Лабораторна робота №2

«Вивчення архітектури персонального комп’ютера за допомогою інфраструктури Windows Management Instrumentation»

**Варіант 21**

**Cоболевський Іван 123-17-1**

**Мета роботи:** Мовою програмування C# написати програму, що збирає та виводить на екран основну системну інформацію про параметри компонентів персонального комп’ютера.

**Організація виконання лабораторної роботи:**

Для виконання лабораторної роботи необхідно вивчити, використовуючи рекомендовану літературу, конспект лекцій і методичні рекомендації до даної роботи, наступні питання:

– основні компоненти персонального комп’ютера;

– реалізація графічного інтерфейсу мовою програмування C# на базі платформи Windows Forms;

– бібліотека System.Managment та колекція керуючих об’єктів ManagementObjectSearcher;

– засоби інфраструктури Windows Management Instrumentation (WMI).

Оформити звіт про виконане завдання.

**Теоретична довідка**

Для отримання інформації про комп’ютер можна використати два методи: − через властивості та методи класу Environment; − через властивості набору класів Win32. Властивості та методи класу Environment надають лише поверхневу інформацію про комп’ютер, тому для отримання вичерпної інформації слід використовувати властивості класів Win32. Для отримання властивостей класів Win32 слід використовувати колекцію керуючих об’єктів ManagementObjectSearcher. ManagementObjectSearcher отримує колекцію керуючих об'єктів відповідно до заданого запиту. Цей клас є одним з найбільш часто використовуваних, запис вказує на отримання відомостей про управління. Наприклад він може використовуватися для перерахування всіх дисків, мережевих адаптерів, процесів та інших керівників об'єктів в системі, або для запиту всіх працюючих мережевих підключень, служб і т. д. При створенні екземпляра, екземпляр цього класу приймає в якості вхідних даних запит Windows Management Instrumentation представлений в ObjectQuery або похідні від нього. Коли Get() викликається метод для цього об'єкта, ManagementObjectSearcher виконує даний запит в заданій області та повертає колекцію керуючих об'єктів, що відповідають запиту в ManagementObjectCollection. Для зручності представлення інформації слід також використовувати наступні методи та властивосиі: Environment.NewLine – перейти на новий рядок (для того щоб розділити інформацію в текстовому полі на категорії); string.Format() – представити об’єкт із колекції керуючих об’єктів у вигляді рядка; Math.Round() – перетворення чисел із байт чи кілобайт у мегабайти та гігабайти. Для виведення повідомлення про розробника додатку будемо виводити вікно повідомлення за допомогою використання функції MessageBox.Show().

**Розробка графічного інтерфейсу** **та опис роботи додатку**

Однією з основних задач при проектуванні та реалізації програмного забезпечення була задача утворення між користувачем та програмою простого інтерфейсу взаємодії. Через це був розроблений графічний інтерфейс, який реалізує можливість звернення до користувача за допомогою системи графічних об’єктів.

**Опис клієнтського додатку.**

Розроблений додаток являє собою єдине вікно з наступними елементами:

– **кнопка «Операційна система»** визначає загальні характеристики операційної системи: її тип, архітектуру, версію та номер збірки, ім’я комп’ютера та активного користувача.

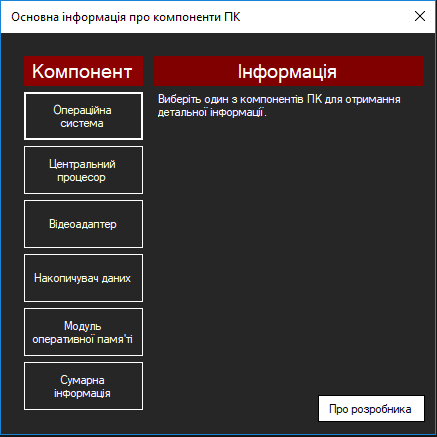
– **кнопка «Центральний процесор»** оброблює та виводить на екран такі характеристики центрального процесора, як модель, тактову частоту, кількість ядер та логічних процесорів;

– **кнопка «Відеоадаптер»** виводить на екран тип адаптера та розмір його графічної пам'яті;

– **кнопка «Накопичувач даних»** надає інформацію про всі встановлені в системі накопичувачі даних, в тому числі – й про флеш-накопичувачі;

– **кнопка «Модуль оперативної пам’яті»** виводить на екран найбільш важливі дані всіх встановлених в системі модулів оперативних запам’ятовуючих пристроїв;

– **кнопка «про розробника»** викликає діалогове вікно, яке інформує користувача про розробника даного додатка.



