**Конфигурация аппаратных средств персонального компьютера**

Курсовой проект по дисциплине «Архитектура ЭВМ»

2005

**Ведение**

Общеизвестно, что системный блок современного компьютера состоит из отдельных модулей, объединённых в одно целое и выполняющих каждый свою , определённую функцию.

На сегодняшний день появилось огромное количество аппаратных средств персональных компьютеров различных производителей, которые выполняют самые разные задачи: ввод информации, выполнение кода программ, вывод результатов работы и т.д. Порой становится важной возможность определения конфигурации при не имении на руках документации на установленное оборудование и без разборки системного блока. В этом случае приходят на помощь программы специально разработанные для решения подобного рода задач – программы определения конфигурации компьютера.

Данная работа посвящена разработке именно такой программы. Программы, которая была бы способна воссоздать внутреннее устройство компьютера с классификацией модулей и информацией по каждому из них.

1. Теоретические сведения

Операционные системы Windows, начиная Windows ’95, имеют условную структуру называемую системным реестром, в которой содержится вся информация об оборудовании персонального компьютера и установленном на нём программном обеспечении, а также служебная, необходимая для оптимального функционирования самой системы и правильного выполнения программных компонентов.

Что такое системный реестр?

Системный реестр - база данных, которая сохраняет параметры

настройки для 32 разрядных версий Microsoft Windows включая; Windows 95, 98 и NT. Он содержит информацию и параметры настройки для всех аппаратных средств, программ, пользователей, и свойств PC. Каждый раз, когда пользователь делает изменения в параметрах настройки Панели управления, или в ассоциациях файлов, системной настройке, или в установленном программном обеспечении, изменения отражаются и сохраняются в системном реестре.

Где находится системный реестр?

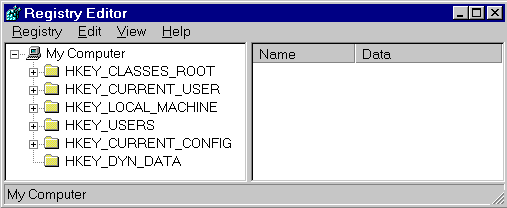
Физические файлы, которые составляют системный реестр,

различаются в зависимости от версии Windows; в Windows 95 и 98 он содержится в двух скрытых файлах каталога Windows, называемыми USER.DAT и SYSTEM.DAT, в то время как в Windows NT файлы содержатся

в каталоге "Windows/System32/Config".

Как можно редактировать системный реестр?

Редактор системного реестра (REGEDIT.EXE) включен в большинство версий Windows (хотя Вы не найдете его в меню "Пуск") он дает возможность просматривать, искать и редактировать данные в пределах системного реестра. Имеется несколько методов для запуска редактора, самый простой - нажать на кнопку "Пуск", затем выбрать Выполнить, дальше в поле "Открыть:" напечатать "regedit" и откроется редактор системного реестра.



Использование Regedit для изменения системного реестра

Как только Regedit открыт, Вы заметите, что левую сторону занимает дерево с папками, а правую содержание выбранной папки.

Чтобы развернуть некоторую ветвь, нажмите на знак "плюс" [+] слева от любой папки, или дважды щелкните на папке. Для отображения содержание папки, нажмите на нужный ключ, и Вы увидите параметры, перечисленные на правой стороне. Вы можете добавить новый ключ или параметр, выбирая пункт «Создать», из меню «Правка», или щелкая правой кнопкой мыши. Так же Вы можете переименовать любой параметр и почти любой ключ тем же методом, что и переименовываете файлы; щелкните правой кнопкой мыши на объекте, и выберете «Переименовать», или нажмите на нем дважды (медленно), или нажмите F2 на клавиатуре. Наконец, Вы можете удалить ключ или параметр, выбирая его, и, нажимая «Delete» на клавиатуре, или щелкая правой кнопкой мыши на параметре, и выбирая «Удалить». Обратите внимание: Всегда резервируйте Ваш системный реестр перед созданием любых изменений в нем. Это даст Вам возможность не переустанавливать операционную систему в случае неправильных действий. Гораздо лучше перестраховаться, чем потом жалеть!

