

**Министерство образования и науки Украины
Севастопольский национальный технический университет**



**Методические указания
к лабораторным работам по дисциплине
«Периферийные устройства ЭВМ»
для студентов
направления 6.050102 «Компьютерная инженерия»
дневного и заочного отделения**

Севастополь
2010

Методические указания к лабораторным работам по дисциплине «Периферийные устройства ЭВМ» для студентов направления 6.050102 «Компьютерная инженерия» дневного и заочного отделения / сост. Я.Ф.Яцек, К.С.Ткаченко - Севастополь: Изд-во СевНТУ, 2010. - 44 с.

Методические указания содержат требования к выполнению лабораторных работ и оформлению отчетов к лабораторным работам, варианты индивидуальных заданий, контрольные вопросы.

Методические указания рассмотрены и утверждены на заседании кафедры кибернетики и вычислительной техники, протокол № __ от __.__.2010 г.

Допущено учебно-методическим центром СевНТУ в качестве методических указаний.

Рецензент: Иванченко О.В., канд. техн. наук, доцент кафедры кибернетики и вычислительной техники

СОДЕРЖАНИЕ

1	Лабораторная работа № 1. Изучение пространства ввода-вывода BIOS	4
2	Лабораторная работа № 2. Изучение пространства ввода-вывода CMOS	6
3	Лабораторная работа № 3. Исследование работы последовательного интерфейса. Работа Mouse	7
4	Лабораторная работа № 4. Клавиатура. Скан - коды	8
5	Лабораторная работа № 5. Исследования видеоадаптера	10
6	Лабораторная работа № 6. Внешняя память. Изучение работы НГМД и НЖМД	12
7	Лабораторная работа № 7. Изучение пространства WMI	13
	Библиографический список	14
	ПРИЛОЖЕНИЕ А. Пространство ввода-вывода ПК (CMOS, BIOS)	15
	ПРИЛОЖЕНИЕ Б. Интерфейс RS-232	19
	ПРИЛОЖЕНИЕ В. Стандартные прерывания ROM BIOS	21
	ПРИЛОЖЕНИЕ Г. Скан-коды клавиатуры	23
	ПРИЛОЖЕНИЕ Д. Видеорежимы	26
	ПРИЛОЖЕНИЕ Е. Классы WMI	28

1 ЛАБОРАТОРНАЯ РАБОТА № 1

Изучение пространства ввода-вывода BIOS

Цель работы: Изучить и исследовать пространство ввода/вывода компьютерной системы с использованием CMOS.

1.1 Теоретические сведения

Базовая система ввода-вывода (BIOS - Basic Input/Output System) для ПЭВМ семейства IBM PC и IBM PS/2 - это программный интерфейс между программами и оборудованием ПЭВМ. BIOS изолирует операционную систему и прикладные программы от аппаратных особенностей конкретных устройств и позволяет программистам, пишущим на языках ассемблера, Си и т. п., выполнять операции ввода-вывода, не заботясь об адресах устройств или их аппаратных характеристиках. Кроме того, BIOS обеспечивает ряд системных услуг, например, позволяет узнать размер памяти ПЭВМ или текущее время дня. Основные функции BIOS таковы: тестирование ПЭВМ при ее включении; обеспечение управления периферийными устройствами; системные услуги.

Для доступа к BIOS используются программные прерывания. При этом каждая точка входа в BIOS использует свой вектор прерываний. Если точка входа обслуживает несколько процедур (называемых функциями BIOS), то номер функции задается в регистре AH. За процедурами BIOS закреплены прерывания Int 10h - Int 1Ah. Процедуры BIOS сохраняют значения всех регистров, кроме тех из них, в которых возвращаются значения. Области данных BIOS - это либо области ОЗУ, в которых хранится текущая информация о состоянии ПЭВМ, либо области ПЗУ, в которых записаны аппаратные характеристики устройств. Подробнее вы можете ознакомиться с этим вопросом в приложении А и В.

Во время инициализации системы BIOS опрашивает порты, к которым подключены переключки, и определяет содержимое ячеек памяти CMOS, содержащих информацию о конфигурации компьютера. Результат записывается в область данных BIOS, откуда программа может его извлечь с помощью прерываний BIOS. Сведения о наличии основных устройств компьютера записываются в область данных BIOS с адресом 0000:0410 размером в двухбайтовое слово - слово конфигурации. С помощью прерывания INT 11h программа может получить в регистре AX слово конфигурации из указанной выше области данных BIOS. Значения представлены в таблице 1.1.

На этапе инициализации BIOS записывает в свою область данных по адресу FFFFh:FFFEh байт идентификатора модели компьютера. Более подробную информацию можно получить, вызвав функцию C0h прерывания BIOS INT 15h. После выполнения прерывания регистры ES:BX будут указывать на таблицу в области ПЗУ BIOS. В этой таблице имеется более точная информация о типе компьютера, номер версии BIOS, сведения об аппаратных особенностях конкретной модели.

Таблица 1.1 – Определение конфигурации с помощью BIOS

Биты регистра АХ	Содержимое
0	В системе установлен накопитель на магнитном диске (НМД)
1	В системе установлен арифметический сопроцессор
2 – 3 (все модели, кроме IBM PS/2)	Количество банков оперативной памяти на системной плате. Для компьютера IBM PC размер одного банка равен 16 Кбайт, для IBM PC/XT – 64 Кбайт. Для определения объема оперативной памяти в более современных компьютерах следует использовать другие средства, о которых мы расскажем позже
2 (IBM PS/2)	Компьютер IBM PS/2 оборудован мышью
3 (IBM PS/2)	Не используется в IBM PS/2
4 – 5	Начальный режим видеоадаптера: 00 – EGA или VGA 01 – цветной, 40х25 10 – цветной, 80х25 11 – монохромный, 80х25
6 – 7	Количество установленных накопителей на гибких магнитных дисках (НГМД)
8	Установлен контроллер прямого доступа к памяти DMA
9 – 11	Количество установленных асинхронных последовательных портов
12	Установлен игровой порт
13	Установлен последовательный порт (только для компьютера PCjr)
14 – 15	Количество установленных параллельных адаптеров

1.2 Порядок выполнения работы

Написать программу, позволяющую по прерываниям BIOS определить набор устройств, подключенных к ПК. Произвести опрос размера ОЗУ в ПК. Определить тип жесткого диска.

1.3 Варианты заданий

Для получения сведений использовать свой домашний персональный компьютер.

1.4 Содержание отчета

Цель работы. Программная реализация опроса BIOS. Результаты выполнения программы. Выводы по проделанной работе. Контрольные вопросы (рукописно).

1.5 Контрольные вопросы

Основные функции BIOS. Определение конфигурации с помощью BIOS.

2 ЛАБОРАТОРНАЯ РАБОТА № 2

Изучение пространства ввода-вывода CMOS

Цель работы: Изучить и исследовать пространство ввода/вывода компьютерной системы с использованием стандартных прерываний BIOS.

2.1 Теоретические сведения

Для хранения конфигурация оборудования (объем памяти, количество дисководов и т. п.) используется CMOS Memory and Real Time Clock. Микросхемы CMOS RTC имеют встроенную систему контроля непрерывности питания, отслеживающую и разряд батареи ниже допустимого уровня. Достоверность информации конфигурирования проверяется с помощью контрольной суммы. Доступ к ячейкам CMOS RTC осуществляется через порты ввода/вывода 070h (индекс ячейки) и 071h (данные). Поскольку эта память имеет быстродействие порядка единиц микросекунд, между командами записи адреса и чтения/записи данных необходима программная задержка. Назначение ячеек приведено в приложении А, таблица А.2.

2.2 Порядок выполнения работы

Написать программу, позволяющую с использованием CMOS определить набор устройств, подключенных к ПК. Произвести опрос размера ОЗУ в ПК. Определить тип жесткого диска.

2.3 Варианты заданий

Для получения сведений использовать свой домашний ПК.

2.4 Содержание отчета

Цель работы. Программная реализация опроса CMOS. Результаты выполнения программы. Карта портов ввода/вывода исследуемого РС. Выводы по проделанной работе. Контрольные вопросы (рукописно).

2.5 Контрольные вопросы

Основные функции CMOS. Определение конфигурации с помощью CMOS.

3 ЛАБОРАТОРНАЯ РАБОТА № 3

Исследование работы последовательного интерфейса. Работа Mouse

Цель работы: Научиться опрашивать положение на плоскости и нажатие кнопок Mouse.

3.1 Теоретические сведения

Устройство ввода мышь (mouse) передает в систему информацию о своем перемещении по плоскости и нажатии кнопок. Обычная конструкция имеет свободно вращающийся массивный прорезиненный шарик в днище корпуса, передающий вращение на два координатных диска с фотоэлектрическими датчиками. Датчики для каждой координаты представляют собой две открытые оптопары (светодиод - фотодиод), в оптический канал которых входит вращающийся диск с прорезями. Оптопары датчиков могут быть оформлены в виде монолитных конструкций, а могут и быть просто отдельными элементами, установленными на печатной плате. По интерфейсу с компьютером различают три основных вида мышей: Bus Mouse, Serial Mouse, PS/2- Mouse.

В приложении В можно ознакомиться с программным интерфейсом мыши.

3.2 Порядок выполнения работы

Составить алгоритм работы программы, реализующей опрос мыши. Учесть опрос положения мыши на плоскости и опрос кнопок. Реализовать программу на любом языке программирования. Оформить отчет. Сделать выводы по проделанной работе.

3.3 Варианты заданий

Вариантом заданий для данной лабораторной работы является выбор элемента интерфейса для перетаскивания. Четные варианты делают с шариком, нечетные – с квадратиком.

3.4 Содержание отчета

Цель работы. Программная реализация опроса мыши. Результаты выполнения программы. Выводы по проделанной работе. Контрольные вопросы (рукописно).

3.5 Контрольные вопросы

Основные функции мыши.

4 ЛАБОРАТОРНАЯ РАБОТА № 4

Клавиатура. Скан - коды

Цель работы: Научиться производить обращение к клавиатуре через порты ввода/вывода.

