**Задание к курсовому проекту.**

Написать программу, реализующую возможность мониторинга программно-аппаратной части персонального компьютера под управлением операционной системы семейства Windows 7. Программа должна позволять просматривать список основных аппаратных узлов компьютера, а так же отображать информацию о них. Кроме того, программа должна отображать информацию о программной составляющей компьютера, её настройках и возможностях. В режиме реального времени программа должна отображать загрузку оперативной памяти, процессора, температуру жёсткого диска, и выводить эти данные в виде графиков. Так же необходимо реализовать тест производительности компьютера, в частности, скорости пересылки данных.

**Введение**

C# — объектно-ориентированный язык программирования. Разработан в 1998—2001 годах группой инженеров в компании Microsoft как основной язык разработки приложений для платформы Microsoft .NET Framework. Компилятор C# входит в стандартную установку .NET Framework. C# разрабатывался как язык программирования прикладного уровня для CLR и, как таковой, зависит, прежде всего, от возможностей самой CLR. Это касается, прежде всего, системы типов C#, которая отражает BCL. Присутствие или отсутствие тех или иных выразительных особенностей языка диктуется тем, может ли конкретная языковая особенность быть транслирована в соответствующие конструкции CLR. Так, с развитием CLR от версии 1.1 к 2.0 значительно обогатился и сам C#; подобного взаимодействия следует ожидать и в дальнейшем. CLR предоставляет C#, как и всем другим .NET-ориентированным языкам, многие возможности, которых лишены «классические» языки программирования. Например, сборка мусора не реализована в самом C#, а производится CLR для программ, написанных на C# точно так же, как это делается для программ на VB.NET, J# и др.

Наиболее быстрым методом сбора информации о программной и аппаратной составляющих компьютера является интерфейс WMI - Windows Management Instrumentation. WMI — это одна из базовых технологий для централизованного управления и слежения за работой различных частей компьютерной инфраструктуры под управлением платформы Windows.

Технология WMI — это расширенная и адаптированная под Windows реализация стандарта WBEM (на англ.), принятого многими компаниями, в основе которого лежит идея создания универсального интерфейса мониторинга и управления различными системами и компонентами распределенной информационной среды предприятия с использованием объектно-ориентированных идеологий и протоколов HTML и XML.

В основе структуры данных в WBEM лежит Common Information Model (CIM), реализующая объектно-ориентированный подход к представлению компонентов системы. CIM является расширяемой моделью, что позволяет программам, системам и драйверам добавлять в неё свои классы, объекты, методы и свойства.

WMI, основанный на CIM, также является открытой унифицированной системой интерфейсов доступа к любым параметрам операционной системы, устройствам и приложениям, которые функционируют в ней.

Важной особенностью WMI является то, что хранящиеся в нём объекты соответствуют динамическим ресурсам, то есть параметры этих ресурсов постоянно меняются, поэтому параметры таких объектов не хранятся постоянно, а создаются по запросу потребителя данных.

Для обращения к объектам WMI используется специфический язык запросов WMI Query Language (WQL), который является одним из разновидностей SQL. Основное его отличие от ANSI SQL — это невозможность изменения данных, то есть с помощью WQL возможна лишь выборка данных с помощью команды SELECT. Помимо ограничений на работу с объектами, WQL не поддерживает такие операторы как DISTINCT, JOIN, ORDER, GROUP, математические функции. Конструкции IS и NOT IS применяются только в сочетании с константой NULL.

Таким образом, при помощи интерфейса WMI можно легко и быстро получать наиболее полные данные о компьютере, независимо от аппаратной реализации. Но существует ряд ограничений на использование данной программы, как то использование её под управлением операционной системы Windows 7 и наличие в системе установленного пакета .Net Framework версии не ниже 3.5 sp1. Так же в связи с постоянно развивающимся уровнем высоких технологий, не всегда удаётся получить полные данные о компьютере, так как в инструментарий WMI ещё не были внесены способы получения информации о самых новых аппаратных частях компьютера.

**Обзор литературы**

Для наиболее полного понимания принципов работы проекта, необходимо точное понимание работы инструментария WMI. Вот как об этом пишет Агуров :

Для обращения к объектам WMI используется специфический язык запросов WMI Query Language (WQL), который является одним из разновидностей SQL. Основное его отличие от ANSI SQL — это невозможность изменения данных, то есть с помощью WQL возможна лишь выборка данных с помощью команды SELECT. Помимо ограничений на работу с объектами, WQL не поддерживает такие операторы как DISTINCT, JOIN, ORDER, GROUP, математические функции. Конструкции IS и NOT IS применяются только в сочетании с константой NULL.