Системный реестр имеет иерархическую структуру, которая подобна структуре каталогов на Вашем жестком диске, а Regedit подобен Проводнику Windows. Каждая главная ветвь (обозначенная значком папки в редакторе системного реестра, см. ниже) называется Корневой и содержит ключи. Каждый ключ может содержать другие ключи (иногда называемые подключами), а также параметры. Параметры содержат фактическую информацию, сохраненную в системном реестре. Имеется три типа параметров; Строковые, Двоичные, и DWORD. Реестр имеет шесть главных ветвей, каждая из которых содержит определенную часть информации. Это следующие ветви:

· HKEY\_CLASSES\_ROOT: Эта ветвь содержит все типы Ваших ассоциаций к файлам, информацию об OLE и данные по ярлыкам.

· HKEY\_CURRENT\_USER: Эта ветвь связана с ветвью HKEY\_USERS, и соответствует пользователю, работающему в настоящее время на PC.

· HKEY\_LOCAL\_MACHINE: Эта ветвь содержит определенную информацию о типах аппаратных средств, программного обеспечения, и других настройках на данном PC, эта информация используется для всех пользователей, которые работают на этом компьютере.

· HKEY\_USERS: Эта ветвь содержит индивидуальные настройки каждого пользователя компьютера, каждый пользователь представлен под ключом SID, расположенном под главной ветвью.

· HKEY\_CURRENT\_CONFIG: Эта ветвь связана с ветвью HKEY\_LOCAL\_MACHINE, и соответствует текущей аппаратной конфигурации.

· HKEY\_DYN\_DATA: Эта ветвь связана с частью HKEY\_LOCAL\_MACHINE, и служит для использования особенностей Plug-&-Play в Windows, этот раздел динамически изменятся, когда устройства добавляются и удаляются из системы.

Импорт и Экспорт параметров настройки системного реестра

Полезной особенностью Редактора системного реестра является способность импортировать и экспортировать параметры системного реестра в текстовый файл, этот текстовый файл имеет расширение .REG, может быть сохранен а затем использован другими людьми для боле простого изменения локальных параметров системного реестра. Вы можете просмотреть содержание этих текстовых файлов, экспортируя ключ в испытательный файл и открыв его в Блокноте. В Редакторе системного реестра выберете ключ, затем из меню "Реестр" выберете "Экспорт файл реестра ... ", укажите имя сохраняемого файла. Если Вы откроете этот файл в Блокноте, то увидите текст, подобный приведенному ниже:

REGEDIT4

[HKEY\_LOCAL\_MACHINE\SYSTEM\Setup]

"SetupType"=dword:00000000

"CmdLine"="setup -newsetup"

"SystemPrefix"=hex:c5,0b,00,00,00,40,36,02

Содержание его весьма простое, REGEDIT4 указывает тип файла, [HKEY\_LOCAL\_MACHINE\SYSTEM\Setup] указывает ключ, где находятся параметры, "SetupType" =dword:00000000 это параметр, часть записи указанная непосредственно после "=" определяется в зависимости от типа данных; Строковое, Двоичное, или DWORD.

Отредактировав этот файл, Вы можете легко произвести изменения в системном реестре, для этого дважды щелкните на нем, или выберете "Импорт файла реестра,,," из меню "Реестр", и параметры будут добавлены в системный реестр.

Вся информация об установленных компонентах и приложениях заносится в реестр при установке операционной системы и обновляется при каждом запуске компьютера, а потому является одним из источников получения данных о конфигурации. Анализ этих данных был взят за основу решения поставленной задачи.

2. Выбор языка программирования

Поскольку работа с реестром подразумевает выполнение задачи под управлением операционной системы Windows, то это обстоятельство повлияло на выбор языка программирования. Необходимо было также обратить внимание на удобство создания интерфейса и отображение всей информации в стандартном окне Windows.

Все поставленные задачи в полной мере решались в среде Visual Basic. Она и была выбрана в качестве средства создания рассматриваемой программы.

3. Описание алгоритма работы программы

3.1. Сбор сведений

.Inf – файлы.

В папке Windows/INF собраны файлы, в которых находится информация об устанавливаемом оборудовании при инсталляции или загрузке операционной системы. И среди прочей служебной информации есть сведения о классе того или иного устройства, описываемого конкретным файлом. Все файлы этой папки мы просматриваем (.inf-файлы, в которых отсутствуют такие сведения нами не рассматриваются, а в программе пропускаются) и создаём список классов всех возможно установленных устройств.