4.1 Теоретические сведения

Обычно программе нужен не порядковый номер нажатой клавиши, а код, соответствующий обозначению на этой клавише (код ASCII). Код ASCII не связан напрямую со скан-кодом, так как одной и той же клавише могут соответствовать несколько значений кода ASCII в зависимости от состояния других клавиш. Например, клавиша с обозначением «1» используется еще и для ввода символа «!» (если она была нажата вместе с клавишей <Shift>). Поэтому все преобразования скан-кода в код ASCII выполняются программно. Как правило, в операционной системе MS-DOS эти преобразования выполняют модули BIOS. Для использования символов кириллицы эти модули расширяются клавиатурными драйверами, как входящими в состав локализованных версий MS-DOS, так и созданными в виде отдельных программ. Если нажать на клавишу и не отпускать ее, клавиатура перейдет в режим автоповтора. При этом в компьютер автоматически через некоторый период времени, называемый периодом автоповтора, посылается код нажатой клавиши. Режим автоповтора облегчает ввод с клавиатуры большого количества одинаковых символов. Следует отметить, что клавиатура содержит внутренний 16-байтовый буфер, через который она осуществляет обмен данными с компьютером.

4.2 Порядок выполнения работы

Составить алгоритм опроса клавиатуры. Реализовать программу на любом языке программирования. Оформить отчет. Сделать выводы по проделанной работе.

4.3 Варианты заданий

№	Задание на лабораторную работу
1	Составить программу опроса клавиатуры, показывающую какая из управляющих клавиш Alt, Ctrl, Shift является приоритетной при их одновременном нажатии с любой из клавиш клавиатуры.
2	Определить емкость буфера клавиатуры экспериментальным путем.
3	Написать программу, позволяющую определить расширенные ASCII- коды, которые доступны только через функции 00h- 02h INT 16h.
4	Написать программу, позволяющую определить расширенные ASCII- коды, которые доступны только через функции 10h- 12h INT 16h.
5	Исследовать возможные сочетания управляющих клавиш, анализируя состояние байтов управления в порядке возрастания.
6	Реализовать программу, позволяющую производить демонстрационные действия без участия оператора
7	Реализовать управления светодиодами, расположенными на лицевой панели клавиатуры

8	Реализовать опрос переключающих клавиш - <NumLock>, <ScrollLock>, <CapsLock>, <Insert>.
9	Составить программу опроса клавиатуры, показывающую какая из управляющих клавиш Alt, Ctrl, Shift является приоритетной при их одновременном нажатии с любой из клавиш клавиатуры.
10	Определить емкость буфера клавиатуры экспериментальным путем.
11	Написать программу, позволяющую определить расширенные ASCII- коды, которые доступны только через функции 00h- 02h INT 16h.
12	Написать программу, позволяющую определить расширенные ASCII- коды, которые доступны только через функции 10h- 12h INT 16h.
13	Исследовать возможные сочетания управляющих клавиш, анализируя состояние байтов управления в порядке возрастания.
14	Реализовать программу, позволяющую производить демонстрационные действия без участия оператора
15	Реализовать управления светодиодами, расположенными на лицевой панели клавиатуры
16	Реализовать опрос переключающих клавиш - <NumLock>, <ScrollLock>, <CapsLock>, <Insert>.
17	Составить программу опроса клавиатуры, показывающую какая из управляющих клавиш Alt, Ctrl, Shift является приоритетной при их одновременном нажатии с любой из клавиш клавиатуры.
18	Определить емкость буфера клавиатуры экспериментальным путем.
19	Написать программу, позволяющую определить расширенные ASCII- коды, которые доступны только через функции 00h- 02h INT 16h.
20	Написать программу, позволяющую определить расширенные ASCII- коды, которые доступны только через функции 10h- 12h INT 16h.
21	Исследовать возможные сочетания управляющих клавиш, анализируя состояние байтов управления в порядке возрастания.
22	Реализовать программу, позволяющую производить демонстрационные действия без участия оператора
23	Реализовать управления светодиодами, расположенными на лицевой панели клавиатуры
24	Реализовать опрос переключающих клавиш - <NumLock>, <ScrollLock>, <CapsLock>, <Insert>.

4.4 Содержание отчета

Цель работы. Программная реализация опроса клавиатуры. Результаты выполнения программы. Выводы по проделанной работе. Контрольные вопросы (рукописно).

4.5 Контрольные вопросы

Классификация типов клавиатур. Функциональное назначение клавиатуры. Интернационализация на примере клавиатуры.

5 ЛАБОРАТОРНАЯ РАБОТА № 5

Исследования видеоадаптера

Цель работы: Научиться использовать видеоадаптер.

5.1 Теоретические сведения

Дисплейный адаптер, как обязательный компонент PC, имеет поддержку основных функций BIOS. Эти функции выполняются через вызов программного прерывания INT 10h – видеосервиса BIOS. Видеосервис позволяет установить видеорежим (BIOS Video Mode), определяющий формат экрана. Первоначально для задания номера режима отводится один байт и режим устанавливался параметрами функции 0 INT 10h (AH=0, AL=Mode). Режимы 0 – 13h являются стандартными для адаптеров MDA, CGA, EGA, VGA. Режимы 14h – 7Fh используются с нестандартными VGA- или SVGA- расширениями BIOS, они специфичны для конкретных моделей графических адаптеров.

5.2 Порядок выполнения работы

Составить алгоритм работы графического приложения. Реализовать программу на любом языке программирования. Оформить отчет. Сделать выводы по проделанной работе.

5.3 Варианты заданий

1	Исследование процесса записи данных в видеопамять из внешнего текстового файла с предварительным предсказанием результата и соответствующим отображением в графическом режиме.
2	Считывание данных из видеопамяти и фиксирование результатов во внешний файл с указанием адресов обращения. Исследовать какое изменение данных еще не приводит к видимому изменению изображения.
3	Реализовать управление спрайтом без видимого скачкообразного изменения образа, используя задержку хода луча.
4	Реализовать управление спрайтом без видимого скачкообразного изменения образа, используя наличие обратного хода луча.
5	Оценить возможность управления графическим примитивом за время междустрочной развертки с учетом послесвечения.
6	Оценить возможность управления графическим примитивом за время междустрочной развертки с принудительным гашением примитива.
7	Перепрограммирование палитры цветов. Демонстрация палитры, задаваемой разными способами, с исходными данными, выводимыми из внешнего текстового файла.
8	Чтение и запись символа и атрибута с использованием базовых функций BIOS INT 10h
9	Чтение и запись графического пиксела с использованием базовых функций BIOS INT 10h

5.4 Содержание отчета

Цель работы. Программная реализация. Результаты выполнения программы. Выводы по проделанной работе. Контрольные вопросы (рукописно).

5.5 Контрольные вопросы

Видеоадаптер как неотъемлемая составная часть ПК. Видеоинтерфейс BIOS. Использование GDI и GDI+ в Win32 и Win64. Прямой и косвенный доступ к видеопамяти в режиме VGA и MCGA.

6 ЛАБОРАТОРНАЯ РАБОТА № 6

Внешняя память. Изучение работы НГМД и НЖМД

Цель работы: Научиться работать с НГМД и НЖМД.

6.1 Теоретические сведения

Дисковые устройства памяти поддерживаются BIOS компьютера. Поддержка заключается в предоставлении вышестоящим уровням программного обеспечения возможности чтения и записи секторов диска, форматирование трека и выполнение вспомогательных функций. Эти возможности предоставляются программным вызовом прерывания BIOS INT 13h – дискового сервиса. Традиционно дисковый сервис подразделяет физические диски на дискеты и фиксированные диски. Классы различаются по диапазонам номеров физических устройств. Для дискет отводятся номера 0 – 7Fh, а для фиксированных дисков – 80h – FFh. Программный код обслуживания гибких дисков можно вызывать и через прерывание INT 40h. При вызове дискового сервиса принимаются следующие соглашения: Номер функции задается в регистре AH. Номер диска – в регистре DL. В каждом классе устройства нумеруются с нуля. Номер цилиндра ограничен в 10 бит, два старших бита в регистре CL[7,6], а младшие 8 бит в регистре CH. Номер головки в регистре DH – 8 бит. Номер начального сектора – 6 бит в регистре CL[5:0]. Нумерация секторов начинается с единицы. Количество секторов, участвовавшее в операции в регистре AL. Результат выполнения операции определяется по флагу переноса: CF=0 – успешное выполнение операции, CF=1 – обнаружены ошибки. Код состояния возвращается в регистр AH.

6.2 Порядок выполнения работы

Составить алгоритм работы приложения. Реализовать программу на любом языке программирования. Оформить отчет. Сделать выводы по проделанной работе.

6.3 Варианты заданий

Выдается преподавателем индивидуально.

6.4 Содержание отчета

Цель работы. Программная реализация. Результаты выполнения программы. Выводы по проделанной работе. Контрольные вопросы (рукописно).

6.5 Контрольные вопросы

Эволюция НГМД и НЖМД. Понятие CHS и LBA. Ограничения адресации в зависимости от методов доступа (int 25h, int 21h, int 13h, CHS, LBA, DevIOCtrl). Предназначение MBR и BS.

7 ЛАБОРАТОРНАЯ РАБОТА № 7

Изучение пространства WMI

Цель работы: Научиться работать с Windows Management Instrumentation.

7.1 Теоретические сведения

Технология WMI (Windows Management Instrumentation) - это одно из базовых средств для централизованного управления и слежения за работой различных частей компьютерной сети под управлением Windows. Это реализация модели управления предприятием на базе Web. Задачей WBEM является разработка таких стандартов удалённого управления информационной средой предприятия, которые не зависят от конкретного оборудования, сетевой инфраструктуры, операционной системы, файловой системы и т.д. В рамках WBEM была предложена схема Common Information Model (CIM), которая представляет структуру компьютерной системы в виде единой расширяемой объектно-ориентированной модели и поддерживается в WMI. Можно автоматизировать работу с WMI с помощью сценариев Windows Script Host (WSH), используя при этом совместно с WMI и другие ActiveX-технологии (например, ActiveX Data Object (ADO) для доступа к базам данных или Active Directory Service Interface (ADSI) для работы со службами каталогов), получив таким образом мощный и удобный инструмент для системного администратора и программиста. Полное описание интерфейса WMI содержится в приложении Е.