Запросы WQL обычно применяются в скриптах, но их также можно протестировать в программе Wbemtest и в консольной утилите Wmic (утилита wmic не требует написания ключевого слова SELECT и полей выборки)

Так же в книге Гука описываются классы WMI:

Так как WMI построен по объектно-ориентированному принципу, то все данные операционной системы представлены в виде объектов и их свойств и методов.

Все классы группируются в пространства имен, которые иерархически упорядочены и логически связаны друг с другом по определенной технологии или области управления. В WMI имеется одно корневое пространство имен Root, которое в свою очередь имеет 4 подпространства: CIMv2, Default, Secutiry и WMI.

Классы имеют свойства и методы и находятся в иерархической зависимости друг от друга, то есть классы-потомки могут наследовать или переопределять свойства классов-родителей, а также добавлять свои свойства.

Свойства классов используются для однозначной идентификации экземпляра класса и для описания состояния используемого ресурса. Обычно все свойства классов доступны только для чтения, хотя некоторые из них можно модифицировать определенным методом. Методы классов позволяют выполнить действия над управляемым ресурсом.

Экземпляры классов могут генерировать события, к которым можно подписываться. При наступлении события WMI автоматически создает экземпляр того класса, которому соответствует это событие. Такой механизм удобно использовать для выполнения определенной команды при наступлении определенного события, то есть следить за состоянием объектов операционной системы.

То есть можно сделать вывод, что для пользования WMI необходимы базовые знания языка SQL для формирования запросов. Сформировав запрос по необходимому классу, мы можем получить некий набор данных, и после этого найти среди этих данных интересующий нас поля.

**Системное проектирование**

Дынный проект можно условно разбить на четыре основных блока: интерфейсный блок, блок формирования и обработки запросов WMI, блок получения динамических данных, блок отображения информации.



В интерфейсном блоке осуществляется анализ и обработка выбранных пользователем действий, а так же предлагаются сами действия. Основные действия (получение и отображение информации) выбираются в качестве ветви дерева, расположенной в правой части программы. После выборы действия, управление передаётся либо блоку формирования и обработки запросов, либо блоку получения динамических данных. В блоке формирования и обработки запросов осуществляется запрос по заданному ключу, и среди полученных данных, выбираются необходимые. Полученные данные выводятся в таблицу в блоке отображения информации, после чего память очищается сборщиком мусора. В блоке получения динамических данных осуществляется контроль над такими данными, как температура жёсткого диска, загруженность процессора и оперативной памяти. Так же присутствует возможность проверки скорости обмена данными между процессором и памятью. После отображения любой информации управление заново передаётся интерфейсному блоку.

Интерфейс программы разрабатывался при помощи интерфейса программирования приложений (API), отвечающего за графический интерфейс пользователя и являющегося частью Microsoft .NET Framework. Данный интерфейс упрощает доступ к элементам интерфейса Microsoft Windows за счет создания обертки для существующего Win32 API в управляемом коде. Причем управляемый код — классы, реализующие API для Windows Forms, не зависят от языка разработки. С одной стороны Windows Forms рассматривается как замена более старой и сложной библиотеке MFC, изначально написанной на языке C++, но с другой стороны, WF не предлагает парадигму, сравнимую с MVC. Для исправления этой ситуации и реализации данного функционала в WF существуют сторонние библиотеки. Одной из наиболее используемых подобных библиотек является User Interface Process Application Block, выпущенная специальной группой Microsoft, занимающейся шаблонами и практиками, для бесплатного скачивания. Эта библиотека также содержит исходный код и обучающие примеры для ускорения обучения.

Приложение Windows Forms представляет собой событийно-ориентированное приложение, поддерживаемое Microsoft .NET Framework. В отличие от пакетных программ большая часть времени тратится на ожидание от пользователя каких-либо действий, как например, ввод текста в текстовое поле или клика мышкой по кнопке.