3.2. Сравнение с информацией системного реестра

После того как получен список, каждый найденный класс проверяем на его присутствие в системном реестре и, если он обнаружен, то включаем счётчик возможно установленных устройств данного класса. При каждом пересчёте счётчика делаем обращение к реестру на признак присутствия устройства и, если ответ положительный , сохраняем информацию. Если же устройств данного класса в реестре не обнаружено, то переходим к следующему элементу списка до тех пор, пока не исчерпаем его весь.

После отработки данной части алгоритма имеем распределённый по группам список всех установленных устройств и информацию по каждому из них.

Весь процесс сбора информации сопровождается выводом на экран окна «Сбор сведений» (рис.1), в котором динамически отображаются результаты поиска и идентификации.



3.3. Отображение полученной информации.

Интерфейс пользователя

При разработке интерфейса основное внимание уделялось простоте использования и наглядности. Все возможные типы устройств, устанавливаемые в современных компьютерах, поделились на группы. В результате получилось 6 наименований:

Процессор;

Системная плата;

Память;

Диски;

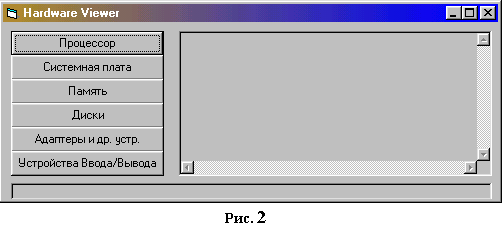
Адаптеры и др.устройства;

Устройства ввода/вывода.

Далее следовало создание формы в среде Visual Basic, на которой были

размещены 6 кнопок (по одной на каждую из выделенных групп), окно просмотра информации по выбранной группе с линиями прокрутки и линейка быстрой помощи.

Внешний вид формы приведён на рисунке 2.



Отображение информации.

По нажатию одной из кнопок вступает в работу соответствующая процедура обработки, которая производит подготовку и вывод нужной информации. В результате в окне справа появляется информация, касающаяся выбранной группы.

Линейка быстрой помощи

При наведении курсора мыши на каждую из кнопок в линейке быстрой помощи появляется информация её функции .

4. Текст программы

Ниже приводится текст разработанной программы:

Код формы HV1(HV1.frm):

Private Sub Form\_Load()

Progress.Show

Call SB\_Sveden

Progress.Hide

End Sub

Private Sub Form\_Unload(Cancel As Integer)

End

End Sub

Private Sub Command1\_Click()

SubK$ = "Hardware\Description\System\CentralProcessor\0"

On Error GoTo Noread

ProcID$ = HV1.RegCtrl1.RReadValue(HK$, SubK$, "Identifier")

Noread: On Error Resume Next

On Error GoTo Noread1

ProcMMX$ = HV1.RegCtrl1.RReadValue(HK$, SubK$, "MMXIdentifier")

Noread1: On Error Resume Next

On Error GoTo Noread2

VendorID$ = HV1.RegCtrl1.RReadValue(HK$, SubK$, "VendorIdentifier")

Noread2: Err.Clear

CpInst$ = ""

If Coproc Then CpInst$ = "Сопроцессор встроенный"

Box1 = ProcID$ & vbCrLf & ProcMMX$ & vbCrLf & VendorID$ & vbCrLf & " " & vbCrLf & CpInst$

On Error GoTo 0

End Sub

Private Sub Command1\_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)

Label1.Caption = "Информация о центральном процессоре."

End Sub

Private Sub Command2\_Click()

Call B\_Text(2)

End Sub

Private Sub Command2\_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)

Label1.Caption = "Информация о системной плате."

End Sub

Private Sub Command3\_Click()

Dim clsMem As New clsMemorySnapshot

Box1 = "Объём физической памяти : " & Format(clsMem.TotalMemory \ 1024, "###,###,###,###,##0") & " KB" & vbCrLf & "Свободно : " & Format(clsMem.FreeMemory \ 1024, "###,###,###,###,##0") & " KB\*" & vbCrLf

End Sub

Private Sub Command3\_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)

Label1.Caption = "Информация о памяти."

End Sub

Private Sub Command4\_Click()

ms = MsgBox("Рекомендуется вставить диски во все дисководы.", vbOKCancel, "ВНИМАНИЕ!")