7.2 Порядок выполнения работы

Составить алгоритм работы приложения. Реализовать программу на любом языке программирования. Оформить отчет. Сделать выводы по проделанной работе.

7.3 Варианты заданий

Выдается преподавателем индивидуально.

7.4 Содержание отчета

Цель работы. Программная реализация. Результаты выполнения программы. Выводы по проделанной работе. Контрольные вопросы (рукописно).

7.5 Контрольные вопросы

Объектная модель WMI. Организация обращения к WMI средствами языков пакета CScript (VBScript, Jscript и т.п.). Организация обращения к WMI средствами PowerShell 2.0.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Лукач Ю.С. Базовая система ввода-вывода / Ю.С.Лукач. - Свердловск: Инженерно-техническое бюро, 1990. - 250 с.
2. Сибиряков А.Е. Архитектура ввода-вывода персональных ЭВМ IBM PC / А.Е. Сибиряков.- Свердловск: Инженерно-техническое бюро, 1990. - 140 с.
3. Фролов А.В. Аппаратное обеспечение персонального компьютера / А.В. Фролов, Г.В. Фролов. - М.: Питер, 1997. - 348 с.
4. Айден К. Аппаратные средства PC / К. Айден.-СПб: BHV, 1998. - 608 с.
5. Учи Г. Персональные компьютеры для научных работников / Г.Учи - М.: Мир, 1993. - 268с.

ПРИЛОЖЕНИЕ А (обязательное).

Пространство ввода-вывода ПК IBM PC (CMOS и BIOS ROM)

Прерывания, обслуживаемые системной BIOS, перечислены ниже. Кроме них несколько векторов используются как указатели на различные структуры данных. *Внутренние прерывания:*

- Int 00h — деление на 0;
- Int 01h — пошаговый режим;
- Int 03h — точка останова;
- Int 04h — переполнение;
- Int 06h — недопустимая команда 286+;
- Int 07h — вызов отсутствующего NPU.

Аппаратные прерывания:

- Int 02h - немаскируемое прерывание;
- Int 08h - таймер 8253/8254;
- Int 09h - клавиатура;
- Int 0Ah - IRQ2/9;
- Int 0Bh - IRQ3;
- Int 0Ch - IRQ4;
- Int 0Dh - IRQ5;
- Int 0Eh - IRQ6 — контроллер гибких дисков;
- Int 0Fh - IRQ7;
- Int 70h - CMOS-таймер;
- Int 71h - IRQ9 (перенаправлено на Int 0Ah);
- Int 72h - IRQ10;
- Int 73h - IRQ11;
- Int 74h - IRQ12 (контроллер мыши PS/2);
- Int 75h - IRQ13 - исключение сопроцессора;
- Int 76h - IRQ14-контроллер жестких дисков;
- Int 77h - IRQ15.

Прерывания Int 70h-77h имеют место только в АТ.

Функции ROM BIOS (16-битные сервисы):

- Int 05h (FOOO:FF54h) ~ печать экрана;
- Int 10h — видеосервис;
- Int 11h — чтение списка оборудования (слово из BDA 0040:0010h), возвращает в AX:
 - биты 15:14 — число обнаруженных LPT-портов: 00 — 0, ... 11 — 3;
 - бит 13 — резерв;
 - бит 12 — обнаружен игровой адаптер;
 - биты 11:9 — число обнаруженных COM-портов: 000 — 0, ... 111 — 7;
 - бит 8 — наличие контроллера DMA;
 - биты 7:6 — число обнаруженных НГМД: 00 — 1, ... 11 — 4;
 - биты 5:4 — активный видеорежим: 00 — резерв, 10 — 80-колоночный цветной, 01 — 40-колоночный цветной, 11 — монохромный;
 - биты 3:2 — размер ОЗУ на системной плате (теперь обычно 00);
 - бит 1 — присутствие математического сопроцессора;
 - бит 0 — присутствие дисководов;
- Int 12h — размер непрерывной памяти;
- Int 13h — дисковый сервис (блочный ввод-вывод);

- Int 14h — обслуживание COM-портов;
- Int 15h — АТ-функции (системный сервис, функции определяются значением AH/AX):
 - 00-03h — управление и обмен данными с кассетным магнитофоном (были когда-то и такие «стримеры»!) на старых PC;
 - 4fh — перехват клавиатуры;
 - 53xxh — сервисы управления потреблением АРМ (Advanced Power Management);
 - 8300 h — запуск таймера, устанавливающего флаг в заданной ячейке;
 - 8301h — сброс того же таймера;
 - 84h — джойстик;
 - 86h — программируемая задержка;
 - 87h — перемещение блока расширенной памяти;
 - 88h — получение размера расширенной памяти;
 - 89h — переключение в режим V86;
 - C0h — получение системной конфигурации, при успешном выполнении (CF=0, AH=0) ES:BX указывает на таблицу данных конфигурации;
 - 80-82h, 85h, 90h, 91h — функции многозадачных ОС (BIOS устанавливает заглушки);
- Int 16h — клавиатурный ввод-вывод;
- Int 17h — обслуживание LPT-портов;
- Int 18h — процедура восстановления при неудаче начальной загрузки (прежде - ROM-Basic);
- Int 19h — начальная загрузка (вызов процедуры Bootstrap);
- Int 1Ah — системное время, дата, будильники 16-битные вызовы сервисов PCI;
- Int 1Bh — обработчик нажатия клавиш Ctrl+Break;
- Int 1Ch — User Timer Interrupt, процедура, вызываемая обработчиком Int 08h каждые 55 мс; BIOS устанавливает простую заглушку (IRET), по программы могут перехватывать это прерывание; на время отработки этой процедуры *все аппаратные прерывания запрещены* (кроме NMI).
- Int 33h — поддержка мыши;
- Int 4Ah — обработчик будильника пользователя, установленного функцией BIOS Int 1Ah(6); прерывание вызывается асинхронно, так что при возврате из процедуры все регистры и флаги должны быть в том же состоянии, что и при входе; BIOS ставит заглушку (IRET);
- Int 67h — EMS-функции.

Указатели на таблицы:

- Int 1Dh — видеопараметры;
- Int 1Eh — параметры дискет;
- Int 1Fh — знакогенератор CGA;
- Int 41h — параметры HDD0;
- Int 46h — параметры HDD1;
- Int 43h — знакогенератор EGA.

Таблица А.2. Стандартное назначение ячеек CMOS RTC

Индекс	Назначение
00h-09h, 32h	<i>Ячейки RTC в BCD-формате: (37 в PS/2)</i> 00 - секунды; 01 — секунды будильника; 02 — минуты;

	03 — минуты будильника; 04 — часы; 05 — часы будильника; 06 — день недели; 07 — день месяца; 08 — месяц; 09 — год (2 младшие цифры); 32h — век-1 (2 старшие цифры года); 37h — век-1 (2 старшие цифры года) в PS/2
0Ah	<i>RTC Status Register A (статус-регистр):</i> Бит 7 — обновление времени: 0=готов к чтению. Биты [6:4] — делитель частоты, для 32,768 кГц — 010. Биты [3:0]=0110 — выходная частота меандра 1024 Гц
0Bh	<i>RTC Status Register B (статус-регистр):</i> Бит 7 — остановка часов: 0=нормальный ход. Бит 6 — разрешение периодических прерываний: 0=запрещено. Бит 5 — разрешение прерывания от будильника: 0=запрещено. Бит 4 — разрешение прерывания по окончании смены времени: 0=запрещено. Бит 3 — разрешение выходного меандра (см. регистр 0Ah): 0=запрещено. Бит 2 - BCD/BIN формат: 0=BCD. Бит 1 — 12/24-часовой режим: 1=24-часовой. Бит 0 — зимнее/летнее время: 0=переключение запрещено
0Ch	<i>RTC Status Register C — чтение флагов идентификаторов прерывания:</i> Бит 7 — IRQF — общий запрос прерывания. Бит 6 — PF — периодические прерывания. Бит 5 — AF — прерывание от будильника. Бит 4 — UF — прерывание по смене времени. Биты [3:0] — зарезервированы
0Dh	<i>RTC Status Register D:</i> Бит 7 — питание: 1=норма, 0=разряд батареи. Биты [6:0] — зарезервированы
0Eh	<i>POST Diagnostic Status Byte:</i> Бит 7 — l=Power Lost — терялось питание CMOS. Бит 6 — l=Checksum Bad — ошибка контрольной суммы CMOS. Бит 5 — l=Bad config — ошибка конфигурации. Бит 4 — l=RAM Size Error — несоответствие размера ОЗУ, определенного тестом, записи в CMOS. Бит 3 — l=HDD Error — ошибка при инициализации жесткого диска. Бит 2 — l=Time Valid — нет формальной ошибки часов-календаря (30 февраля, 25 часов). Биты [1:0] — зарезервированы
0Fh	<i>Shutdown Code</i> — используется POST для определения предыстории останова: 00=аппаратный или программный сброс; 01=размер памяти определен; 02=тест памяти прошел; 03=тест памяти выявил ошибку; 04=POST завершен, идет загрузка системы; 05=JMP FAR [0:0467h] с инициализацией контроллера прерываний;

