Міністерство освіти і науки України

Центральноукраїнський національний технічний університет

**МЕХАНІКО-ТЕХНОЛОГІЧНИЙ ФАКУЛЬТЕТ**

Кафедра програмування та захисту інформації

**Звіт**

з виконаної лабораторної роботи № 6

дисципліни “ Системне програмування”

на тему

«**ОСНОВИ ВИКОРИСТАННЯ РЕЄСТРУ ОС WINDOWS.**»

Виконав :

студент академічної групи КІ-15

Аннаєв А. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Перевірив :

Викладач

Константинова K.В.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кропивницкий- 2017

**МЕТА:** Отримати практичні навики в використанні функцій WIN API 32,

використання системи повідомлень ОС Windows.

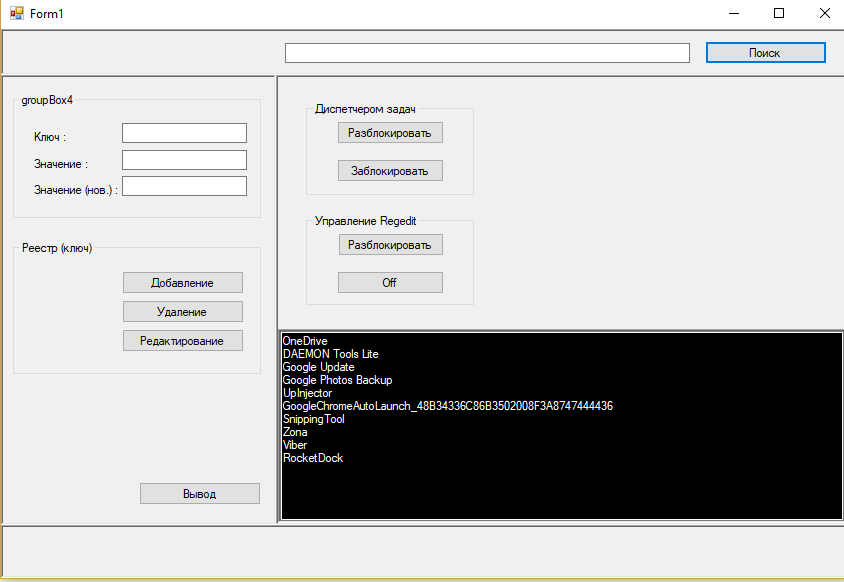
**ЗНАТИ:** Основи RAD (наприклад Builder або Delphi).

**ВМІТИ:** Знати основи розробки та роботи програм під ОС Windows.

**ЗАВДАННЯ**

Використовуючи наявну електронну документацію по напрямку програмування з використанням WIN API (функції наведені у завданні), написати програму з наступним функціоналом.

1. Пошук текстових та числових даних у реєстрі.
2. Додає/видаляє/редагує заданий ключ реєстру.
3. Виводить на екран список програм які автозавантажуються з ОС Windows.
4. Блокує/розблокує доступ до диспетчера задач Windows.
5. Блокує/розблокує доступ до системної програми Regedit.

#region Конструктор

public Form1()

{

InitializeComponent( );

//

// Отображение программ на автозапуске

//

using ( RegistryKey startUp =

Registry.CurrentUser.OpenSubKey

("Software\\Microsoft\\Windows\\CurrentVersion\\Run"))

{

foreach (var name in startUp.GetValueNames( ))

{

txtConsole.Text += name;

txtConsole.Text += Environment.NewLine;

}

}

}

#region Методы

RegFindValue RegFind( RegistryKey key, string find )

{

if (key == null || string.IsNullOrEmpty(find))

return null;

string[] props = key.GetValueNames( );

object value = null;

if (props.Length != 0)

foreach (string property in props)

{

if (property.IndexOf(find, StringComparison.OrdinalIgnoreCase) != -1)

return new RegFindValue(key, property, null, RegFindIn.Property);

value = key.GetValue(property,

null,

RegistryValueOptions.DoNotExpandEnvironmentNames);

if (value is string && ((string)value).IndexOf(find,

StringComparison.OrdinalIgnoreCase) != -1)

return new RegFindValue(key, property, (string)value, RegFindIn.Value);

}

string[] subkeys = key.GetSubKeyNames( );

RegFindValue retVal = null;

if (subkeys.Length != 0)

foreach (string subkey in subkeys)

{

try

{

retVal = RegFind(key.OpenSubKey(subkey,

RegistryKeyPermissionCheck.ReadSubTree), find);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

if (retVal != null)

return retVal;

}

key.Close( );

return null;

