**МИНОБРНАУКИ РОССИИ**

Федеральное государственное бюджетное образовательное учреждение высшего образования «Ижевский государственный  
технический университет имени М.Т. Калашникова»

(ФГБОУ ВО «ИжГТУ имени М.Т. Калашникова»)

Кафедра «Вычислительная техника»

Отчет по лабораторной работе №2

по дисциплине «Системное программное обеспечение»

на тему «Резидентные программы»

Вариант 14

Выполнил: студент гр. Б06-781-1

Пермитин И.А.

Проверил: к.т.н., доцент кафедры ВТ

Марков Е.М.

Ижевск 2020

**Задание**

Создать резидентную программу

**Описание использованных технологий**

Лабораторная работа выполнялась в IDE «Visual Studio 2019».

Использованный язык программирования – C#

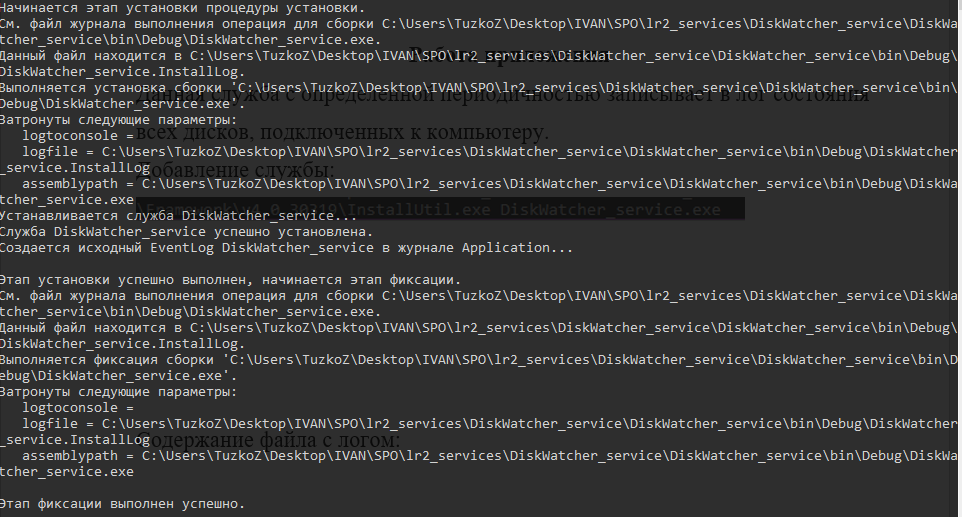
Тип проекта – Служба Windows.

**Работа приложения**

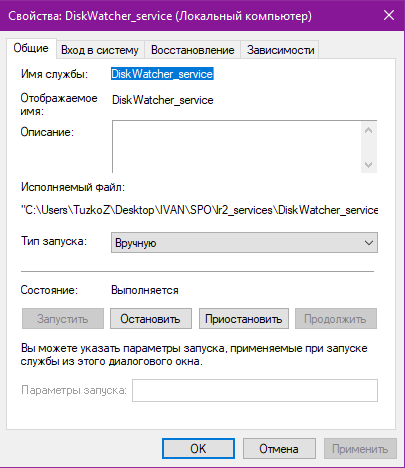
Данная служба с определенной периодичностью записывает в лог состояния всех дисков, подключенных к компьютеру.

Добавление службы:

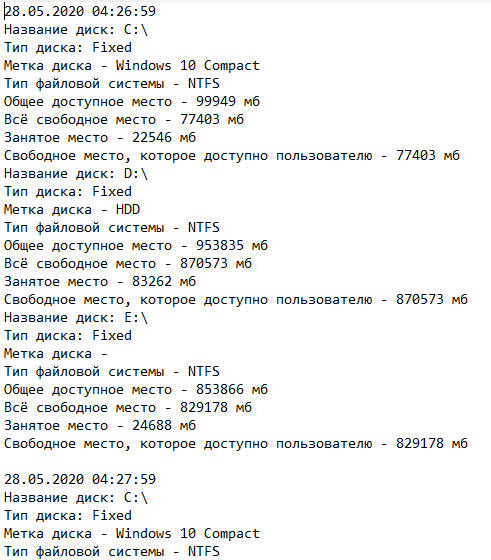




Включение службы:



Содержание файла с логом:



**Листинг программы**

Service1.cs:

using System;

using System.Collections.Generic;

using System.Collections.Specialized;

using System.ComponentModel;

using System.Data;

using System.Diagnostics;

using System.IO;

using System.Linq;

using System.ServiceProcess;

using System.Text;

using System.Threading;

using System.Threading.Tasks;

namespace DiskWatcher\_service

{

public partial class Service1 : ServiceBase

{

DiskWacher diskWatcher;

public Service1()

{

InitializeComponent();

this.CanStop = true;

this.CanPauseAndContinue = true;

}

protected override void OnStart(string[] args)

{

diskWatcher = new DiskWacher();

Thread watcherThread = new Thread(new ThreadStart(diskWatcher.start));

watcherThread.Start();

}

protected override void OnStop()

{

diskWatcher.stop();

Thread.Sleep(1000);

}

}

public class DiskWacher

{

private bool isEnabled;

private int timer = 60000; //минута

public void start()

{

isEnabled = true;

Logger Logger = new Logger("E:\\DiskLog.txt");

while (isEnabled)

{

StringCollection record = new StringCollection();

string rec;

record.Clear();

record.Add(String.Format("{0}", DateTime.Now.ToString("dd/MM/yyyy hh:mm:ss")));

foreach (DriveInfo drive in DriveInfo.GetDrives())

{

record.Add(String.Format("Название диск: {0}", drive.Name));

record.Add(String.Format("Тип диска: {0}", drive.DriveType));

if(drive.IsReady)

{

record.Add(String.Format("Метка диска - {0}", drive.VolumeLabel));

record.Add(String.Format("Тип файловой системы - {0}", drive.DriveFormat));

record.Add(String.Format("Общее доступное место - {0} мб", GetSizeInMegabytes(drive.TotalSize)));

record.Add(String.Format("Всё свободное место - {0} мб", GetSizeInMegabytes(drive.TotalFreeSpace)));

record.Add(String.Format("Занятое место - {0} мб", GetSizeInMegabytes(drive.TotalSize) - GetSizeInMegabytes(drive.TotalFreeSpace)));

record.Add(String.Format("Свободное место, которое доступно пользователю - {0} мб", GetSizeInMegabytes(drive.AvailableFreeSpace)));

}

}

Logger.addRecord(record);

Thread.Sleep(timer);

}

}

public void stop()

{

isEnabled = false;

}

static long GetSizeInMegabytes(long bytes)

{

return bytes / 1024 / 1024;

}

}

public class Logger

{

private string logFilePath;

public Logger(string logFilePath)

{

this.logFilePath = logFilePath;

}

public void addRecord(StringCollection record)

{

try

{

using (StreamWriter sw = new StreamWriter(logFilePath, true, System.Text.Encoding.Default))

{

foreach(String rec in record)

{

sw.WriteLine(rec);

}

sw.WriteLine("");

}

}

catch (Exception ex)

{

}

}

}

}

Installer1.cs:

using System;

using System.Collections;

using System.Collections.Generic;

using System.ComponentModel;

using System.Configuration.Install;

using System.Linq;

using System.ServiceProcess;

using System.Threading.Tasks;

namespace DiskWatcher\_service

{

[RunInstaller(true)]

public partial class Installer1 : System.Configuration.Install.Installer

{

ServiceInstaller serviceInstaller;

ServiceProcessInstaller processInstaller;

public Installer1()

{

InitializeComponent();

serviceInstaller = new ServiceInstaller();

processInstaller = new ServiceProcessInstaller();

processInstaller.Account = ServiceAccount.LocalSystem;

serviceInstaller.StartType = ServiceStartMode.Manual;

serviceInstaller.ServiceName = "DiskWatcher\_service";

Installers.Add(processInstaller);

Installers.Add(serviceInstaller);

}

}

}