	06=тест защищенного режима прошел; 07=тест защищенного режима выявил ошибку; 08=ошибка при определении размера памяти; 09=перемещен блок Extended Memory (INTlSh); 0A=JMP FAR [0:0467h] без инициализации контроллера прерываний; 0B=используется 80386
10h	<i>Типы НГМД:</i> Биты [7:4] — дисковод A Биты [3:0] — дисковод B 0=нет, 1=360 Кбайт, 2=1,2 Мбайт, 3=720 Кбайт, 4=1,44 Мбайт
11h	<i>Зарезервирован</i>
12h	<i>Типы НЖМД:</i> Биты [7:4] — привод 0; Биты [3:0] — привод 1: 0=нет, 1-Eh=типы 1-14, Ph=тип в байте 19h (для второго привода — в 1Ah)
13h	<i>Зарезервирован</i>
14h	<i>Установленное оборудование</i> Биты [7:6] — количество НГМД: 00=1,01=2. Биты [5:4] — тип первичного видеоадаптера: 00 - EGA или VGA, 01 - CGA, 40 столбцов, 10 - CGA, 80 столбцов, 11 - MDA, 80 столбцов. Биты [3:2] — зарезервированы. Бит 1 — 1=есть математический сопроцессор. Бит 0 - 1=есть НГМД
15h-16h	<i>Размер базовой памяти, Кб (Low/High) 0280h=640K)</i>
17h-18h	<i>Размер •расширенной памяти, Кб (Low/High)</i>
19h, 1Ah	<i>Расширенный тип диска C, D (в PS/2 — зарезервированы)</i>
1Bh-2Dh	<i>Зарезервированы</i>
2Eh-2Fh	<i>Контрольная сумма CMOS с 10h по 20h (High/Low)</i>
30h-31h	<i>Реальный размер расширенной памяти, Кб (Low/High)</i>
32h-33h	<i>В PS/2 контрольная CRC-сумма CMOS с 10h по 31 h (High/Low)</i>
33h	<i>Флаги POST:</i> Бит 7 — наличие 128 Кбайт ОЗУ под границей 1 Мбайт. 1 -есть, доступна теневая память. Бит 6 - SETUP флаг: 1=первая загрузка после выполнения SETUP; обычно установлен 0
34h-3Fh	<i>Зарезервированы (можно писать свою информацию для привязки ПО к машине)</i>
38h-3Fh	<i>В PS/2 пароль, доступ по несуществующим адресам 78h-7Fh</i>

ПРИЛОЖЕНИЕ Б (обязательное). Интерфейс RS-232

ТАБЛИЦА Разъемы и сигналы интерфейса RS-232C.

Обозначение цепи		Контакт разъема		Провод шлейфа выносного разъема PC				Направление	Название цепи
RS-232	Стык 2	DB25S	DB9S	1*	2*	3*	4*	I/O	
PG	101	1	-	(10)	(10)	(10)	1	-	Protect ground- Защитная земля
TD	103	2	3	3	5	3	3	O	Transmit Data- Передаваемые данные
RD	104	3	2	2	3	4	5	I	Receive Data- Принимаемые данные
RTS	105	4	7	7	4	8	7	O	Request To Send- Запрос на передачу
CTS	106	5	8	8	6	7	9	I	Clear To Send- Готовность модема к приему данных для передачи
DSR	107	6	6	6	2	9	11	I	Data Set Ready- Готовность модема к работе
SG	102	7	5	5	9	1	13	-	Signal Ground- Схемная земля
DCD	109	8	1	1	1	5	15	I	Data Carrier Detected- Несущая обнаружена
DTR	108/2	20	4	4	7	2	14	O	Data Terminal Ready- Готовность терминала PC к работе
RI	125	22	9	9	8	6	18	I	Ring Indicator- Индикатор вызова

ТАБЛИЦА Назначение сигналов интерфейса RS-232C.

Сигнал	Назначение
PG	Защитная земля, соединяется с корпусом устройства и экраном кабеля.
SG	Сигнальная (схемная земля), относительно которой действуют уровни сигналов.
TD	Последовательные данные – выход передатчика.
RD	Последовательные данные – вход приемника.
RTS	Выход запроса передачи данных: состояние «включено» уведомляет модем о наличии у терминала данных для передачи. В полудуплексном режиме используется для управления направлением – состояние «включено» является сигналом модему на переключение в режим передачи.
CTS	Вход разрешения терминалу передавать данные. Состояние «выключено» аппаратно запрещает передачу данных. Сигнал используется для аппаратного управления потоком данных.
DTR	Выход сигнала готовности терминала к обмену данными. Состояние «включено» поддерживает коммутируемый канал в состоянии соединения.
DSR	Вход сигнал готовности от аппаратуры передачи данных.
DCD	Вход сигнала обнаружения несущей удаленного модема.
RI	Вход индикатора вызова. В коммутируемом канале этим сигналом модем сигнализирует о принятии вызова.

ПРИЛОЖЕНИЕ В (обязательное). Стандартные прерывания MS-DOS и ROM BIOS

Определение положение курсора мыши

Функция 03h возвращает текущие координаты курсора мыши и состояние клавиш.

Регистры на входе: AX = 0003h
Регистры на выходе: BX = состояние клавиш мыши;
CX = координата X курсора;
DX = координата Y курсора

В зависимости от того, была ли нажата какая-либо клавиша мыши в момент вызова функции, в регистре BX могут быть установлены следующие флаги:

<i>Установленный бит регистра BX</i>	<i>Клавиша, которая была нажата</i>
0	Левая
1	Правая
2	Средняя

Для графических режимов координаты располагаются в различных диапазонах, в зависимости от текущего режима видеоадаптера:

<i>Размер экрана в пикселах</i>	<i>Номер режима</i>	<i>Диапазон координат по оси X</i>	<i>Диапазон координат по оси Y</i>
320x200	4, 5	0 - 638	0 - 199
640x200	6	0 - 639	0 - 199
320x200	0Dh	0 - 638	0 - 199
640x200	0Eh	0 - 639	0 - 199
640x350	0Fh	0 - 639	0 - 349

Программы, работающие в текстовом режиме, должны разделить полученные координаты на 8 (как координату X, так и координату Y).

Функции дискового сервиса.

Номер функции (AH)	Назначение
00h	Reset Disk System – сброс контроллера
01h	Read Status of Last Operation – чтение состояния последней операции
02h	Read Desired Sector into Memory – чтение секторов с диска в память

03h	Write Desired Sector into Memory – запись секторов из памяти на диск
04h	Verify Desired Sector - верификация секторов
05h	Format Desired Track – форматирование трека ³
08h	Read Drive Parameters – получение параметров диска
09h¹	Initialize Drive Parameter Characteristics – инициализация таблиц параметров диска ³
0Ah¹	Read Long – «длинное» чтение ³
0Bh¹	Write Long – «длинная» запись ³
0Ch¹	Seek –поиск цилиндра
0Dh¹	Alternative Disk Rest – альтернативный сброс
10h¹	Test Drive Ready – проверка готовности
11h¹	Recalibrate –рекалибровка
14h¹	Controller Internal Diagnostics – диагностика контроллера диска
15h	Read DEASD Type – получение типа диска
16h²	Diskette Change Line Status – проверка статуса смены диска
17h²	Set Diskette Type for Format – установка типа диска (для форматирования)
18h²	Set MediaType for Format - установка типа носителя (для форматирования)
20h²	Set MediaType – получение типа установленного носителя
24h²	Set Multiple Mode – установка параметров режима многосекторного обращения
25h²	Identify Driver ATA – идентификация накопителя (только для ATA-дисков)
41h-44h	Резерв для Extended Disk-Access Support – поддержки расширенного доступа к диску
45h-46h	Резерв для Removeable-Media Control Support – поддержки управления сменой носителя
47h-48h	Резерв для Extended Disk-Access Support - поддержки расширенного доступа к диску
49h	Резерв для Removeable-Media Control Support - поддержки управления сменой носителя

ПРИЛОЖЕНИЕ Г (обязательное).
Скан-коды клавиатуры

Клавиша	Set#1	Set#2	Set#3	Клавиша	Set#1	Set#2	Set#3
1!	02	16	16	A	1E	1C	1C
2@	03	1E	1E	S	1F	1B	1B
3#	04	26	26	D	20	23	23
4\$	05	25	25	F	21	2B	2B
5%	06	2E	2E	G	22	34	34
6^	07	36	36	H	23	33	33
7&	08	3D	3D	J	24	3B	3B
8*	09	3E	3E	K	25	42	42
9(	0A	46	46	L	26	4B	4B
0)	0B	45	45	; :	27	4C	4C
- _	0C	4E	4E	, «	28	52	52
= +	0D	55	55	' ~	29	OE	OE
Backspace	0E	66	66	Shift(L)	2A	12	12
Tab	0F	0D	0D	\	2B	61	13
Q	10	15	15	Z	2C	1A	1A
W	11	1D	1D	X	2D	22	22
E	12	24	24	C	2E	21	21
R	13	2D	2D	V	2F	2A	2A
T	14	2C	2C	B	30	32	32
Y	15	35	35	N	31	31	31
U	16	3C	3C	M	32	3A	3A
I	17	43	43	,<	33	41	41
O	18	44	44	.>	34	49	49
P	19	4D	4D	/?	35	4A	4A
[{	1A	54	54	Shift (R)	36	59	59
]}	1B	5B	5B	Alt (L)	38	11	19
Enter	1C	5A	5A	(Space bar)	39	29	29
Ctrl(L)	1D	14	11	Caps Lock	3A	58	14

Клавиша	Set#1	Set#2	Set#3	Примечание
Esc	01	76	08	-
F1	3B	05	07	-
F2	3C	06	0F	-
F3	3D	04	17	-
F4	3E	0C	1F	-
F5	3F	03	27	-
F6	40	0B	2F	-
F7	41	83	37	-
F8	42	0A	3F	-
F9	43	01	47	-
F10	44	09	4F	-
F11	57	78	56	101/102
F12	58	07	5E	101/102

Scroll Lock	46	7E	5F	-
Sys Rq	54	-	-	AT-84*
*Prt Sc	37	-	-	XT,AT-84
*(Keypad)	37	7C	7E	101/102
Num Lock	45	77	76	-
Home[7]	47	6C	6C	-
↑[8]	48	75	75	-
PgUp[9]	49	7D	7D	-
-(Keypad)	4A	7B	84	-
←[4]	4B	6B	6B	-
[5]	4C	73	73	-
→[6]	4D	74	74	-
+(Keypad)	4E	E0-5A	7C	-
End[1]	4F	69	69	-
↓[2]	50	72	72	-
PgDn[3]	51	7A	72	-
Ins[0]	52	70	70	-
Del[.]	53	71	71	-