}

/// <summary>

/// Установка диспетчера задач

/// </summary>

/// <param name="enable"></param>

public void SetTaskManager( bool flag )

{

RegistryKey key = Registry.CurrentUser.CreateSubKey(

@"Software\Microsoft\Windows\CurrentVersion\Policies\System");

if (flag && key.GetValue("DisableTaskMgr") != null)

key.DeleteValue("DisableTaskMgr");

else

key.SetValue("DisableTaskMgr", "1");

key.Close( );

}

/// <summary>

/// Установка значения Regedit

/// </summary>

/// <param name="enable"></param>

public void SetRegedit( bool flag )

{

RegistryKey key = Registry.CurrentUser.CreateSubKey(

@"Software\Microsoft\Windows\CurrentVersion\Policies\System");

if (flag && key.GetValue("DisableRegistryTools") != null)

key.DeleteValue("DisableRegistryTools");

else

key.SetValue("DisableRegistryTools", 1);

key.Close( );

}

/// <summary>

///

/// </summary>

/// <param name="rkey"></param>

/// <param name="textBox0"></param>

void PrintKeys( RegistryKey rkey, TextBox textBox0 )

{

String[] names = rkey.GetSubKeyNames( );

int icount = 0;

foreach (String s in names)

{

textBox0.Text += s + "\r\n";

icount++;

if (icount >= 10)

break;

}

}

#endregion

#region кнопки

/// <summary>

/// Кнопка включение доступа к диспетчеру задач

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

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

{

btnTaskMOff.Visible = false;

btnTaskMOn.Visible = true;

SetTaskManager(true);

}

/// <summary>

/// Кнопка выключение доступа к диспетчеру задач

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

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

{

btnTaskMOff.Visible = true;

btnTaskMOn.Visible = false;

SetTaskManager(false);

}

/// <summary>

/// Кнопка включение доступа к Regedit

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

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

{

btnRegOn.Visible = true;

btnRegOff.Visible = false;

SetRegedit(true);

}

/// <summary>

/// Кнопка выключение доступа доступа к Regedit

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

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

{

btnRegOn.Visible = false;

btnRegOff.Visible = true;

SetRegedit(false);

}

/// <summary>

/// Добаление ключа

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

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

{

RegistryKey key;

key = Registry.CurrentUser.CreateSubKey("SP\_Labarator\_#\_6");

key.SetValue(txtKeyName.Text, txtKeyValue.Text);

key.Close( );

}

/// <summary>

/// Удаление ключа

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

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

{

try

{

RegistryKey key;

key = Registry.CurrentUser.CreateSubKey("SP\_Labarator\_#\_6");

key.DeleteValue(txtKeyName.Text);

key.Close( );

}

catch(Exception ex)

{

MessageBox.Show(ex.Message );

}

}

/// <summary>

/// Редактирование ключа

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

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

{

RegistryKey key = Registry.CurrentUser;

RegistryKey edit\_key = key.OpenSubKey(txtKeyName.Text, true);

key.SetValue(txtKeyName.Text, txtNewKeyValue.Text);

key.Close( );

}

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

{

RegistryKey key = Registry.Users;

PrintKeys(key, txtConsole);

}

private void btnFind\_Click\_1( object sender, EventArgs e )

{

RegistryKey[] keys = {

Registry.ClassesRoot,

Registry.CurrentConfig,

Registry.LocalMachine,

Registry.CurrentUser,

Registry.Users };

foreach (RegistryKey key in keys)

{

RegFindValue retVal = RegFind(key, txtFindValue.Text);

if (retVal != null)

txtConsole.Text += retVal.Key.Name +

" " + retVal.Property.ToString( ) +

Environment.NewLine + "\r\n";

}

}

#endregion

class RegFindValue

{

RegistryKey regKey;

string mProps;

string mVal;

RegFindIn mWhereFound;

public RegistryKey Key

{

get

{

return regKey;

}

}

public string Property

{

get

{

return mProps;

}

}

public string Value

{

get

{

return mVal;

}

}

public RegFindIn WhereFound

{

get

{

return mWhereFound;

}

}

public RegFindValue( RegistryKey key, string props, string val, RegFindIn where )

{

regKey = key;

mProps = props;

mVal = val;

mWhereFound = where;

