

Практическая работа 15

Получение системных параметров компьютера

Цель занятия: Получить практический опыт разработки модулей для определения параметров системы

Перечень оборудования и программного обеспечения

Персональный компьютер
Microsoft Office (Word, Visio)
Microsoft Visual Studio

Краткие теоретические сведения

Получение параметров компьютера

Есть два основных инструмента, с помощью которых можно извлечь информацию из любой версии Windows: **System.Environment** и **System.Management**.

Из двух этих `system.environment` проще в использовании; это просто набор неизменяемых переменных, которые выдают информацию о версии операционной системы, числе процессоров, количестве жестких дисков и имени компьютера. Узнать данную информацию можно в любое время.

А вот `system.management`, с другой стороны, хотя тоже не так уж и сложен в использовании, требует обработки полученных данных, чтобы добраться до каких-нибудь конкретных деталей. Однако, `system.management` отличается тем, что может выдать больший объем информации о компьютере.

using System.Diagnostics

Класс `Environment` предоставляет сведения о текущей среде и платформе, а также необходимые для управления ими средства.

№	Свойство	Действие
1.	<code>MachineName</code>	Возвращает имя NetBIOS данного локального компьютера.
2.	<code>CurrentDirectory</code>	Возвращает или задает полный путь к текущей рабочей папке.
3.	<code>OSVersion</code>	Получает объект <code>OperatingSystem</code> , который содержит идентификатор текущей платформы и номер версии
4.	<code>ProcessorCount</code>	Возвращает число процессоров на текущем

		компьютере.
5.	SystemDirectory	Возвращает полный путь к системному каталогу.
6.	TickCount	Возвращает время, истекшее с момента загрузки системы (в миллисекундах).
7.	SystemPageSize	Получает количество байтов на странице памяти операционной системы.
8.	UserDomainName	Возвращает имя сетевого домена, связанное с текущим пользователем
9.	UserInteractive	Возвращает значение, позволяющее определить, выполняется ли текущий процесс в режиме взаимодействия с пользователем.
10.	UserName	Возвращает имя пользователя, который на данный момент выполнил вход в операционную систему Windows
11.	Version	Получает объект Version, который описывает основной и дополнительный номера, а также номер построения и редакции среды CLR
12.	WorkingSet	Возвращает объем физической памяти, сопоставленной контексту процесса

```
private void butProcCount2_Click(object sender, EventArgs e)
{
    MessageBox.Show(
        String.Format(
            "Число процессоров: {0}", Environment.ProcessorCount.ToString()));
}
```

Для определения операционной системы нужно использовать свойство `os.Version` класса `Environment` из пространства имен `System`.

```
private void button1_Click(object sender, EventArgs e)
```

```
{
    OperatingSystem os = Environment.OSVersion;
    listBox1.Items.Add(os.Version);
    listBox1.Items.Add(os.Platform);
    listBox1.Items.Add(os.ServicePack);
    listBox1.Items.Add(os.VersionString);
}
```

```
private void CheckOSVersion()
```

```
{
    OperatingSystem os = Environment.OSVersion;
    Version version = os.Version;
    if ((version.Major == 5) && (version.Minor == 1) || version.Major >= 6)
```

```

{
    MessageBox.Show("Программа может запускаться в вашей
операционной системе");
}
else
{
    MessageBox.Show
        ("Эта версия операционной системы не поддерживается." +
        "\r\n Используйте Windows XP или Windows Vista");
}
}

```

```

private void butCheckOS_Click(object sender, EventArgs e)
{
    CheckOSVersion();
}
}

```

Определение имени пользователя системы

MessageBox.Show(SystemInformation.UserName);