В основе проектирования графического интерфейса заложены принципы простоты и достаточности. Это выражается в простоте реализации операций и интуитивного понимания возможностей программы. Так, с правой стороны основного окна программы расположен список в виде дерева, отображающий основные компоненты компьютера. По мере развёртывания узлов дерева, можно выбрать и получить интересующую информацию. Основным элементом главного окна программы является таблица, состоящая из двух колонок. В первой колонке располагаются названия свойств, отображаемых во второй колонке таблицы. При однократном нажатии правой кнопки мыши появляется меню, позволяющее очистить таблицу. Над таблицей расположены графики, отображающие загрузку процессора и оперативной памяти.

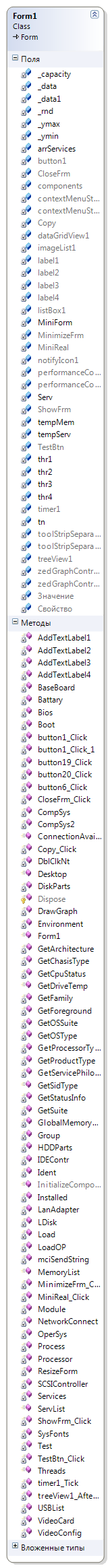
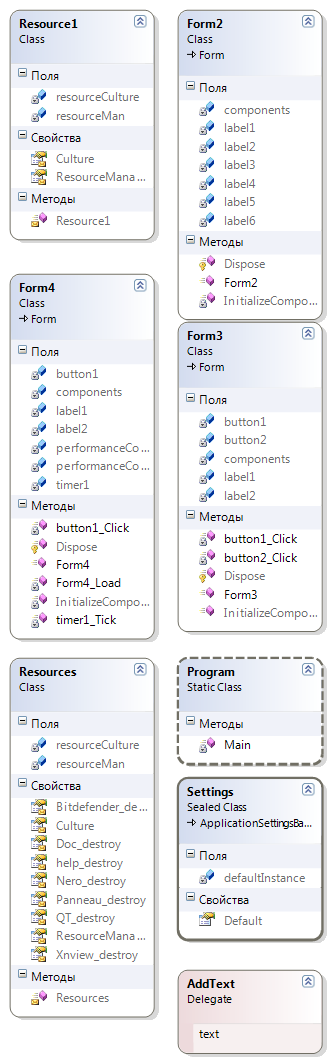
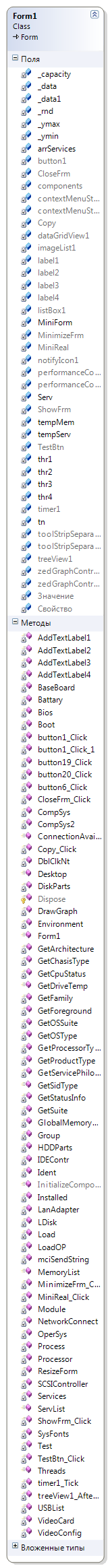
Принцип работы блока формирования и обработки запросов прост. В начале формируется запрос к объектам WMI. Для обращения к объектам WMI используется специфический язык запросов WMI Query Language (WQL), который является одним из разновидностей SQL. Основное его отличие от ANSI SQL — это невозможность изменения данных, то есть с помощью WQL возможна лишь выборка данных с помощью команды SELECT. Помимо ограничений на работу с объектами, WQL не поддерживает такие операторы как DISTINCT, JOIN, ORDER, GROUP, математические функции. Конструкции IS и NOT IS применяются только в сочетании с константой NULL. Далее, из полученной структуры по имени выбираются поля, и записываются в таблицу при помощи блока отображения информации.

Сбор динамической информации происходит при помощи счетчиков производительности (performanceCounters). Счетчик — это механизм, посредством которого собираются данные производительности. В реестре хранятся имена всех счетчиков, каждый из которых относится к определенной области функциональных возможностей системы. Примеры включают время занятости процессора, использование памяти или количество байтов, принятых через сетевое соединение. Информация счетчика должна включать категорию или объект производительности, для которого счетчик измеряет данные. Категории компьютера включают физические компоненты, например процессоры, диски, память. Существуют также системные категории, например, процессы и потоки. Каждая категория относится к некоторому функциональному элементу компьютера и имеет набор стандартных счетчиков, присвоенных ему. Для получения данных производительности для счетчиков, требующих начальное или предыдущее значение для выполнения необходимых вычислений. Получение информации о текущей температуре жёсткого диска основано на WMI запросе к соответствующему классу, и смещении на 194 бита, для получения данных о температуре.

