Applications) — визуальный объектно-ориентированный язык программирования высокого уровня.

Язык программирования высокого уровня — язык программирования, в который введены элементы, допускающие описание программы в наглядном, легко воспринимаемом человеком виде, упрощающие и автоматизирующие процесс программирования.

В

BIOS (Basic Input Output System) — базовая система ввода-вывода. Программа, с которой начинается запуск ПК.

С

CASE-средство — программное средство, поддерживающее процессы жизненного цикла ПО (определенные в стандарте ISO/IEC 12207:1995), включая анализ требований к системе, проектирование прикладного ПО и баз данных, генерацию кода, тестирование, документирование, обеспечение качества, конфигурационное управление и управление проектом, а также другие процессы.

СПИСОК ЛИТЕРАТУРЫ

Основная литература

1. Бочаров Е. П. Интегрированные корпоративные информационные системы: Принципы построения. Лабораторный практикум на базе системы «Галактика»: Учебное пособие. М.: Финансы и статистика, 2005.
2. Бройдо В. Л. Вычислительные системы, сети и телекоммуникации: Учебник для вузов. 2-е изд. СПб.: Питер, 2005.
3. Вендров А. М. Проектирование программного обеспечения экономических информационных систем. М.: Финансы и статистика, 2005.
4. Гарнаев А. Ю. Excel, VBA, Internet в экономике и финансах. СПб.: БХВ-Петербург, 2002.
5. Гук М. Ю. Аппаратные средства IBM PC. Энциклопедия. 3-е изд. СПб.: Питер, 2006.

Дополнительная литература

1. Бройдо В. Л. Вычислительные системы, сети и телекоммуникации: Учебник для вузов. 2-е изд. СПб.: Питер, 2005.
2. Васильев А., Андреев А. VBA в Office 2000: Учебный курс. СПб.: Питер, 2001.
3. Цилькер Б. Я., Орлов С. А. Организация ЭВМ и систем: Учебник для вузов. СПб.: Питер, 2007.

Игорь Николаевич ГЛУХИХ
Дмитрий Витальевич СЕМИХИН

КОМПЬЮТЕРНЫЕ ПРОДУКТЫ

Учебное пособие

Редактор
Технический редактор
Компьютерная верстка
Трафаретная печать
Офсетная печать

*Ю. Ф. Евстигнеева
Н. Г. Яковенко
Т. С. Бакиева
А. В. Ольшанский
В. В. Торопов, С. Г. Наумов*

Подписано в печать 01.09.2008. Тираж 2050 экз.
Объем 7,0 усл. печ. л. Формат 60x84/16. Заказ 667.

Издательство Тюменского государственного университета
625003, г. Тюмень, ул. Семакова, 10
Тел./факс (3452) 45-56-60; 46-27-32
E-mail: izdatelstvo@utmn.ru