Класс **System.Windows.Forms.SystemInformation** представляет много других свойств для получения системной информации.

```

private void Form1_Load(object sender, EventArgs e)
{
    // Получим системную информацию для локальной машины.
    AddSysInfoItem("ComputerName",
SystemInformation.ComputerName.ToString());
    AddSysInfoItem("UserDomainName",
SystemInformation.UserDomainName.ToString());
    AddSysInfoItem("UserName", SystemInformation.UserName.ToString());
    AddSysInfoItem("Network", SystemInformation.Network.ToString());
    AddSysInfoItem("Secure", SystemInformation.Secure.ToString());
    AddSysInfoItem("Monitor Count",
SystemInformation.MonitorCount.ToString());
    AddSysInfoItem("MousePresent",
SystemInformation.MousePresent.ToString());
    AddSysInfoItem("MouseButtons",
SystemInformation.MouseButtons.ToString());
    AddSysInfoItem("MouseButtonsSwapped",
SystemInformation.MouseButtonsSwapped.ToString());
    AddSysInfoItem("Mouse Wheel",
SystemInformation.MouseWheelPresent.ToString());
}

```

```

        AddSysInfoItem("UserInteractive",
SystemInformation.UserInteractive.ToString());
    }

```

Работа с реестром

Windows хранит море информации о системе, настройках программ, пользователях и множество других данных в специальном хранилище—реестре. Существует возможность улучшить работу программ путем изменения параметров, содержащихся в реестре. В состав .NET Framework входит класс Microsoft.Win32.Registry, специально предназначенный для работы с реестром. Пример позволяет изменить заголовок браузера Internet Explorer.

```

private void button1_Click(object sender, EventArgs e)
    {
        RegistryKey newIETitle = Registry.CurrentUser.OpenSubKey
            (@"SOFTWARE\Microsoft\Internet Explorer\Main", true);
        newIETitle.SetValue("Window Title", "Изменениеназвания IE");
        newIETitle.Close();
        MessageBox.Show("Закройте IE и запустите его снова");
    }
private string GetProcessorArchitecture()
    {
        RegistryKey environmentKey = Registry.LocalMachine; // раздел HKLM
        environmentKey = environmentKey.OpenSubKey
            (@"System\CurrentControlSet\Control\Session Manager\Environment",
false);
        string strEnvironment =
environmentKey.GetValue("PROCESSOR_ARCHITECTURE").ToString();
        return strEnvironment;
    }
private void butDetectBitVersion_Click(object sender, EventArgs e)
    {
        MessageBox.Show(GetProcessorArchitecture());
    }

```

Быстрый взгляд на System.Management

Возможности system.management поистине огромны; даже удивительно, если вы раньше не знали о том, как можно использовать это пространство имен. Доступ к нему несложен. Существует 430 вариантов обращения за необходимой информацией. К примеру, если вам требуется узнать, каковы возможности вашего процессора, то необходимо извлечь информацию из

Win32_Processor

Информация о классе. Вы получаете доступ к информации путем создания объекта ManagementClass следующим образом:

```
ManagementClass myManagementClass = new  
ManagementClass("Win32_Processor");
```

а затем доступ к свойствам для данного класса:

```
ManagementObjectCollection myManagementCollection =  
myManagementClass.GetInstances ();  
PropertyDataCollection myProperties = myManagementClass.Properties;
```

Которые затем можно перебрать, используя два вложенных друг в друга цикла:

```
foreach (ManagementObject obj in myManagemenCollection)  
{  
    foreach (PropertyData property in myProperties)  
    {  
        // Доступ к вашим свойствам здесь, используя 'property'  
    }  
}
```

Переменная “property“ во внутреннем цикле будет содержать имя ключа и тип значения. Можно создавать собственные классы обертки, но здесь я просто демонстрирую базовые возможности.