GetDiskInfo

Box1 = ""

For Ka = 1 To n

tc$ = Str((BytesPerSec(Ka) \* SecsPerClus(Ka) \* TotalNumOfClus(Ka) / 1000) / 1000)

fc$ = Str((BytesPerSec(Ka) \* SecsPerClus(Ka) \* NumOfFreeClus(Ka) / 1000) / 1000)

Box1 = Box1 & "Информация о диске: " & Drives(Ka) & vbCrLf & \_

"Метка тома: " & VNBuffer(Ka) & vbCrLf & \_

"Файловая система: " & vSysBuff(Ka) & vbCrLf & \_

"Серийный номер: " & vSerialNum(Ka) & vbCrLf & \_

"Тип диска: " & TypeOfDrive(Ka) & vbCrLf & \_

"Общее количество кластеров: " & TotalNumOfClus(Ka) & vbCrLf & \_

"Количество свободных кластеров: " & NumOfFreeClus(Ka) & vbCrLf & \_

"Секторов в кластере: " & SecsPerClus(Ka) & vbCrLf & \_

"Байтов в секторе: " & BytesPerSec(Ka) & vbCrLf & \_

"Емкость: " & tc$ & "mb" & vbCrLf & \_

"Свободно: " & fc$ & "mb" & vbCrLf & " " & vbCrLf

Next

End Sub

Private Sub Command4\_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)

Label1.Caption = "Информация о дисках."

End Sub

Private Sub Command5\_Click()

Call B\_Text(5)

End Sub

Private Sub Command5\_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)

Label1.Caption = "Информация о установленных адаптерах (звук, видео, модем и т.д.)."

End Sub

Private Sub Command6\_Click()

Call B\_Text(6)

End Sub

Private Sub Command6\_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)

Label1.Caption = "Информация о устройствах ввода/вывода (монитор, клавиатура, принтер и т.д.)."

End Sub

Sub B\_Text(Comm As Integer)

Select Case Comm

Case 2

l = 0

k = k0

Case 5

l = 2

k = k2

Case 6

l = 1

k = k1

End Select

For i = 1 To k

s$ = s$ + (Sv(l, i) & vbCrLf)

Next i

Box1 = s$

End Sub

Код формы Progress(Pr.frm):

Private Sub Form\_Load()

DrawWidth = 3

End Sub

Код модуля Module1(Hwm.bas):

Public Declare Function GetDiskFreeSpace Lib "kernel32" Alias "GetDiskFreeSpaceA" (ByVal lpRootPathName As String, lpSectorsPerCluster As Long, lpBytesPerSector As Long, lpNumberOfFreeClusters As Long, lpTotalNumberOfClusters As Long) As Long

Public Declare Function GetDriveType Lib "kernel32" Alias "GetDriveTypeA" (ByVal nDrive As String) As Long

Public Declare Function GetVolumeInformation Lib "kernel32" Alias "GetVolumeInformationA" (ByVal lpRootPathName As String, ByVal lpVolumeNameBuffer As String, ByVal nVolumeNameSize As Long, lpVolumeSerialNumber As Long, lpMaximumComponentLength As Long, lpFileSystemFlags As Long, ByVal lpFileSystemNameBuffer As String, ByVal nFileSystemNameSize As Long) As Long

Public Sv(2, 1000) As String

Public Coproc As Boolean

Public X1, X2, Y1, dX As Integer

Public k0 As Integer

Public k1 As Integer

Public k2 As Integer

Public Const HK$ = "HKEY\_LOCAL\_MACHINE"

Public cpuspd As Long

Public FF As Boolean

Public Drives(100) As String

Public n

Public Ka

Public vSerialNum(100) As Long

Public vCompLen(100) As Long

Public vFlags(100) As Long

Public vSysBuff(100) As String

Public vSysSize(100) As Long

Public SecsPerClus(100) As Long

Public BytesPerSec(100) As Long

Public NumOfFreeClus(100) As Long

Public TotalNumOfClus(100) As Long

Public TypeOfDrive(100) As String

Public VNBuffer(100) As String

Public VNSize(100) As Long

Public Const DRIVE\_CDROM = 5

Public Const DRIVE\_FIXED = 3

Public Const DRIVE\_RAMDISK = 6

Public Const DRIVE\_REMOTE = 4

Public Const DRIVE\_REMOVABLE = 2

Sub SB\_Sveden()

