Оглавление

[Введение 3](#_Toc168558939)

[1 Аналитическая часть (проектная часть) 4](#_Toc168558940)

[1.1 Характеристика предметной области 4](#_Toc168558941)

[1.2 Постановка задачи 4](#_Toc168558942)

[1.3 Проектирование Windows-приложения 5](#_Toc168558943)

[2 Практическая часть 8](#_Toc168558944)

[2.1.1 Используемые пространства имен 8](#_Toc168558945)

[2.1.2 Метод GetSystemInfo 9](#_Toc168558945)

[2.1.3 Обработка изменения параметров 10](#_Toc168558945)

[2.1.4 Метод buttonChangeWallpaper\_Click](#_Toc168558945)

# Введение

В современном мире компьютеры и операционные системы играют ключевую роль в повседневной жизни и профессиональной деятельности людей. Административное управление системами и мониторинг их состояния становятся все более важными задачами для обеспечения стабильной и безопасной работы. Разработка приложений, способных автоматически собирать и обрабатывать системную информацию, помогает эффективно управлять ресурсами, контролировать состояние системы и быстро реагировать на возможные проблемы.

Целью данной курсовой работы является разработка приложения, которое будет получать и, при необходимости, устанавливать такие параметры, как имя компьютера, имя пользователя, системные цвета, версии операционной системы, информация об оборудовании, системные метрики, параметры рабочего стола и окон.

# 1 Аналитическая часть (проектная часть)

## 1.1 Характеристика предметной области

## Предметная область данной курсовой работы охватывает аспекты управления и мониторинга компьютерных систем. Это включает в себя получение и установку различных системных параметров, которые важны для поддержания стабильной и эффективной работы компьютеров и операционных систем.

## 1.2 Постановка задачи

**Создать интерфейс пользователя**

* Разработать удобный и интуитивно понятный графический интерфейс для отображения и управления системными параметрами.

**Реализовать функционал получения системной информации**

* Включить возможность получения имени компьютера и имени пользователя.
* Реализовать функционал для получения системных цветов и версии операционной системы.
* Обеспечить получение информации об оборудовании, включая процессор, оперативную память, жесткие диски и видеокарты.
* Включить функционал для сбора системных метрик, таких как загрузка процессора, использование оперативной памяти и активность дисков.
* Реализовать получение параметров рабочего стола и окон, таких как разрешение экрана и расположение окон.

**Обеспечить функционал установки системных параметров**

* Разработать функции для изменения имени компьютера и системных цветов, если это допустимо.

**Разработать механизм обработки ошибок**

* Обеспечить обработку ошибок и исключительных ситуаций для стабильной работы приложения.

**Реализовать систему тестирования**

* Создать и провести модульное тестирование для проверки корректности работы отдельных компонентов.
* Выполнить интеграционное тестирование для проверки взаимодействия всех модулей.
* Провести функциональное тестирование для оценки полноты и качества реализации функций.

**Обеспечить создание документации**

* Подготовить руководство пользователя, описывающее процесс установки и использования приложения.
* Создать техническую документацию, включающую описание архитектуры приложения и используемых технологий.

**Обеспечить безопасность данных**

* Реализовать меры для защиты данных и предотвращения несанкционированного доступа.

## 1.3 Проектирование Windows-приложения

### Функциональные возможности

1. **Добавление нового параметра:**
   * **Имя компьютера:** Пользователь может добавить или изменить имя компьютера, используя соответствующее поле ввода и кнопку "Изменить".
   * **Системные цвета:** Пользователь может выбрать и установить новые системные цвета через интерфейс выбора цветов.
2. **Получение и обновление информации:**
   * **Имя пользователя:** Приложение автоматически отображает имя текущего пользователя системы.
   * **Версия операционной системы:** Отображается текущая версия и сборка установленной ОС.
   * **Информация об оборудовании:** Приложение собирает и отображает данные о процессоре, оперативной памяти, жестких дисках и видеокартах.
   * **Системные метрики:** Отображается текущая загрузка процессора, использование оперативной памяти, активность дисков и сети.
3. **Работа с параметрами рабочего стола и окон:**
   * **Разрешение экрана:** Пользователь может просмотреть текущее разрешение экрана и изменить его при необходимости.
   * **Расположение окон:** Пользователь может управлять расположением окон на рабочем столе, используя интерфейс приложения.
4. **Управление доступом и безопасностью:**
   * **Управление пользователями:** Администратор может добавлять новых пользователей и назначать им роли, используя соответствующие функции в интерфейсе.
   * **Настройка безопасности:** Администратор может устанавливать параметры безопасности для предотвращения несанкционированного доступа.

### Архитектура приложения

При разработке приложения были реализованы различные экраны:

* **Главный экран:** Приложение встречает пользователя рабочим столом с различными окнами действий (Рисунок 1). Здесь отображаются основные параметры системы и предоставляется доступ к основным функциям.
* **Экран управления параметрами:** Пользователь может получать и изменять системные параметры через соответствующие формы и кнопки.
* **Экран управления пользователями:** Администратор может добавлять, редактировать и удалять учетные записи пользователей.
* **Экран безопасности:** Администратор настраивает параметры безопасности и управляет доступом.