}

}

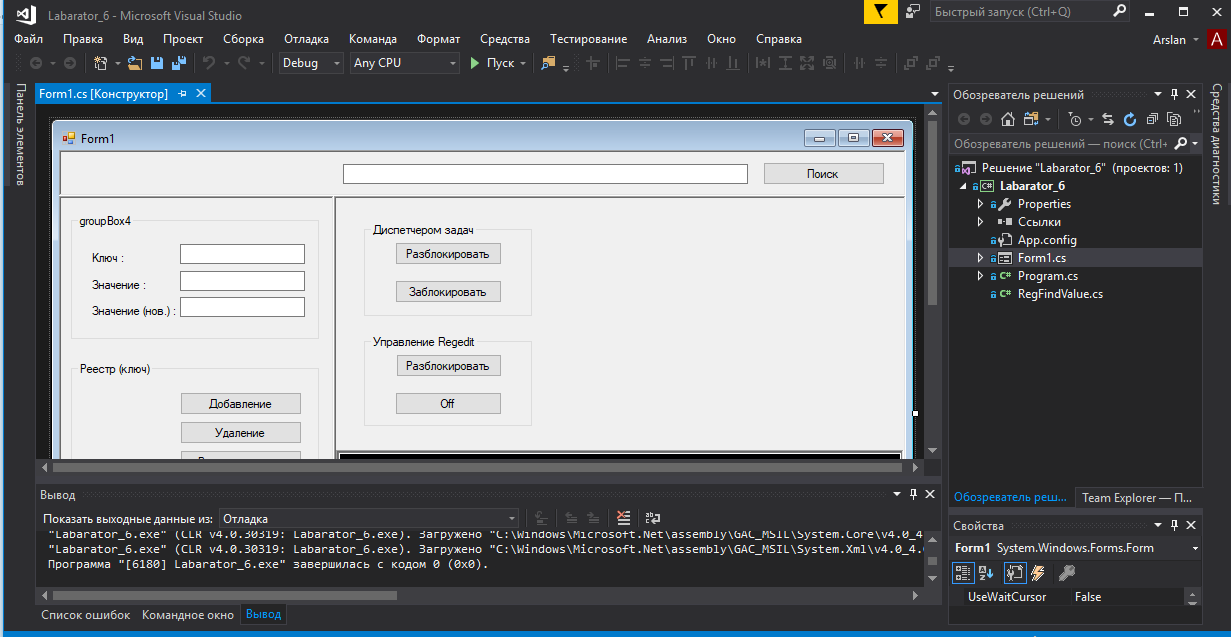
enum RegFindIn

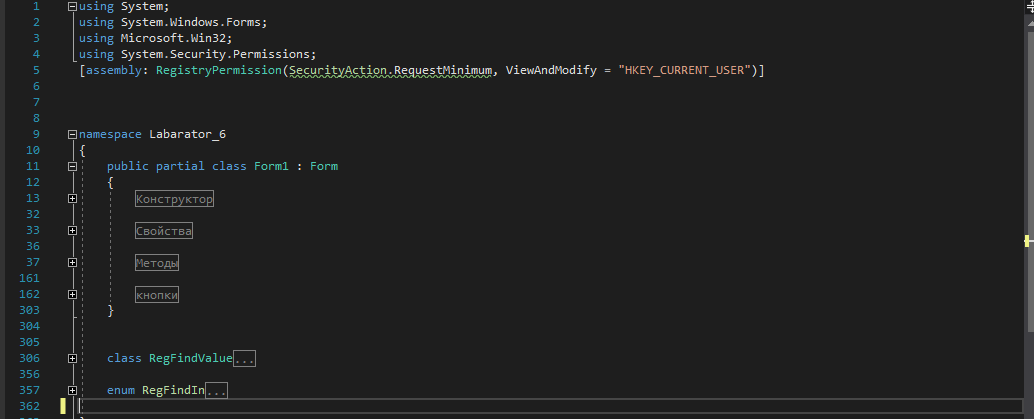
{

Property,

Value

}





**КОНТРОЛЬНІ ЗАПИТАННЯ**

1. Що таке і навіщо потрібен системний реєстр ОС Windows?

Для контролювання програм, та надання доступу для цих програм.

1. Де знаходиться системний реєстр ОС Windows?

C:\Windows\REG.DAT

1. Які існують основні кущі реєстру та навіщо вони потрібні?

Існують такі кущі реєстру: HKEY\_CLASSES\_ROOT,

HKEY\_CURRENT\_USER,

HKEY\_LOCAL\_MACHINE,

HKEY\_USERS,

HKEY\_CURRENT\_CONFIG.

1. Наведіть приклади використання системного реєстру ОС Windows?

Основний приклад – це автозапуск. За допомогою автозапуску ОС Windows запускає програми, які загружаються за замовченням і ці програми беруться с реєстру*.*