Dim mDir(1000), mDir1, mStr, mDDir(100) As String

Dim mClass, nClass(1000) As String

Dim s, s1 As String

Dim a As Integer

X1 = Progress.Line1.X1: X2 = Progress.Line1.X2

Y1 = Progress.Line1.Y1

ChDir ("C:\WINDOWS\INF")

mDDir(0) = "C:\Windows\INF\"

mDTMP = Dir(mDDir(0), vbDirectory)

i = 0

Do While mDTMP <> ""

If mDTMP <> "." And mDTMP <> ".." Then

If (GetAttr(mDDir(0) & mDTMP) And vbDirectory) = vbDirectory Then

i = i + 1: mDDir(i) = mDTMP

End If

End If

mDTMP = Dir

Loop

On Error GoTo EndFindINF

For j = 1 To i

mDir1 = Dir("C:\Windows\INF\" + mDDir(j) + "\\*.inf")

While mDir1 <> ""

a = a + 1

mDir(a) = mDDir(0) + mDDir(j) + "\" + mDir1

mDir1 = Dir()

Wend

Next j

mDir1 = Dir("C:\WINDOWS\INF\\*.inf")

While mDir1 <> ""

a = a + 1

mDir(a) = mDDir(0) + mDir1

mDir1 = Dir()

Wend

EndFindINF:

Err.Clear

dX = (X2 - X1) / a

For i = 1 To a

On Error GoTo 0

Open mDir(i) For Input As #1

XE = X1 + (dX \* i)

Progress.Line (X1, Y1)-(XE, Y1), &H8000000D

f = 0

sClFind:

If Not (EOF(1)) And f = 0 Then

Input #1, mClass

If Mid(mClass, 1, 5) = "Class" And (Mid(mClass, 6, 1) = "=" Or Mid(mClass, 6, 1) = " ") Then

a1 = a1 + 1: f = 1

mClass = Mid(mClass, 7)

For j = 1 To Len(mClass)

mStr = Mid(mClass, j, 1)

If mStr <> " " And mStr <> "=" And mStr <> Chr(34) Then nClass(a1) = nClass(a1) + mStr

Next j

For j = 1 To a1 - 1

s = StrConv(nClass(a1), vbLowerCase)

s1 = StrConv(nClass(j), vbLowerCase)

If s = s1 Then nClass(a1) = "": a1 = a1 - 1: f = 0: Exit For

Next j

If f = 1 Then

If nClass(a1) <> "DiskDrive" And nClass(a1) <> "NetClient" And nClass(a1) <> "NetService" And nClass(a1) <> "NetTrans" And nClass(a1) <> "CDROM" Then Call FClassCH(nClass(a1))

End If

Else: GoTo sClFind

End If

End If

Close #1

Next i

End Sub

Sub FClassCH(FClass As String)

Num$ = "\0000"

For i = 0 To 1999

tmp$ = Mid(Str(i), 2)

tmp1 = Len(tmp$)

Mid(Num$, 6 - tmp1, tmp1) = tmp$

SubK$ = "System\CurrentControlSet\Services\Class\" + FClass + Num$

On Error GoTo NoDev

DDesc$ = HV1.RegCtrl1.RReadValue(HK$, SubK$, "DriverDesc")

On Error GoTo 0

If i = 0 Then

DD$ = " "

Call GroupDev(FClass, DD$, "")

SubK$ = "System\CurrentControlSet\Services\Class\" + FClass

DD$ = HV1.RegCtrl1.RReadValue(HK$, SubK$, "")

Call GroupDev(FClass, DD$, "")

DD$ = String(70, "-")

Call GroupDev(FClass, DD$, "")

End If

If DDesc$ <> "Coprocessor" And DDesc$ <> "Сопроцессор" Then Call GroupDev(FClass, DDesc$, Num$) Else Coproc = True

NoDev: If Err <> 0 Then Exit For

Next i

Err.Clear

End Sub

Sub GroupDev(DClass, DDsc, Nm As String)

If DClass = "System" Or DClass = "fdc" Or DClass = "hdc" Or DClass = "Infrared" Then k0 = k0 + 1: Sv(0, k0) = DDsc: Exit Sub

If DClass = "MTD" Or DClass = "MultiFunction" Or DClass = "PCMCIA" Or DClass = "Ports" Then k0 = k0 + 1: Sv(0, k0) = DDsc: Exit Sub