Объединяя все это вместе, мы получаем следующий код, выдающий основную информацию о нашем процессоре:

```
using System;  
using System.Management;  
using System.Collections.Generic;  
  
namespace YourComputer  
{  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            ManagementClass myManagementClass = new  
ManagementClass("Win32_Processor");  
            ManagementObjectCollection myManagementCollection =  
myManagementClass.GetInstances();
```

```

PropertyDataCollection myProperties = myManagementClass.Properties;
Dictionary myPropertyResults = new Dictionary();

foreach (var obj in myManagementCollection)
{
    foreach (var myProperty in myProperties)
    {
        myPropertyResults.Add(myProperty.Name,
            obj.Properties[myProperty.Name].Value);
    }
}

foreach (var myPropertyResult in myPropertyResults)
{
    Console.WriteLine("{0}: {1}", myPropertyResult.Key,
        myPropertyResult.Value);
}

Console.ReadKey();
}
}
}

```

Который, при запуске на моей машине, дает мне:

```

file:///C:/Users/npo/documents/visual studio 2010/Projects/Your-Compu
AddressWidth: 64
Architecture: 9
Availability: 3
Caption: Intel64 Family 6 Model 42 Stepping 7
ConfigManagerErrorCode:
ConfigManagerUserConfig:
CpuStatus: 1
CreationClassName: Win32_Processor
CurrentClockSpeed: 2100
CurrentVoltage: 0
DataWidth: 64
Description: Intel64 Family 6 Model 42 Stepping 7
DeviceID: CPU0
ErrorCleared:
ErrorDescription:
ExtClock: 1333
Family: 206
InstallDate:
L2CacheSize: 256
L2CacheSpeed:
L3CacheSize: 3072
L3CacheSpeed: 0
LastErrorCode:
Level: 6
LoadPercentage: 15
Manufacturer: GenuineIntel
MaxClockSpeed: 2100
Name: Intel(R) Core(TM) i3-2310M CPU @ 2.10GHz
NumberOfCores: 2
NumberOfLogicalProcessors: 4
OtherFamilyDescription:
PNPDeviceID:
PowerManagementCapabilities:
PowerManagementSupported: False
ProcessorId: BFEFBFF000206A7
ProcessorType: 3
Revision: 10759
Role: CPU
SocketDesignation: CPU1
Status: OK
StatusInfo: 3
Stepping:
SystemCreationClassName: Win32_ComputerSystem
SystemName: NPO-III
UniqueId:
UpgradeMethod: 1
Version:
VoltageCaps:

```

Сложно повторять полный список известных классов в этой статье. Вместо этого, вы можете найти их все на MSDN здесь:
[https://msdn.microsoft.com/en-us/library/aa389273\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/aa389273(v=vs.85).aspx)

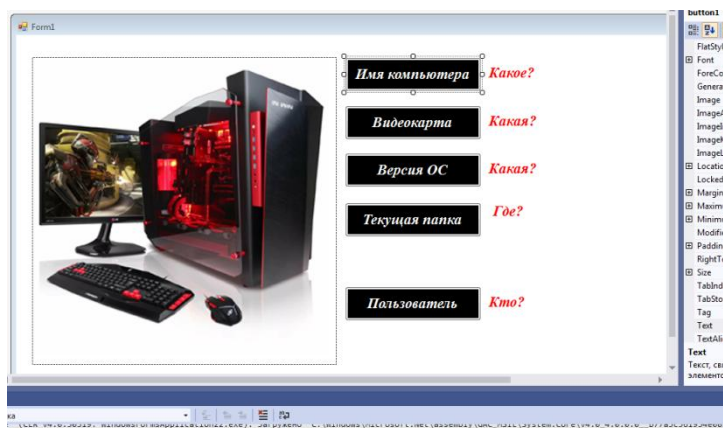
Задания

- 1 Изучить теоретические сведения и задание к работе
- 2 В соответствии с вариантом задания разработать отлаженный модуль.

Порядок выполнения работы

Получение информации о компьютере.