Тест производительности отображает время, за которое компьютер выполняет алгоритм. Суть алгоритма заключается в многократном выполнении операций с длинными числами. На каждом шаге цикла выполняется рекурсивный расчет тригонометрической функции. Параллельно выполняется сортировка одного массива, и перестановка элементов другого массива. Таким образом, выполняется проверка скорости пересылки данных.

**Функциональное проектирование**

Программы написана на языке программирование C#, следовательно, структура разбита на классы. Диаграмма классов представлена на рис 4.1.

****

**Рис 4.1** Диаграмма классов

Класс Form1 представляет собой описание событий и действий, происходящих в главном окне программы. Методы этого класса позволяют получить информацию о соответствующих частях компьютера, либо это вспомогательные методы. Так, например, методы AddTextLabel1(), AddTextLabel2(), AddTextLabel3(), AddTextLabel4() позволяют изменять текст на элементах управления интерфейса из другого потока, используя делегат.

Методы VideoCard(), NetworkConnect(), Ident(), Boot(), CompSys(), CompSys2(), Desktop(), DiskParts(), Environment(), Group(), LDisk(), OperSys(), Module(), HDDParts(), MemoryList(), Bios()Installed(), Services(), ServList(), USBList(), SysFonts(), BaseBoard(), IDEContr(), SCSIController(), VideoConfig(), Battary(), LanAdapter() служат для получения информации о составляющих частях компьютера, и выводят её на экрана. По своей структуре и способу получения ифнормации эти методы схожи. Так, в начале метода формируеться WQL запрос, производящий поиск информации по заданному ключу. В качестве ключа используеться название класса, отвечающего за хранение информации о соответсвующим состоявляющим компьютера. Далее создаётся объект класса, производяшего поиск информации. Найденная информация записывается в коллекцию. Далее, в этой коллекции происходит поиск полей с необходимой информацией. Поиск осуществляется по названию поля. Для каждого поля, найденная информация тут же записывается в таблицу. В связи с тем, что на различных компьютерах установлено различное аппаратное и программное обеспечение, для каждого поля написан свой обработчик исключений, не позволящий программе, в случае ошибки определения того или иного свойства, прекратить корректное выполнение задачи.

Метод DrawGraph() используется для отрисовки графиков загрузки оперативной памяти и процессора. Первоначально, в данном методе происходит созднание панелей, на которых будет происходить отрисовка графиков. Далее, эти подвергаются очистке. Далее, происходит инициализация панелей, настройка типа линий и размеров координатных осей. Затем, данные из массивов добавляются на графики, графики обновляются, и по заданным координатам отображаются графики.

Метод GetDriveTemp() используется для получения информации о текущей температуре жёсткого диска. В ходе выполнения метода, считывается информация при помощт системы S.M.A.R.T. (Self-Monitoring, Analysis and Reporting Technology — технологии оценки состояния жёсткого диска встроенной аппаратурой самодиагностики, а также механизм предсказания времени выхода его из строя). Затем, в считанном блоке данных происходит смещение на 194 бита, где и располагается значение текущей температуры винчестера. Что бы информация постоянно обновлялась, данный метод запускаеться в отдельном потоке, а информация выводиться на экран при помощи делегата.

.

Рис 4.2 блок-схема метода GetDriveTemp()

Глобальные переменные используются в коде для обеспечения большей простоты и надёжности программы. Так, например, очень удобным является объявление объектов потока Thread thr1, thr2, thr3,thr4 глобальными. После инициализации этих объектов, они будут использоваться для управления потоками, выполняющими параллельное считывание температуры винчестера, загрузку оперативной памяти и процессора, а так же рассчёт времени выполнения алгоритма пересылки данных.

TreeNode[] tn и TreeNode[] Serv используются как глобальные массивы элементов дерева, которые заполняются динамически данными о службах компьютера и о носителях информации.

List<double> \_data и List<double> \_data1 используются для хранения информации о текущем и предыдущих значениях загрузки процессора и памяти.

string []tempMem и string[] tempServ хранят в себе списки названий тех сменных носителей и служб, которые будут отображаться в качестве ветвей дерева.

В большинстве методов, получающих информацию о составляющих компьютера, объявлена переменная string ret, которой присвоено значение "Null". Эта переменная используется для возврата строки, содержащей слово “Null” в том случае, если во время выполнения метода возникла некая ощибка, и получить информацию невозможно.

Скрытие и отображение главного окна программы происходит при помощи вызовов методов hide() и show() формы, а также установкой свойств Enabled и Visible в соответствующие значения.