If DClass = "USB" Then k0 = k0 + 1: Sv(0, k0) = DDsc: Exit Sub

If DClass = "Monitor" Or DClass = "Keyboard" Or DClass = "Mouse" Or DClass = "Printer" Then k1 = k1 + 1: Sv(1, k1) = DDsc: Exit Sub

SubK$ = "System\CurrentControlSet\Services\Class\" + DClass + Nm

On Error GoTo NoMD

MDId$ = HV1.RegCtrl1.RReadValue("HKEY\_LOCAL\_MACHINE", SubK$, "MatchingDeviceId")

On Error GoTo 0

If Mid(MDId$, 1, 3) = "PCI" Then DDsc = "(PCI) " + DDsc

If Mid(MDId$, 1, 6) = "ISAPNP" Then DDsc = "(ISA) " + DDsc

NoMD:

k2 = k2 + 1: Sv(2, k2) = DDsc

Err.Clear

End Sub

Sub GetDiskInfo()

n = 0

For i = 65 To 90

If GetDriveType(Chr$(i) & ":" & "\") <> 1 Then n = n + 1: Drives(n) = Chr$(i) & ":" & "\"

Next i

For i = 1 To n

Call GetDiskFreeSpace(Drives(i), SecsPerClus(i), BytesPerSec(i), NumOfFreeClus(i), TotalNumOfClus(i))

Select Case GetDriveType(Drives(i))

Case DRIVE\_CDROM

TypeOfDrive(i) = "CD-ROM"

Case DRIVE\_REMOVABLE

TypeOfDrive(i) = "Floppy disk"

Case DRIVE\_FIXED

TypeOfDrive(i) = "Hard disk drive"

Case DRIVE\_RAMDISK

TypeOfDrive(i) = "Virtual disk"

Case DRIVE\_REMOTE

TypeOfDrive(i) = "Net disk"

Case Else

End Select

Next

For i = 1 To n

VNBuffer(i) = Space$(255)

VNSize(i) = 255

vSysBuff(i) = Space$(255)

vSysSize(i) = 255

vFlags(i) = 0

vCompLen(i) = 255

vSerialNum(i) = 255

lRet = GetVolumeInformation(Drives(i), VNBuffer(i), VNSize(i), vSerialNum(i), vCompLen(i), vFlags(i), vSysBuff(i), vSysSize(i))

If lRet = 1 Then VNBuffer(i) = Left$(VNBuffer(i), Len(RTrim$(VNBuffer(i))) - 1): vSysBuff(i) = Left$(vSysBuff(i), Len(RTrim$(vSysBuff(i))) - 1): vSerialNum(i) = Left$(vSerialNum(i), Len(RTrim$(vSerialNum(i))) - 1)

If lRet = False Then VNBuffer(i) = "None": vSysBuff(i) = "None"

Next

End Sub

Код класса clsMemorySnapshot(Memory.cls)

Option Explicit

Private Type MEMORYSTATUS

dwLength As Long

dwMemoryLoad As Long

dwTotalPhys As Long

dwAvailPhys As Long

dwTotalPageFile As Long

dwAvailPageFile As Long

dwTotalVirtual As Double

dwAvailVirtual As Double

End Type

Private Declare Sub GlobalMemoryStatus Lib "kernel32" \_

(lpBuffer As MEMORYSTATUS)

Private mmemMemoryStatus As MEMORYSTATUS

Public Property Get FreeMemory() As Long

FreeMemory = mmemMemoryStatus.dwAvailPhys

End Property

Public Property Get TotalMemory() As Long

TotalMemory = mmemMemoryStatus.dwTotalPhys

End Property

Public Property Get TotalVirtualMemory() As Double

TotalVirtualMemory = mmemMemoryStatus.dwTotalVirtual

End Property

Public Property Get AvailableVirtualMemory() As Double

AvailableVirtualMemory = mmemMemoryStatus.dwAvailVirtual

End Property

Private Sub Class\_Initialize()

mmemMemoryStatus.dwLength = Len(mmemMemoryStatus)

GlobalMemoryStatus mmemMemoryStatus

End Sub

Public Sub Refresh()

GlobalMemoryStatus mmemMemoryStatus

End Sub

Заключение

Программа полностью выполнила все поставленные перед ней задачи при тестировании, что позволяет сделать вывод о её пригодности для определения конфигурации компьютера и использования в качестве дополнения к, уже имеющимся в составе операционной системы Windows , средствам получения информации об аппаратных средствах с более простым и удобным интерфейсом.