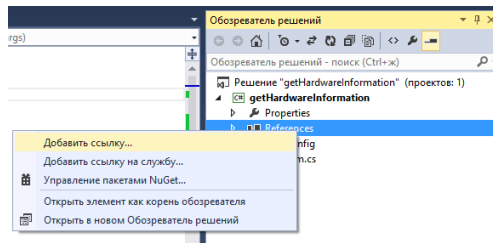
На форме расположим нужные элементы:



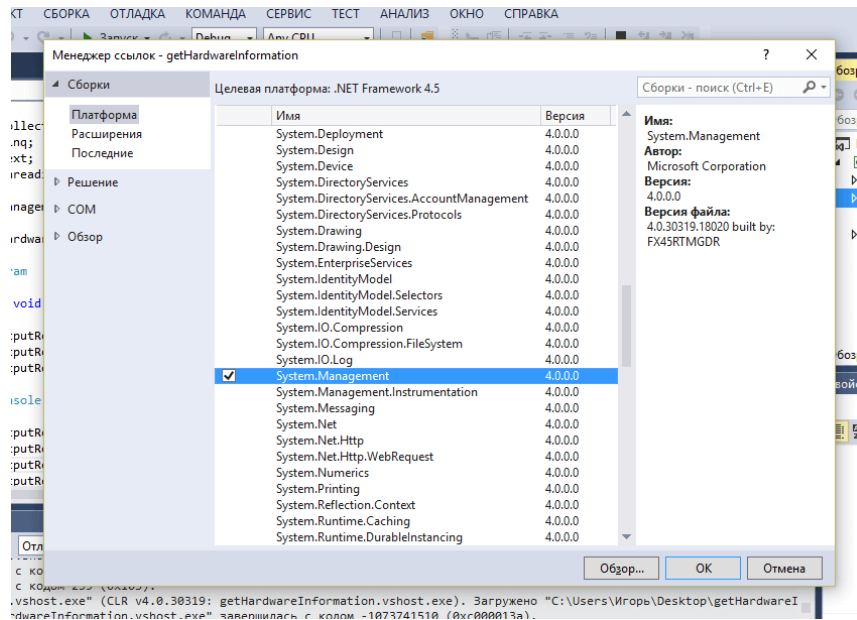
Для начала работы над программой необходимо включить в исходный код пространство имен `System.Diagnostics` и `System.Management`:

```
using System.Diagnostics;  
using System.Management;
```

Но этот код подчеркнётся красным цветом, чтобы это исправить также нужно добавить ссылку на сборку `System.Management`. Для этого жмём правой кнопкой мыши на `References` в “обозревателе решений”.



Далее щёлкаем на пункт “Добавить ссылку...”.
 В появившемся окне ищем сборку System.Management, ставим галочку и жмём кнопку “ОК”.



Теперь для определения названия видеокарты напишем код, который создает экземпляр класса ManagementObjectSearcher, отправляя запрос системе, затем в цикле ищет нужный раздел и передает результат в label.

```

46
47     ссылка:1
48     private void button2_Click(object sender, EventArgs e)
49     {
50         ManagementObjectSearcher searcher = new ManagementObjectSearcher("select * from Win32_VideoController");
51         foreach (ManagementObject share in searcher.Get())
52             label2.Text = share["Caption"].ToString();
53     }
54
55
  
```

Для определения имени пользователя, текущего каталога, версии ОС и имени компьютера воспользуемся методами класса Environment.

```

--
32     ссылка:1
33     private void button5_Click(object sender, EventArgs e)
34     {
35         label15.Text = Environment.UserName;
36     }
37
38     ссылка:1
39     private void button4_Click(object sender, EventArgs e)
40     {
41         textBox1.Text = Environment.CurrentDirectory;
42     }
  
```



```

ссылка: 1
42     private void button3_Click(object sender, EventArgs e)
43     {
44         label3.Text = Convert.ToString(Environment.OSVersion.Version);
45     }
46
ссылка: 1
private void button1_Click(object sender, EventArgs e)
{
    label1.Text = SystemInformation.ComputerName.ToString();
}

```

Результат:



Содержание отчета

- 1 Название работы
- 2 Цель работы
- 3 Технические средства обучения
- 4 Задания (условия задач)
- 5 Порядок выполнения работы
- 6 Ответы на контрольные вопросы
- 7 Вывод

Варианты заданий

1 Разработка модуля, содержащего форму, на которую помещена кнопка и элементы управления Label. При нажатии на кнопку в элементы управления Label выдаются имя машины, имя пользователя, количество и частоту процессоров, описание (description) процессора, название (Name) USB.