Аналіз отриманих даних Для детального аналізу даних, отриманих за допомогою розробленого додатку, порівняємо ці дані з аналогічними, отриманих за допомогою програмного забезпечення ‘AIDA64’. Порівняльний аналіз проводиться за п’ятьма категоріями:

− характеристики операційної системи;

− основні параметри центрального процесора;

− основні параметри встановлених відеоадаптерів;

− основні параметри встановлених накопичувачів даних;

− основні параметри встановлених модулів оперативних запам’ятовуючих пристроїв.

|  |  |  |
| --- | --- | --- |
| **Параметр** | **Додаток** | **AIDA64** |
| Тип операційної системи | Майкрософт Windows 10 Pro | Операционная система Microsoft Windows 10 Pro |
| Архітектура | 64-разрядная | Multiprocessor Free (64-bit) |
| Версія | 10.0.17134 | 10.0.17134.706 (Win10 RS4 [1803] April 2018 Update) |
| Номер збірки | 17134 |  |
| Ім'я комп'ютера | DESKTOP-MCPSRTM | DESKTOP-MCPSRTM |
| Користувач | ivanet520@gmail.com | ivane |
|  |  |  |
| **Параметр** | **Додаток** | **AIDA64** |
| Виробник | GenuineIntel | Intel Corporation |
| Модель | Intel64 Family 6 Model 78 Stepping 3 | DualCore Intel Core i5-6300U, 2900 MHz (29 x 100) |
| Тактова частота | 798 МГц | 2400 МГц |
| Кількість ядер | 2 | 2 |
| Кількість логічних процесорів: | 4 | 4 |
|  |  |  |
| **Параметр** | **Додаток** | **AIDA64** |
| Модель: | Intel(R) HD Graphics 520 | Intel(R) HD Graphics 520 |
| Графічний процесор | Intel(R) HD Graphics Family | Intel Skylake-U GT2 - Integrated Graphics Controller |
| Тип адаптера | Internal | Internal |
| Розмір графічної пам'яті | 1024 Мб | 1024 Мб |
|  |  |  |
|  |  |  |
| **Параметр** | **Додаток** | **AIDA64** |
| Модель | Samsung SSD 860 EVO 250GB | Samsung SSD 860 EVO 250GB |
| Виробник | (Стандартные дисковые накопители) | Samsung |
| Тип інтерфейсу | IDE | SATA-III |
| Розмір | 232,88 Гб | 250 ГБ |
| Тип носія | Fixed hard disk media |  |
|  |  |  |
| **Параметр** | **Додаток** | **AIDA64** |
| Виробник | Samsung | Samsung SSD 860 EVO 250GB |
| Ємність | 8192 Мб | 7592 МБ |
| Швидкість пам'яті | 1600 MT/s | 10512 Мб/c |

**Висновок**

При детальному порівнянні інформації з розробленої програми та інформації з ПЗ ‘AIDA64’ було визначено, що дані, отримані за допомогою програмного забезпечення ‘AIDA64’ були більш детальними, ніж дані, отримані за допомогою колекції керуючих об’єктів ManagmentObjectSearcher з бібліотеки System.Managment. Однак, так чи інакше, розроблений додаток виконує всі необхідні функції, а інформація, отримана за його допомогою, є достатньою для звичайних користувачів персонального комп’ютера.

using System;

using System.Collections.Generic;

using System.Windows.Forms;

using System.Management;

using System.Runtime.InteropServices;

namespace PC\_AIDA64

{

public partial class AIDA\_LIGHT : Form

{

//Наступна частина коду використовується для реалізації функції пересування вікна

#region DragWindow

public const int WM\_NCLBUTTONDOWN = 0xA1;

public const int HT\_CAPTION = 0x2;