Клавиша	Set#1	Set#2	Set#3
Alt (R)	E0-38	E0-11	39
Ctrl (R)	E0-1D	E0-14	58
Print Screen	E0-2A-E0-37	E0-12-E0-7C	57
Shift +Print Screen(SysReq)	E0-37	-	-
Ctrl +Print Screen(SysReq)	E0-37	-	-
Alt + Print Screen	54	-	-
Pause	E1-1D-45-E1-9D-C5	E1-14-77-E1-F0-14-F0-77	62
Ctrl + Pause	E0-46-E0-C6	-	-
Insert	E0-52	E0-70	67
Shift-Insert	E0-AA-E0	-	52
Delete	E0-53	E0-71	64
Shift- Delete	E0-AA-E0-4B	-	-
←	E0-4B	E0-6B	61
Shift + ←	E0-AA-E0-4B	-	-
Home	E0-47	E0-6C	6E
Shift + Home	E0-AA-E0-47	-	-
End	E0-4F	E0-69	65
Shift + End	E0-AA-E0-4f	-	-
↑	E0-48	E0-75	63
Shift + ↑	E0-AA-E0-48	-	-
↓	E0-50	E0-72	60
Shift + ↓	E0-AA-E0-50	-	-
Page Up	E0-49	E0-7D	6F
Shift + Page Up	E0-AA-E0-49	-	-
Page Down	E0-51	E0-7A	6D
Shift + Page Down	E0-AA-E0-51	-	-

→	E0-4d	E0-74	6A
Shift + →	E0-AA-E0-4d	-	-
Key Enter	E0-1C	E0-5A	79
Keypad/	E0-35	E0-4A	77
Shift + Keypad/	E0-AA- E0-35	-	-

ПРИЛОЖЕНИЕ Д (обязательное).
Видорежимы

Функция	Название
<i>Стандартные (базовые) функции</i>	
00h	Установка видеорежима
01h	Установка размера и формы курсора
02h	Установка позиции курсора
03h	Чтение позиции курсора
04h	Чтение светового пера
05h	Выбор активной видеостраницы
06h	Прокрутка (очистка) в окне вверх
07h	Прокрутка в окне вниз
08h	Чтение символа и атрибута
09h	Запись символа и атрибута
0Ah	Запись символа
0Bh	Выбор палитры и бордюра
0Ch	Запись графического пиксела
0Dh	Чтение графического пиксела
0Eh	Вывод в телетайпном режиме
0Fh	Чтение видеорежима
<i>Расширенные функции для адаптера EGA и VGA</i>	
10h	Установка палитры(EGA, VGA)
11h	Знакогенератор (EGA, VGA)
12h	Специальные функции (EGA, VGA)
13h	Вывод строки (EGA, VGA)
14h-15h	Функции LCD- дисплея (PC Convertable)
1Ah	Чтение/ установка кодов активного и дополнительного дисплеев (VGA)
1Bh	Чтение блока данных состояния и возможностей видеосистемы(VGA)
1Ch	Сохранение и восстановление состояния видеосистемы(VGA)
Графические функции SVGA VBE (VESA BIOS Extensions)	
4F00h	Проверка наличия поддержки VBE
4F01h	Информация о режимах VBE
4F02h	Установка видеорежимов VBE
4F03h	Информация о текущем видеорежиме VBE
4F04h	Сохранение и восстановление состояния видеосистемы SVGA
4F05h	Определение окна видеопамати SVGA
4F06h	Установка/чтение длины строки
4F07h	Установка/чтение начала экрана
4F08h	Установка/чтение формата палитр (DAC)
4F09h	Установка/чтение регистра палитр (RAMDAC)
4F0Ah	Интерфейс VBE защищенного режима
<i>Функции дополнительной поддержки VBE</i>	
4F10h	Управление энергопотреблением
4F11h	Интерфейс плоских экранных панелей

4F12h	Расширение интерфейса курсора
4F13h	Расширение аудиоинтерфейса
4F14h	Расширение по усмотрению производителя
4F15h	Канал связи с монитором (DDC)
4F16h	Конфигурация графической системы

Видеорежим		Формат	Кол-во цветов
100h	Gr	640x400	256
101h*	Gr	640x480	256
102h**	Gr	800x600	16
103h	Gr	800x600	256
104h	Gr	1024x768	16
105h	Gr	1024x768	256
108h	TXT	80x60	16
109h	TXT	132x25	16
10Ah	TXT	132x43	16
10Bh	TXT	132x50	16
10Ch	TXT	132x60	16
10Dh	Gr	320x200	32768
10Eh	Gr	320x200	65536
10Fh	Gr	320x200	16,7M
110h	Gr	640x480	32768
111h	Gr	640x480	65536
112h	Gr	640x480	16,7M
113h	Gr	800x600	32768
114h	Gr	800x600	65536
116h	Gr	1024x768	32768
117h	Gr	1024x768	65536
118h	Gr	1024x768	16,7M
119h	Gr	1280x1024	32768
11Ah	Gr	1280x1024	65536
11Bh	Gr	1280x1024	16,7M
11Ch	Gr	1600x1200	32768
11Dh	Gr	1600x1200	65536
11Eh	Gr	1600x1200	16,7M

ПРИЛОЖЕНИЕ Е (обязательное).

Классы WMI

Класс Win32_PnPEntity

Класс Win32_PnPEntity представляет все установленные устройства Plug-and-Play. Пример получения информации:

On Error Resume Next

Set objService = GetObject("winmgmts:{impersonationLevel=impersonate}!\\.\root\CIMV2")

If Err.Number <> 0 Then

WScript.Echo Err.Number & ": " & Err.Description

WScript.Quit

End If

For Each objPnP In objService.ExecQuery("SELECT * FROM Win32_PnPEntity")

WScript.Echo objPnP.Name 'наименование устройства

WScript.Echo objPnP.Description 'описание устройства

WScript.Echo objPnP.Manufacturer 'производитель

WScript.Echo objPnP.PNPDeviceID 'идентификатор логического устройства

WScript.Echo

Next

Класс Win32_SystemEnclosure

Класс Win32_SystemEnclosure представляет свойства, связанные с физическим системным корпусом компьютера. Пример получения информации:

On Error Resume Next

Set objService = GetObject("winmgmts:{impersonationLevel=impersonate}!\\.\root\CIMV2")

If Err.Number <> 0 Then

WScript.Echo Err.Number & ": " & Err.Description

WScript.Quit

End If

For Each objObject In objService.ExecQuery("SELECT * FROM Win32_SystemEnclosure")

WScript.Echo objObject.Caption 'наименование устройства

WScript.Echo objObject.Description 'описание

WScript.Echo Join(objObject.ChassisTypes) 'тип корпуса (7 - Tower, 10 - Notebook и

т.д.)

Next

Класс Win32_BaseBoard

Класс Win32_BaseBoard предоставляет сведения о материнской плате. Пример получения информации:

On Error Resume Next

Set objService = GetObject("winmgmts:{impersonationLevel=impersonate}!\\.\root\CIMV2")

If Err.Number <> 0 Then

WScript.Echo Err.Number & ": " & Err.Description

WScript.Quit

End If

For Each objObject In objService.ExecQuery("SELECT * FROM Win32_BaseBoard")

WScript.Echo objObject.Caption 'наименование устройства

WScript.Echo objObject.Description 'описание

WScript.Echo objObject.Manufacturer 'производитель

WScript.Echo objObject.Product 'тип, определённый производителем

WScript.Echo objObject.SerialNumber 'серийный номер производителя

WScript.Echo objObject.Tag 'идентификатор в системе

WScript.Echo objObject.Version 'версия

Next

Класс Win32_MotherboardDevice

Класс Win32_MotherboardDevice предоставляет сведения о материнской плате. Пример получения информации:

```
On Error Resume Next
```

```
Set objService = GetObject("winmgmts:{impersonationLevel=impersonate}!\\.\root\CIMV2")
```

```
If Err.Number <> 0 Then
```

```
    WScript.Echo Err.Number & ": " & Err.Description
```

```
    WScript.Quit
```

```
End If
```

```
For Each objMoth In objService.ExecQuery("SELECT * FROM Win32_MotherboardDevice")
```

```
    WScript.Echo objMoth.SystemName 'имя компьютера
```

```
    WScript.Echo objMoth.Caption 'наименование устройства
```

```
    WScript.Echo objMoth.Description 'описание устройства
```

```
    WScript.Echo objMoth.DeviceID 'идентификатор устройства
```

```
    WScript.Echo objMoth.PrimaryBusType 'тип первичной шины системной платы
```

```
    WScript.Echo objMoth.SecondaryBusType 'тип вторичной шины системной платы
```

```
Next
```

Класс Win32_OnBoardDevice

Класс Win32_OnBoardDevice представляет общие адаптеры, встроенные в системную плату.