2 Разработка модуля, содержащего форму, на которую помещена кнопка и элементы управления Label. При нажатии на кнопку в элементы управления Label выдаются текущий каталог, версия операционной системы, путь к каталогу операционной системы, описание (description) BIOS, CreationClassName для USB.

3 Разработка модуля, содержащего форму, на которую помещена кнопка и элементы управления Label. При нажатии на кнопку в элементы

управления Label выдаются имя компьютера в сети, имя пользователя, подпись (caption) процессора, описание (description) USB, название (Name) видеокарты.

4 Разработка модуля, содержащего форму, на которую помещена кнопка и элементы управления Label. При нажатии на кнопку в элементы управления Label выдаются текущий каталог, версия операционной системы, путь к каталогу операционной системы, логические имена дисков, описание (description) видеокарты, версию BIOS (BIOSVtrSION).

5 Разработка модуля, содержащего форму, на которую помещена кнопка и элементы управления Label. При нажатии на кнопку в элементы управления Label выдаются имя компьютера, машины, имя пользователя, адрес шины (AddressWidth) процессора, описание (description) BIOS, название (Name) USB.

6 Разработка модуля, содержащего форму, на которую помещена кнопка и элементы управления Label. При нажатии на кнопку в элементы управления Label выдаются версия и каталог операционной системы, название (Name) BIOS, описание (description) видеокарты.

7 Разработка модуля, содержащего форму, на которую помещена кнопка и элементы управления Label. При нажатии на кнопку в элементы управления Label выдаются имя компьютера в сети, имя пользователя, CreationClassName видеокарты, описание (description) USB, CreationClassName кэш.

8 Разработка модуля, содержащего форму, на которую помещена кнопка и элементы управления Label. При нажатии на кнопку в элементы управления Label выдаются текущий каталог, версия операционной системы, путь к каталогу операционной системы, логические имена дисков, описание (description) USB, SystemCreationClassName кэш.

9 Разработка модуля, содержащего форму, на которую помещена кнопка и элементы управления Label. При нажатии на кнопку в элементы управления Label выдаются текущий каталог, путь к каталогу операционной системы, CreationClassName видеокарты и USB, версию BIOS (BIOSVtrSION).

10 Разработка модуля, содержащего форму, на которую помещена кнопка и элементы управления Label. При нажатии на кнопку в элементы управления Label выдаются имя компьютера, машины, имя пользователя, адрес шины (AddressWidth) процессора, серийный номер (SerialNumber) BIOS, описание (description) USB.

11 Разработка модуля, содержащего форму, на которую помещена кнопка и элементы управления Label. При нажатии на кнопку в элементы управления Label выдаются версия и каталог операционной системы, название (Name) видеокарты, описание (description) BIOS, SystemCreationClassName кэш.

12 Разработка модуля, содержащего форму, на которую помещена кнопка и элементы управления Label. При нажатии на кнопку в элементы управления Label выдаются имя компьютера в сети, имя пользователя,

CreationClassName видеокарты, описание (description) USB, CreationClassName BIOS.

13 Разработка модуля, содержащего форму, на которую помещена кнопка и элементы управления Label. При нажатии на кнопку в элементы управления Label выдаются имя машины, имя пользователя, количество и частоту процессоров, описание (description) видеокарты, подпись (caption) USB.