[DllImportAttribute("user32.dll")]

public static extern int SendMessage(IntPtr hWnd, int Msg, int wParam, int lParam);

[DllImportAttribute("user32.dll")]

public static extern bool ReleaseCapture();

#endregion

public AIDA\_LIGHT()

{

InitializeComponent();

}

//Визначення параметрів операційної системи за натиском кнопки

private void buttonOS\_Click(object sender, EventArgs e)

{

textBoxInfo.Text = "";

InfoOS();

}

//Визначення параметрів CPU за натиском кнопки

private void buttonCPU\_Click(object sender, EventArgs e)

{

textBoxInfo.Text = "";

InfoCPU();

}

//Визначення параметрів графічного процесора за натиском кнопки

private void buttonGPU\_Click(object sender, EventArgs e)

{

textBoxInfo.Text = "";

InfoGPU();

}

//Визначення параметрів носіїв даних за натиском кнопки

private void buttonHDD\_Click(object sender, EventArgs e)

{

textBoxInfo.Text = "";

InfoHDD();

}

//Визначення параметрів модулів оперативної пам'яті за натиском кнопки

private void buttonRAM\_Click(object sender, EventArgs e)

{

textBoxInfo.Text = "";

InfoRAM();

}

//Виведення загальної інформації про ПК за натиском кнопки

private void buttonAll\_Click(object sender, EventArgs e)

{

textBoxInfo.Text = "";

InfoOS();

textBoxInfo.Text += Environment.NewLine + Environment.NewLine;

InfoCPU();

textBoxInfo.Text += Environment.NewLine + Environment.NewLine;

InfoGPU();

textBoxInfo.Text += Environment.NewLine;

InfoHDD();

textBoxInfo.Text += Environment.NewLine;

InfoRAM();

}

//Виклик вікна з інформацією про розробника

private void buttonDeveloper\_Click(object sender, EventArgs e)

{

List<string> about = new List<string>() { "Соболевський Іван", "студент групи 123-17-1", "Національний технічний університет ", "'Дніпровська політехніка'" };

string aboutInfo = "";

foreach (string aboutString in about)

{

aboutInfo += aboutString + '\n';

}

MessageBox.Show(aboutInfo);

}

void InfoOS()

{

textBoxInfo.Text += "Інформація про ОС:" + Environment.NewLine;

ManagementObjectSearcher OperatingSystem = new ManagementObjectSearcher //ініціалізація нової колекції керуючих елементів

("SELECT BuildNumber, Version, RegisteredUser,OSArchitecture, Caption, CSName FROM Win32\_OperatingSystem"); // WMI - запит

foreach (ManagementObject queryObj in OperatingSystem.Get())

{

textBoxInfo.Text += Environment.NewLine + string.Format("Назва: {0}", queryObj["Caption"]);

textBoxInfo.Text += Environment.NewLine + string.Format("Архітектура: {0}", queryObj["OSArchitecture"]);

textBoxInfo.Text += Environment.NewLine + string.Format("Версія: {0}", queryObj["Version"]);

textBoxInfo.Text += Environment.NewLine + string.Format("Номер збірки: {0}", queryObj["BuildNumber"]);

textBoxInfo.Text += Environment.NewLine + string.Format("Ім'я комп'ютера: {0}", queryObj["CSName"]);

textBoxInfo.Text += Environment.NewLine + string.Format("Користувач: {0}", queryObj["RegisteredUser"]);

}

}

void InfoCPU()

{

textBoxInfo.Text += "Інформація про ЦП:" + Environment.NewLine;

ManagementObjectSearcher Processor = new ManagementObjectSearcher

("SELECT NumberOfLogicalProcessors, NumberOfCores,Manufacturer,Description,CurrentClockSpeed FROM Win32\_Processor");

foreach (ManagementObject queryObj in Processor.Get())