Пример получения информации:

```
On Error Resume Next
```

```
Set objService = GetObject("winmgmts:{impersonationLevel=impersonate}!\\.\root\CIMV2")
```

```
If Err.Number <> 0 Then
```

```
    WScript.Echo Err.Number & ": " & Err.Description
```

```
    WScript.Quit
```

```
End If
```

```
For Each objObject In objService.ExecQuery("SELECT * FROM Win32_OnBoardDevice")
```

```
    WScript.Echo objObject.Caption 'наименование устройства
```

```
    WScript.Echo objObject.Description 'описание
```

```
    WScript.Echo objObject.DeviceType 'тип устройства
```

```
    '(3 - Video, 4 - SCSI Controller, 5 - Ethernet, 6 - Token Ring, 7 - Sound и т.д.)
```

```
    WScript.Echo objObject.PartNumber 'серийный номер производителя
```

```
    WScript.Echo objObject.SerialNumber 'серийный номер производителя
```

```
Next
```

Класс Win32_Bus

Класс Win32_Bus представляет физические шины. Пример получения информации:

```
On Error Resume Next
```

```
Set objService = GetObject("winmgmts:{impersonationLevel=impersonate}!\\.\root\CIMV2")
```

```
If Err.Number <> 0 Then
```

```
    WScript.Echo Err.Number & ": " & Err.Description
```

```
    WScript.Quit
```

```
End If
```

```
For Each objObject In objService.ExecQuery("SELECT * FROM Win32_Bus")
```

```
    WScript.Echo objObject.Caption 'наименование устройства
```

```
    WScript.Echo objObject.Description 'описание
```

```
    WScript.Echo objObject.BusType 'тип шины (1 - ISA, 5 - PCI Bus, 15 - PNP Bus и т.д.)
```

```
    WScript.Echo objObject.DeviceID 'идентификатор устройства
```

```
    WScript.Echo objObject.SystemName 'имя компьютера
```

```
    WScript.Echo
```

```
Next
```

Класс Win32_SystemSlot

Класс Win32_SystemSlot предоставляет информацию о слотах системной платы. Пример получения информации:

```
On Error Resume Next
```

```
Set objService = GetObject("winmgmts:{impersonationLevel=impersonate}!\\.\root\CIMV2")
```

```
If Err.Number <> 0 Then
```

```
    WScript.Echo Err.Number & ": " & Err.Description
```

```
    WScript.Quit
```

```
End If
```

```
For Each objObject In objService.ExecQuery("SELECT * FROM Win32_SystemSlot")
```

```
    WScript.Echo objObject.Caption 'наименование устройства
```

```
    WScript.Echo objObject.Description 'описание
```

```
    WScript.Echo Join(objObject.ConnectorType) 'массив типов слота, напр., 43 - PCI
```

```
    WScript.Echo objObject.CurrentUsage 'используется: 3 - Available, 4 - In use
```

```
    WScript.Echo objObject.Number 'номер слота
```

```
    WScript.Echo objObject.SlotDesignation 'идентификатор слота
```

```
    WScript.Echo objObject.Tag 'идентификатор слота
```

```
    WScript.Echo
```

```
Next
```

Класс Win32_Processor

Каждому процессору в системе соответствует экземпляр класса Win32_Processor в WMI. На многопроцессорной машине таких экземпляров будет несколько. Пример получения информации:

```
On Error Resume Next
```

```
Set objService = GetObject("winmgmts:{impersonationLevel=impersonate}!\\.\root\CIMV2")
```

```
If Err.Number <> 0 Then
```

```
    WScript.Echo Err.Number & ": " & Err.Description
```

```
    WScript.Quit
```

```
End If
```

```
For Each objProc In objService.ExecQuery("SELECT * FROM Win32_Processor")
```

```
    WScript.Echo objProc.Caption 'краткое описание процессора
```

```
    WScript.Echo objProc.Description 'краткое описание процессора
```

```
    WScript.Echo Trim(objProc.Name) 'наименование процессора
```

```
    'номер версии процессора, зависящий от архитектуры (не исп. в Windows 95):
```

```
    WScript.Echo objProc.Version
```

```
    WScript.Echo objProc.Manufacturer 'производитель
```

```
    WScript.Echo objProc.DeviceID 'уникальный идентификатор процессора в системе
```

```
    'статус:
```

```
    '0 - Unknown
```

```
    '1 - CPU Enabled
```

```
    '2 - CPU Disabled by User via BIOS Setup
```

```
    '3 - CPU Disabled By BIOS (POST Error)
```

```
    '4 - CPU is Idle
```

```
    '5 - Reserved
```

```
    '6 - Reserved
```

```
    '7 - Other
```

```
    WScript.Echo objProc.CpuStatus 'статус
```

```
    WScript.Echo objProc.LoadPercentage 'загрузка процессора в прошлую секунду (в проц.)
```

```
    WScript.Echo objProc.CurrentClockSpeed 'текущая скорость процессора - в MHz
```

```

WScript.Echo objProc.MaxClockSpeed 'максимальная скорость процессора - в MHz
WScript.Echo objProc.ExtClock 'частота внешн. таймера (external clock frequency) в
MHz
WScript.Echo objProc.L2CacheSize 'размер кэша процессора Level 2
WScript.Echo objProc.L2CacheSpeed 'скорость кэша процессора Level 2 - в MHz
'тип процессора:
'1 - Other
'2 - Unknown
'3 - Central Processor
'4 - Math Processor
'5 - DSP Processor
'6 - Video Processor
WScript.Echo objProc.ProcessorType 'тип процессора
WScript.Echo objProc.SocketDesignation 'тип chip socket'a
WScript.Echo objProc.SystemName 'имя компьютера
Next
Класс Win32_CacheMemory

```

Класс Win32_CacheMemory представляет внутреннюю и внешнюю кэш-память в компьютерной системе. Пример получения информации:

```

On Error Resume Next
Set objService = GetObject("winmgmts:{impersonationLevel=impersonate}!\\.\root\CIMV2")
If Err.Number <> 0 Then
    WScript.Echo Err.Number & ": " & Err.Description
    WScript.Quit
End If
For Each objObject In objService.ExecQuery("SELECT * FROM Win32_CacheMemory")
    WScript.Echo objObject.SystemName 'имя компьютера
    WScript.Echo objObject.Caption 'наименование устройства
    WScript.Echo objObject.Description 'описание
    WScript.Echo objObject.DeviceID 'идентификатор устройства
    WScript.Echo objObject.BlockSize 'размер блоков в байтах; 1 - неизвестно
    WScript.Echo objObject.CacheType 'тип кэша (напр., 3 - Instruction, 4 - Data)
    'массив типов статической оперативной памяти (SRAM) используемых для кэш-
    памяти,
    'напр., 2 - Non-Burst, 3 - Burst, 4 - Pipeline Burst, 5 - Synchronous, 6 - Asynchronous
    WScript.Echo Join(objObject.CurrentSRAM)
    WScript.Echo objObject.Level 'уровень кэша (напр., 3 - Primary, 4 - Secondary, 5 -
    Tertiary)
    'физическое местоположение кэш-памяти, 0 - Internal, 1 - External:
    WScript.Echo objObject.Location
    WScript.Echo objObject.MaxCacheSize 'максимальный размер кэша
    WScript.Echo objObject.Purpose 'назначение
    WScript.Echo
Next
Класс Win32_PhysicalMemory

```

Экземпляр класса Win32_PhysicalMemory представляет физическое устройство памяти, расположенное на компьютере и доступное операционной системе. Пример получения информации:

```

On Error Resume Next
Set objService = GetObject("winmgmts:{impersonationLevel=impersonate}!\\.\root\CIMV2")
If Err.Number <> 0 Then
    WScript.Echo Err.Number & ": " & Err.Description
    WScript.Quit
End If
For Each objPhMem In objService.ExecQuery("SELECT * FROM Win32_PhysicalMemory")
    WScript.Echo objPhMem.Name 'наименование устройства
    WScript.Echo objPhMem.Capacity 'размер физической памяти в байтах
    WScript.Echo objPhMem.DeviceLocator 'гнездо или монтажная схема, к которой под-
ключена память
Next
Класс Win32_MemoryDevice

```

Класс Win32_MemoryDevice предоставляет информацию о начальных и конечных адресах для всех устройств памяти, установленных на компьютере. Пример получения информации:

```

On Error Resume Next
Set objService = GetObject("winmgmts:{impersonationLevel=impersonate}!\\.\root\CIMV2")
If Err.Number <> 0 Then
    WScript.Echo Err.Number & ": " & Err.Description
    WScript.Quit
End If
For Each objObject In objService.ExecQuery("SELECT * FROM Win32_MemoryDevice")
    WScript.Echo objObject.SystemName 'имя компьютера
    WScript.Echo objObject.Caption 'наименование устройства
    WScript.Echo objObject.Description 'описание
    WScript.Echo objObject.DeviceID 'идентификатор устройства
    WScript.Echo objObject.StartingAddress 'начальный адрес
    WScript.Echo objObject.EndingAddress 'конечный адрес
    WScript.Echo
Next
Класс Win32_MemoryArray

```

Класс Win32_MemoryArray предоставляет информацию о начальных и конечных адресах для всех массивов памяти, установленных на компьютере. Пример получения информации:

```

On Error Resume Next
Set objService = GetObject("winmgmts:{impersonationLevel=impersonate}!\\.\root\CIMV2")
If Err.Number <> 0 Then
    WScript.Echo Err.Number & ": " & Err.Description
    WScript.Quit
End If
For Each objObject In objService.ExecQuery("SELECT * FROM Win32_MemoryArray")
    WScript.Echo objObject.SystemName 'имя компьютера
    WScript.Echo objObject.Caption 'наименование устройства
    WScript.Echo objObject.Description 'описание
    WScript.Echo objObject.DeviceID 'идентификатор устройства
    WScript.Echo objObject.StartingAddress 'начальный адрес
    WScript.Echo objObject.EndingAddress 'конечный адрес
    WScript.Echo
Next

```


Класс Win32_PhysicalMemoryArray

Класс Win32_PhysicalMemoryArray предоставляет информацию о массиве физической памяти компьютера. Пример получения информации:

```
On Error Resume Next
Set objService = GetObject("winmgmts:{impersonationLevel=impersonate}!\.\root\CIMV2")
If Err.Number <> 0 Then
    WScript.Echo Err.Number & ": " & Err.Description
    WScript.Quit
End If
For Each objObject In objService.ExecQuery("SELECT * FROM Win32_PhysicalMemoryArray")
    WScript.Echo objObject.Caption 'наименование устройства
    WScript.Echo objObject.Description 'описание
    WScript.Echo objObject.MaxCapacity 'максимальный размер памяти в байтах
    WScript.Echo objObject.MemoryDevices 'количество физических слотов или гнезд,
доступных в этом массиве памяти
    WScript.Echo objObject.Tag 'идентификатор массива памяти
    WScript.Echo
Next
Класс Win32_DeviceMemoryAddress
```