14 Разработка модуля, содержащего форму, на которую помещена кнопка и элементы управления Label. При нажатии на кнопку в элементы управления Label выдаются текущий каталог, версия операционной системы, путь к каталогу операционной системы, описание (description) BIOS, CreationClassName кэш.

15 Разработка модуля, содержащего форму, на которую помещена кнопка и элементы управления Label. При нажатии на кнопку в элементы управления Label выдаются имя компьютера в сети, имя пользователя, подпись (caption) процессора, описание (description) USB, название (Name) видеокарты, версию BIOS (BIOSVtrSION).

16 Разработка модуля, содержащего форму, на которую помещена кнопка и элементы управления Label. При нажатии на кнопку в элементы управления Label выдаются имя компьютера, машины, имя пользователя, адрес шины (AddressWidth) процессора, серийный номер (SerialNumber) BIOS, подпись (caption) USB.

17 Разработка модуля, содержащего форму, на которую помещена кнопка и элементы управления Label. При нажатии на кнопку в элементы управления Label выдаются версия и каталог операционной системы, подпись (caption) BIOS, описание (description) видеокарты, SystemCreationClassName кэш.

18 Разработка модуля, содержащего форму, на которую помещена кнопка и элементы управления Label. При нажатии на кнопку в элементы управления Label выдаются имя компьютера в сети, имя пользователя, подпись (caption) процессора, описание (description) USB, название (Name) видеокарты.

19 Разработка модуля, содержащего форму, на которую помещена кнопка и элементы управления Label. При нажатии на кнопку в элементы управления Label выдаются имя компьютера в сети, имя пользователя, подпись (caption) процессора, описание (description) USB, название (Name) видеокарты, версию BIOS (BIOSVtrSION).

20 Разработка модуля, содержащего форму, на которую помещена кнопка и элементы управления Label. При нажатии на кнопку в элементы управления Label выдаются текущий каталог, версия операционной системы, путь к каталогу операционной системы, описание (description) BIOS, CreationClassName для USB.

21 Разработка модуля, содержащего форму, на которую помещена кнопка и элементы управления Label. При нажатии на кнопку в элементы управления Label выдаются имя компьютера, машины, имя пользователя,

адрес шины (AddressWidth) процессора, серийный номер (SerialNumber) BIOS, подпись (caption) USB.

22 Разработка модуля, содержащего форму, на которую помещена кнопка и элементы управления Label. При нажатии на кнопку в элементы управления Label выдаются текущий каталог, версия операционной системы, путь к каталогу операционной системы, описание (description) BIOS, CreationClassName кэш.

23 Разработка модуля, содержащего форму, на которую помещена кнопка и элементы управления Label. При нажатии на кнопку в элементы управления Label выдаются текущий каталог, версия операционной системы, путь к каталогу операционной системы, логические имена дисков, описание (description) USB, SystemCreationClassName кэш.

Контрольные вопросы

- 1 С помощью каких инструментов можно извлечь информацию о компьютере?
- 2 Какие библиотеки необходимо подключить для работы с данными компьютера?
- 3 Как подключить пространство имен System.Management?
- 4 Какой класс System.Management отвечает за доступ к свойствам компьютера?
- 5 Какой класс предоставляет сведения о текущей среде и платформе?
- 6 С помощью какого класса можно определить имя пользователя системы?
- 7 Какую еще системную информацию предоставляет этот класс?

Используемая литература

1. Гниденко, И. Г. Технология разработки программного обеспечения: учеб. пособие для СПО / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — М.: Издательство Юрайт, 2017.
2. Шарп Джон Ш26 Microsoft Visual C#. Подробное руководство. 8-е изд. — СПб.: Питер, 2017.
3. Васильев А.Н. Программирование на C# для начинающих. Основные сведения. — Москва: Эксмо, 2018.
4. Васильев А.Н. Программирование на C# для начинающих. Особенности языка. — Москва: Эксмо, 2019.
5. <http://msdn.microsoft.com/ru-ru/library/67ef8sbd.aspx>.