### Примеры действий

1. **Изменение имени компьютера:**
   * Администратор вводит новое имя в соответствующее поле и нажимает кнопку "Изменить".
2. **Настройка системных цветов:**
   * Пользователь выбирает новый цвет из палитры и нажимает кнопку "Применить".
3. **Просмотр системных метрик:**
   * Пользователь открывает окно "Системные метрики", где отображаются текущие показатели загрузки процессора, использования оперативной памяти и активности дисков.

В рабочей области программы мы видим информацию о параметрах оборудования нашего компьютера такие как: «Имя компьютера», «Имя пользователя», «Версия ОС», «Процессор», «ОЗУ», «Жесткие диски». (Рисунок 1).

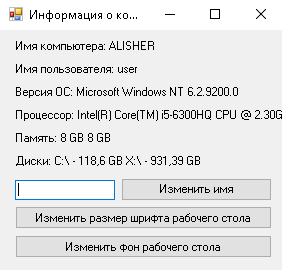


Рисунок 1 — Рабочая область программы.

При добавлении новой имени, надо вводим наше новое имя и нажимаем «Изменить имя», и нас просят перезагрузить компьютер. (Рисунок 2).

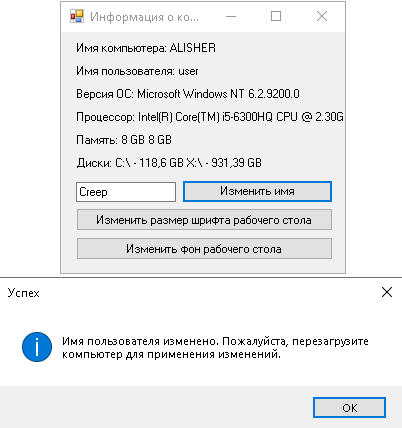


Рисунок 2 — Изменение имени компьютера.

При изменении размера шрифта рабочего стола нажимаем на «Изменить размер шрифта рабочего стола», и нас просят перезагрузить компьютер. (Рисунок 3).

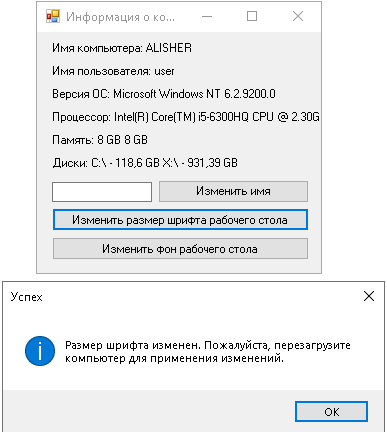


Рисунок 3 — Изменение размера шрифта рабочего стола.

При изменении фона рабочего стола нажимаем «Изменить фон рабочего стола». (Рисунок 4).

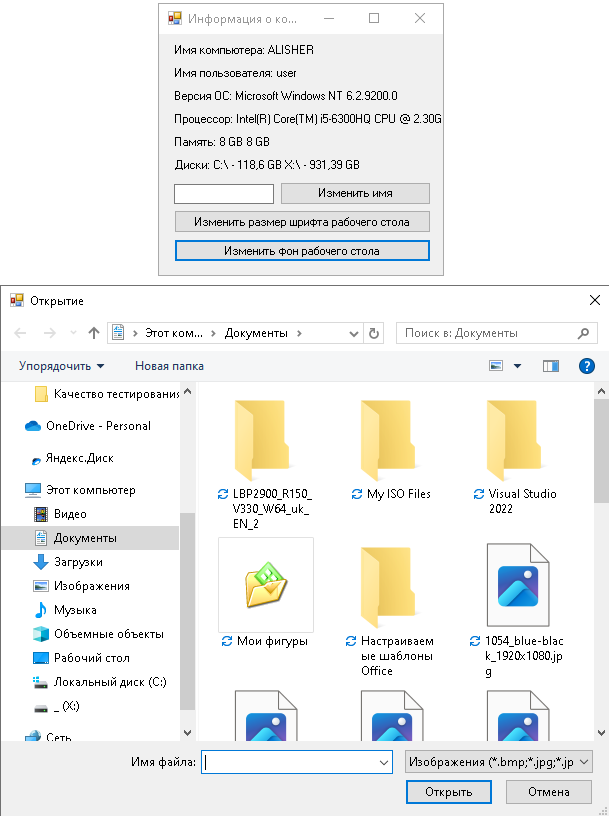


Рисунок 4 — Изменение фона рабочего стола.

# 2 Практическая часть

#### 2.1.1 Используемые пространства имен:

using System;

using System.Management;

using System.Windows.Forms;

using Microsoft.Win32;

using System.Diagnostics;

using System.IO;

using System.Runtime.InteropServices;

* **Объявление классов и переменных**:
* **Form1** наследуется от Form и представляет главную форму приложения.
* Переменные для хранения информации о системе:
  + labelComputerName для отображения имени компьютера.
  + labelUserName для отображения имени пользователя.
  + labelOSVersion для отображения версии операционной системы.
  + labelProcessorInfo для отображения информации о процессоре.
  + labelMemoryInfo для отображения информации об оперативной памяти.
  + labelDiskInfo для отображения информации о дисках.