Класс Win32_DeviceMemoryAddress предоставляет информацию о начальных и конечных адресах памяти для всех устройств, установленных на компьютере. Пример получения информации:

```
On Error Resume Next
Set objService = GetObject("winmgmts:{impersonationLevel=impersonate}!\.\root\CIMV2")
If Err.Number <> 0 Then
    WScript.Echo Err.Number & ": " & Err.Description
    WScript.Quit
End If
For Each objObject In objService.ExecQuery("SELECT * FROM Win32_DeviceMemoryAddress")
    WScript.Echo objObject.CSName 'имя компьютера
    WScript.Echo objObject.Caption 'краткое описание
    WScript.Echo objObject.Description 'описание
    WScript.Echo objObject.StartingAddress 'начальный адрес
    WScript.Echo objObject.EndingAddress 'конечный адрес
    WScript.Echo
Next
Класс Win32_DMACHannel
```

Класс Win32_DMACHannel предоставляет информацию о каналах DMA (прямого доступа к памяти) на компьютере. Прямой доступ к памяти является методом переместить данные от устройства в память (или наоборот) без помощи микропроцессора. Системная плата использует контроллер DMA, чтобы получить установленное количество каналов, каждый из которых может использоваться только одним устройством одновременно. Пример получения информации:

```
On Error Resume Next
Set objService = GetObject("winmgmts:{impersonationLevel=impersonate}!\.\root\CIMV2")
```

```

If Err.Number <> 0 Then
    WScript.Echo Err.Number & ": " & Err.Description
    WScript.Quit
End If
For Each objObject In objService.ExecQuery("SELECT * FROM Win32_DMACHannel")
    WScript.Echo objObject.CSName 'имя компьютера
    WScript.Echo objObject.Caption 'наименование устройства
    WScript.Echo objObject.Description 'описание
    WScript.Echo objObject.DMAChannel 'номер канала DMA
    WScript.Echo
Next
Класс Win32_PortConnector

```

Класс Win32_PortConnector предоставляет информацию о физических портах подключения, таких, как Centronics, PS/2 и т.д. Пример получения информации:

```

On Error Resume Next
Set objService = GetObject("winmgmts:{impersonationLevel=impersonate}!\\.\root\CIMV2")
If Err.Number <> 0 Then
    WScript.Echo Err.Number & ": " & Err.Description
    WScript.Quit
End If
For Each objObject In objService.ExecQuery("SELECT * FROM Win32_PortConnector")
    WScript.Echo objObject.Caption 'наименование устройства
    WScript.Echo objObject.Description 'описание
    'массив атрибутов порта; некоторые возможные значения:
    '16 - ATA 3-1/2 Inch (40 pins)
    '53 - USB
    '61 - PS/2
    '64 - Access.bus
    '88 - SSA SCSI
    WScript.Echo Join(objObject.ConnectorType)
    WScript.Echo objObject.ExternalReferenceDesignator 'идентификатор, определяющий
тип и использование порта
    'тип порта; некоторые возможные значения:
    '5 - Parallel Port ECP/EPP
    '8 - Serial Port 16550 Compatible
    '11 - MIDI Port
    '12 - Joystick Port
    '13 - Keyboard Port
    '14 - Mouse Port
    '16 - USB
    '28 - Video Port
    '29 - Audio Port
    WScript.Echo objObject.PortType
    WScript.Echo objObject.Tag 'идентификатор порта
    WScript.Echo
Next
Класс Win32_PortResource

```

Класс Win32_PortResource предоставляет информацию обо всех портах ввода - вывода (I/O ports), найденных на компьютере. Пример получения информации:

```

On Error Resume Next
Set objService = GetObject("winmgmts:{impersonationLevel=impersonate}!\\.\root\CIMV2")
If Err.Number <> 0 Then
    WScript.Echo Err.Number & ": " & Err.Description
    WScript.Quit
End If
For Each objObject In objService.ExecQuery("SELECT * FROM Win32_PortResource")
    WScript.Echo objObject.Caption 'наименование устройства
    WScript.Echo objObject.Description 'описание
    WScript.Echo objObject.CSName 'имя компьютера
    WScript.Echo objObject.StartingAddress 'начальный адрес
    WScript.Echo objObject.EndingAddress 'конечный адрес
    WScript.Echo
Next
Класс Win32_ParallelPort

```

Класс Win32_ParallelPort содержит сведения о параллельных портах. Пример получения информации:

```

On Error Resume Next
Set objService = GetObject("winmgmts:{impersonationLevel=impersonate}!\\.\root\CIMV2")
If Err.Number <> 0 Then
    WScript.Echo Err.Number & ": " & Err.Description
    WScript.Quit
End If
For Each objPort In objService.ExecQuery("SELECT * FROM Win32_ParallelPort")
    WScript.Echo objPort.Caption 'наименование устройства
    WScript.Echo objPort.Description 'описание устройства
    WScript.Echo objPort.DeviceID 'идентификатор устройства
    WScript.Echo objPort.PNPDeviceID 'идентификатор устройства Plug-and-Play
    WScript.Echo objPort.SystemName 'имя компьютера
Next
Классы Win32_SerialPort и Win32_SerialPortConfiguration

```

Классы Win32_SerialPort и Win32_SerialPortConfiguration содержат сведения о последовательных портах. Пример получения информации:

```

On Error Resume Next
Set objService = GetObject("winmgmts:{impersonationLevel=impersonate}!\\.\root\CIMV2")
If Err.Number <> 0 Then
    WScript.Echo Err.Number & ": " & Err.Description
    WScript.Quit
End If
For Each objPort In objService.ExecQuery("SELECT * FROM Win32_SerialPort")
    WScript.Echo objPort.Caption 'наименование устройства
    WScript.Echo objPort.Description 'описание устройства
    WScript.Echo objPort.DeviceID 'идентификатор устройства
    WScript.Echo objPort.PNPDeviceID 'идентификатор устройства Plug-and-Play
    WScript.Echo objPort.SystemName 'имя компьютера
    WScript.Echo
Next

```

Класс Win32_Keyboard

Класс Win32_Keyboard содержит сведения о клавиатуре. Пример получения информации:

```
On Error Resume Next
```

```
Set objService = GetObject("winmgmts:{impersonationLevel=impersonate}!\\.\root\CIMV2")
```

```
If Err.Number <> 0 Then
```

```
    WScript.Echo Err.Number & ": " & Err.Description
```

```
    WScript.Quit
```

```
End If
```

```
For Each objKeyb In objService.ExecQuery("SELECT * FROM Win32_Keyboard")
```

```
    Exit For
```

```
Next
```

```
WScript.Echo objKeyb.Caption 'наименование устройства
```

```
WScript.Echo objKeyb.Description 'описание устройства
```

```
WScript.Echo objKeyb.DeviceID 'адрес или другая уникальная информация идентификации
```

```
WScript.Echo objKeyb.NumberOfFunctionKeys 'количество функциональных клавиш
```

```
WScript.Echo objKeyb.SystemName 'имя компьютера
```

Класс Win32_PointingDevice

Класс Win32_PointingDevice содержит сведения о мыши. Пример получения информации:

```
On Error Resume Next
```

```
Set objService = GetObject("winmgmts:{impersonationLevel=impersonate}!\\.\root\CIMV2")
```

```
If Err.Number <> 0 Then
```

```
    WScript.Echo Err.Number & ": " & Err.Description
```

```
    WScript.Quit
```

```
End If
```

```
For Each objMouse In objService.ExecQuery("SELECT * FROM Win32_PointingDevice")
```

```
    Exit For
```

```
Next
```

```
WScript.Echo objMouse.Caption 'наименование устройства
```

```
WScript.Echo objMouse.Description 'описание устройства
```

```
WScript.Echo objMouse.HardwareType 'тип устройства
```

```
WScript.Echo objMouse.Manufacturer 'производитель
```

```
WScript.Echo objMouse.DeviceID 'уникальный идентификатор устройства
```

```
WScript.Echo objMouse.DeviceInterface 'интерфейс: 162 - USB, 4 - PS/2, 3 - Serial и т.д.
```

```
WScript.Echo objMouse.Handedness 'настройка: 2 - мышь для правши, 3 - мышь для левши
```

```
WScript.Echo objMouse.NumberOfButtons 'количество кнопок
```

```
WScript.Echo objMouse.SystemName 'имя компьютера
```

Класс Win32_SoundDevice

Класс Win32_SoundDevice содержит сведения о звуковой карте. Пример получения информации:

```
On Error Resume Next
```

```
Set objService = GetObject("winmgmts:{impersonationLevel=impersonate}!\\.\root\CIMV2")
```

```
If Err.Number <> 0 Then
```

```
    WScript.Echo Err.Number & ": " & Err.Description
```

```
    WScript.Quit
```

```
End If
```

```
For Each objSound In objService.ExecQuery("SELECT * FROM Win32_SoundDevice")
```

```
    WScript.Echo objSound.Caption 'наименование устройства
```

```
    WScript.Echo objSound.ProductName 'наименование устройства
```

```
    WScript.Echo objSound.Description 'описание устройства
```

```
    WScript.Echo objSound.Manufacturer 'производитель
```

```
WScript.Echo objSound.DeviceID 'идентификатор устройства
WScript.Echo objSound.SystemName 'имя компьютера
```

```
Next
```

```
Класс Win32_VideoController
```