{

textBoxInfo.Text += Environment.NewLine + string.Format("Виробник: {0}", queryObj["Manufacturer"]);

textBoxInfo.Text += Environment.NewLine + string.Format("Модель: {0}", queryObj["Description"]);

textBoxInfo.Text += Environment.NewLine + string.Format("Тактова частота : {0}", queryObj["CurrentClockSpeed"]) + " МГц";

textBoxInfo.Text += Environment.NewLine + string.Format("Кількість ядер: {0}", queryObj["NumberOfCores"]);

textBoxInfo.Text += Environment.NewLine + string.Format("Кількість логічних процесорів: {0}", queryObj["NumberOfLogicalProcessors"]);

}

}

void InfoGPU()

{

textBoxInfo.Text += "Інформація про відеоадаптер:" + Environment.NewLine;

ManagementObjectSearcher VideoAdapter = new ManagementObjectSearcher

("SELECT Caption, VideoProcessor,AdapterDACType,AdapterRAM FROM Win32\_VideoController");

foreach (ManagementObject queryObj in VideoAdapter.Get())

{

textBoxInfo.Text += Environment.NewLine + string.Format("Модель: {0}", queryObj["Caption"]);

textBoxInfo.Text += Environment.NewLine + string.Format("Графічний процесор: {0}", queryObj["VideoProcessor"]);

textBoxInfo.Text += Environment.NewLine + string.Format("Тип адаптера: {0}", queryObj["AdapterDACType"]);

textBoxInfo.Text += Environment.NewLine + string.Format("Розмір графічної пам'яті: {0}",

Math.Round(System.Convert.ToDouble(queryObj["AdapterRAM"]) / 1024 / 1024, 2)) + " Мб";

textBoxInfo.Text += Environment.NewLine;

}

}

void InfoHDD()

{

textBoxInfo.Text += "Інформація про ЖД:" + Environment.NewLine;

ManagementObjectSearcher DiskDrive = new ManagementObjectSearcher

("SELECT Manufacturer,Model,InterfaceType,Size,MediaType FROM Win32\_DiskDrive");

foreach (ManagementObject queryObj in DiskDrive.Get())

{

textBoxInfo.Text += Environment.NewLine + string.Format("Виробник: {0}", queryObj["Manufacturer"]);

textBoxInfo.Text += Environment.NewLine + string.Format("Модель: {0}", queryObj["Model"]);

textBoxInfo.Text += Environment.NewLine + string.Format("Тип інтерфейсу: {0}", queryObj["InterfaceType"]);

textBoxInfo.Text += Environment.NewLine + string.Format("Розмір: {0}",

Math.Round(System.Convert.ToDouble(queryObj["Size"]) / 1024 / 1024 / 1024, 2)) + " Гб";

textBoxInfo.Text += Environment.NewLine + string.Format("Тип носія: {0}", queryObj["MediaType"]);

textBoxInfo.Text += Environment.NewLine;

}

}

void InfoRAM()

{

textBoxInfo.Text += "Інформація про ОЗП:" + Environment.NewLine;

ManagementObjectSearcher PhysicalMemory = new ManagementObjectSearcher

("SELECT Manufacturer,Capacity,Speed,Description FROM Win32\_PhysicalMemory");

foreach (ManagementObject queryObj in PhysicalMemory.Get())

{

textBoxInfo.Text += Environment.NewLine + string.Format("Опис: {0}", queryObj["Description"]);

textBoxInfo.Text += Environment.NewLine + string.Format("Виробник: {0}", queryObj["Manufacturer"]);

textBoxInfo.Text += Environment.NewLine + string.Format("Ємність: {0}",

Math.Round(System.Convert.ToDouble(queryObj["Capacity"]) / 1024 / 1024, 2)) + " Мб";

textBoxInfo.Text += Environment.NewLine + string.Format("Швидкість пам'яті: {0}", queryObj["Speed"]) + " MT/s";

textBoxInfo.Text += Environment.NewLine;

}

}

//Реалізація функції пересування вікна

private void Form1\_MouseDown(object sender, MouseEventArgs e)

{

if (e.Button == MouseButtons.Left)

{

ReleaseCapture();

SendMessage(Handle, WM\_NCLBUTTONDOWN, HT\_CAPTION, 0);

}

}

private void label2\_Click(object sender, EventArgs e)

{

}

private void textBoxInfo\_TextChanged(object sender, EventArgs e)

{

}

private void label1\_Click(object sender, EventArgs e)

{

}

}

}