### Вызывается метод InitializeComponent(), который инициализирует все компоненты формы. Вызывается метод Form1\_Load, который запускает начальную загрузку данных о системе.

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

GetSystemInfo();

}

* **2.1.2 Метод GetSystemInfo**:
* Инициализация переменных для хранения информации о компьютере, пользователе и ОС.
* Вызов методов для получения информации о процессоре, оперативной памяти и дисках.
* Отображение собранной информации на форме.

private void GetSystemInfo()

{

labelComputerName.Text += Environment.MachineName;

labelUserName.Text += Environment.UserName;

labelOSVersion.Text += Environment.OSVersion;

// Processor Info

string cpuInfo = "";

using (ManagementObjectSearcher searcher = new ManagementObjectSearcher("select \* from Win32\_Processor"))

{

foreach (ManagementObject obj in searcher.Get())

{

cpuInfo += obj["Name"] + " ";

}

}

labelProcessorInfo.Text += cpuInfo;

// Memory Info

string memoryInfo = "";

using (ManagementObjectSearcher searcher = new ManagementObjectSearcher("select \* from Win32\_PhysicalMemory"))

{

foreach (ManagementObject obj in searcher.Get())

{

memoryInfo += Math.Round(Convert.ToDouble(obj["Capacity"]) / 1024 / 1024 / 1024, 2) + " GB ";

}

}

labelMemoryInfo.Text += memoryInfo;

// Disk Info

string diskInfo = "";

foreach (DriveInfo drive in DriveInfo.GetDrives())

{

if (drive.IsReady)

{

diskInfo += drive.Name + " - " + Math.Round((double)drive.TotalSize / 1024 / 1024 / 1024, 2) + " GB ";

}

}

labelDiskInfo.Text += diskInfo;

}

**2.1.3 Обработка изменения параметров**

* Получает новое имя пользователя из текстового поля.
* Изменяет значение в реестре, если введено новое имя.
* Обрабатывает возможные ошибки при изменении имени пользователя.

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

{

string newUserName = textBoxNewUserName.Text;

if (!string.IsNullOrEmpty(newUserName))

{

try

{

RegistryKey key = Registry.CurrentUser.OpenSubKey(@"Software\Microsoft\Windows\CurrentVersion\Explorer", true);

key.SetValue("Logon User Name", newUserName);

MessageBox.Show("Имя пользователя изменено. Пожалуйста, перезагрузите компьютер для применения изменений.", "Успех", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

catch (Exception ex)

{

MessageBox.Show("Ошибка при изменении имени пользователя: " + ex.Message, "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

* **Метод buttonChangeFont\_Click:**



 Изменяет размер шрифта через реестр.

 Обрабатывает возможные ошибки при изменении размера шрифта..

**private void buttonChangeFont\_Click(object sender, EventArgs e)**

**{**

**try**

**{**

**RegistryKey key = Registry.CurrentUser.OpenSubKey(@"Control Panel\Desktop", true);**

**key.SetValue("LogPixels", 120);**

**MessageBox.Show("Размер шрифта изменен. Пожалуйста, перезагрузите компьютер для применения изменений.", "Успех", MessageBoxButtons.OK, MessageBoxIcon.Information);**

**}**

**catch (Exception ex)**

**{**

**MessageBox.Show("Ошибка при изменении размера шрифта: " + ex.Message, "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);**

**}**

**}**

**2.1.4 Метод buttonChangeWallpaper\_Click**

 Открывает диалоговое окно для выбора изображения.

 Устанавливает выбранное изображение в качестве обоев рабочего стола.

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

{

OpenFileDialog openFileDialog = new OpenFileDialog();

openFileDialog.Filter = "Изображения|\*.bmp;\*.jpg;\*.jpeg;\*.png";

if (openFileDialog.ShowDialog() == DialogResult.OK)

{

string wallpaperPath = openFileDialog.FileName;

SetWallpaper(wallpaperPath);

}

}

[DllImport("user32.dll", CharSet = CharSet.Auto)]

private static extern int SystemParametersInfo(int uAction, int uParam, string lpvParam, int fuWinIni);

private void SetWallpaper(string path)

{

const int SPI\_SETDESKWALLPAPER = 20;

const int SPIF\_UPDATEINIFILE = 0x01;

const int SPIF\_SENDCHANGE = 0x02;

try

{

SystemParametersInfo(SPI\_SETDESKWALLPAPER, 0, path, SPIF\_UPDATEINIFILE | SPIF\_SENDCHANGE);

MessageBox.Show("Фон рабочего стола изменен.", "Успех", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

catch (Exception ex)

{

MessageBox.Show("Ошибка при изменении фона рабочего стола: " + ex.Message, "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}