Класс Win32_VideoController содержит сведения о видеокарте. Пример получения информации:

```
On Error Resume Next
```

```
Set objService = GetObject("winmgmts:{impersonationLevel=impersonate}!\\.\root\CIMV2")
```

```
If Err.Number <> 0 Then
```

```
    WScript.Echo Err.Number & ": " & Err.Description
```

```
    WScript.Quit
```

```
End If
```

```
For Each objVideo In objService.ExecQuery("SELECT * FROM Win32_VideoController")
```

```
    WScript.Echo objVideo.SystemName 'имя компьютера
```

```
    WScript.Echo objVideo.Caption 'наименование устройства
```

```
    WScript.Echo objVideo.Description 'описание устройства
```

```
    WScript.Echo objVideo.DeviceID 'идентификатор устройства
```

```
    WScript.Echo objVideo.AdapterRAM 'размер памяти видеоадаптера
```

```
    WScript.Echo objVideo.CurrentHorizontalResolution 'текущее разрешение экрана по го-
ризонтالي (точек)
```

```
    WScript.Echo objVideo.CurrentVerticalResolution 'текущее разрешение экрана по вер-
тикали (точек)
```

```
    WScript.Echo objVideo.CurrentNumberOfColors 'число цветов в текущем разрешении
экрана
```

```
    WScript.Echo objVideo.VideoModeDescription 'текущее разрешение экрана и число
цветов
```

```
    WScript.Echo objVideo.CurrentBitsPerPixel 'качество цветопередачи (количество бит на
пиксель)
```

```
    WScript.Echo objVideo.CurrentRefreshRate 'частота обновления экрана, Гц
```

```
    '(0 - по умолчанию, 0xFFFFFFFF - оптимальная)
```

```
    WScript.Echo objVideo.DriverDate 'дата-время последней модификации текущего ви-
деодрайвера
```

```
    WScript.Echo objVideo.DriverVersion 'версия текущего видеодрайвера
```

```
    WScript.Echo objVideo.PNPDeviceID 'идентификатор устройства Plug-and-Play
```

```
    WScript.Echo objVideo.VideoProcessor 'описание видеопроцессора
```

```
Next
```

```
Класс CIM_VideoControllerResolution
```

Класс CIM_VideoControllerResolution представляет различные видео режимы, которые под-держивает видео контроллер. Пример получения информации:

```
On Error Resume Next
```

```
Set objService = GetObject("winmgmts:{impersonationLevel=impersonate}!\\.\root\CIMV2")
```

```
If Err.Number <> 0 Then
```

```
    WScript.Echo Err.Number & ": " & Err.Description
```

```
    WScript.Quit
```

```
End If
```

```
For Each objObject In objService.ExecQuery("SELECT * FROM
CIM_VideoControllerResolution")
```

```
    WScript.Echo objObject.Description 'описание
```

```

WScript.Echo objObject.SettingID 'идентификатор набора параметров
WScript.Echo objObject.HorizontalResolution 'горизонтальное разрешение, пиксели
WScript.Echo objObject.VerticalResolution 'вертикальное разрешение, пиксели
WScript.Echo objObject.NumberOfColors 'количество цветов
WScript.Echo objObject.RefreshRate 'частота регенерации, Hz
WScript.Echo

```

Next

Классы Win32_NetworkAdapter и Win32_NetworkAdapterConfiguration

Классы Win32_NetworkAdapter и Win32_NetworkAdapterConfiguration содержат сведения о сетевых адаптерах. Пример получения информации:

On Error Resume Next

```
Set objService = GetObject("winmgmts:{impersonationLevel=impersonate}!\\.\root\CIMV2")
```

```
If Err.Number <> 0 Then
```

```
    WScript.Echo Err.Number & ": " & Err.Description
```

```
    WScript.Quit
```

```
End If
```

```
For Each objNtw In objService.ExecQuery("SELECT * FROM Win32_NetworkAdapter")
```

```
    WScript.Echo objNtw.SystemName 'имя компьютера
```

```
    WScript.Echo objNtw.Caption 'наименование устройства
```

```
    WScript.Echo objNtw.Name 'наименование устройства
```

```
    WScript.Echo objNtw.ServiceName 'краткое наименование устройства
```

```
    WScript.Echo objNtw.Description 'описание устройства
```

```
    WScript.Echo objNtw.Manufacturer 'производитель
```

```
    WScript.Echo objNtw.AdapterType 'тип устройства
```

```
    WScript.Echo objNtw.DeviceID 'идентификатор устройства
```

```
    WScript.Echo objNtw.PNPDeviceID 'идентификатор устройства Plug-and-Play
```

```
    WScript.Echo objNtw.Index 'индекс сетевого адаптера в системном реестре
```

```
    WScript.Echo objNtw.MACAddress 'MAC - адрес
```

```
    WScript.Echo
```

Next

Класс Win32_NetworkAdapterConfiguration предоставляет ряд свойств сетевого адаптера, а также включает дополнительные методы, которые поддерживают управление протоколами TCP/IP и IPX, службами DNS, DHCP, WINS и т.д. Перечень свойств и методов этого класса достаточно обширен; для получения информации обращайтесь к MSDN.

Класс Win32_DesktopMonitor

Класс Win32_DesktopMonitor предоставляет сведения о мониторе, подключённом к компьютерной системе. Пример получения информации:

On Error Resume Next

```
Set objService = GetObject("winmgmts:{impersonationLevel=impersonate}!\\.\root\CIMV2")
```

```
If Err.Number <> 0 Then
```

```
    WScript.Echo Err.Number & ": " & Err.Description
```

```
    WScript.Quit
```

```
End If
```

```
For Each objMon In objService.ExecQuery("SELECT * FROM Win32_DesktopMonitor")
```

```
    WScript.Echo objMon.SystemName 'имя компьютера
```

```
    WScript.Echo objMon.Caption 'наименование устройства
```

```
    WScript.Echo objMon.Description 'описание устройства
```

```
    WScript.Echo objMon.DeviceID 'идентификатор устройства
```

```
    WScript.Echo objMon.PNPDeviceID 'идентификатор устройства Plug-and-Play
```

```

WScript.Echo objMon.MonitorManufacturer 'производитель
WScript.Echo objMon.PixelsPerXLogicalInch 'разрешающая способность монитора по
горизонтали
WScript.Echo objMon.PixelsPerYLogicalInch 'разрешающая способность монитора по
вертикали
WScript.Echo objMon.ScreenHeight 'логическая высота дисплея
WScript.Echo objMon.ScreenWidth 'логическая ширина дисплея
Next
Класс Win32_FloppyDrive

```

Класс Win32_FloppyDrive предоставляет сведения о дисководах гибких дисков. Пример получения информации:

```

On Error Resume Next
Set objService = GetObject("winmgmts:{impersonationLevel=impersonate}!\\.\root\CIMV2")
If Err.Number <> 0 Then
    WScript.Echo Err.Number & ": " & Err.Description
    WScript.Quit
End If
For Each objFloppy In objService.ExecQuery("SELECT * FROM Win32_FloppyDrive")
    WScript.Echo objFloppy.SystemName 'имя компьютера
    WScript.Echo objFloppy.Caption 'наименование устройства
    WScript.Echo objFloppy.Description 'описание устройства
    WScript.Echo objFloppy.DeviceID 'идентификатор устройства
    WScript.Echo objFloppy.Manufacturer 'производитель
Next
Класс Win32_DiskDrive

```

Класс Win32_DiskDrive предоставляет сведения о дисководах. Пример получения информации:

```

On Error Resume Next
Set objService = GetObject("winmgmts:{impersonationLevel=impersonate}!\\.\root\CIMV2")
If Err.Number <> 0 Then
    WScript.Echo Err.Number & ": " & Err.Description
    WScript.Quit
End If
For Each objDisk In objService.ExecQuery("SELECT * FROM Win32_DiskDrive")
    WScript.Echo objDisk.SystemName 'имя компьютера
    WScript.Echo objDisk.Caption 'наименование устройства
    WScript.Echo objDisk.Model 'модель, указанная производителем
    WScript.Echo objDisk.Description 'описание устройства
    WScript.Echo objDisk.DeviceID 'идентификатор устройства
    WScript.Echo objDisk.PNPDeviceID 'идентификатор устройства Plug-and-Play
    WScript.Echo objDisk.Manufacturer 'производитель
    WScript.Echo objDisk.Index 'номер диска (если 0xFF - не отображает физический диск)
    WScript.Echo objDisk.InterfaceType 'тип интерфейса (IDE, SCSI)
    WScript.Echo objDisk.MediaType 'тип носителя (Removable media, Fixed hard disk и
т.д.)
    WScript.Echo objDisk.SCSIBus 'номер шины SCSI
    WScript.Echo objDisk.SCSILogicalUnit 'номер SCSI устройства
    WScript.Echo objDisk.SCSIPort 'номер порта SCSI
    WScript.Echo objDisk.SCSITargetId 'идентификационный номер SCSI

```

```

WScript.Echo objDisk.TotalHeads 'количество головок
WScript.Echo objDisk.BytesPerSector 'количество байт в секторе
WScript.Echo objDisk.SectorsPerTrack 'количество секторов на дорожке
WScript.Echo objDisk.TracksPerCylinder 'количество дорожек в цилиндре
WScript.Echo objDisk.TotalCylinders 'количество цилиндров
WScript.Echo objDisk.TotalSectors 'общее количество секторов
WScript.Echo objDisk.TotalTracks 'общее количество дорожек
WScript.Echo objDisk.Size 'размер диска (по количеству цилиндров, дорожек, секторов
и размеру сектора)
WScript.Echo objDisk.Partitions 'количество разделов на диске
WScript.Echo
Next
Класс Win32_CDROMDrive

```

Класс Win32_CDROMDrive предоставляет сведения об устройствах CD-ROM. Пример получения информации:

```

On Error Resume Next
Set objService = GetObject("winmgmts:{impersonationLevel=impersonate}!\\.\root\CIMV2")
If Err.Number <> 0 Then
    WScript.Echo Err.Number & ": " & Err.Description
    WScript.Quit
End If
For Each objDisk In objService.ExecQuery("SELECT * FROM Win32_CDROMDrive")
    WScript.Echo objDisk.SystemName 'имя компьютера
    WScript.Echo objDisk.Caption 'наименование устройства
    WScript.Echo objDisk.Description 'описание устройства
    WScript.Echo objDisk.DeviceID 'идентификатор устройства
    WScript.Echo objDisk.Manufacturer 'производитель
    WScript.Echo objDisk.Id 'drive letter
    WScript.Echo objDisk.Size 'размер диска
    WScript.Echo objDisk.VolumeName 'метка тома
    WScript.Echo objDisk.VolumeSerialNumber 'серийный номер тома
Next

```