

**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук
Образовательная программа «Программная инженерия»

СОГЛАСОВАНО

Кандидат технических наук,
доцент

_____ В. А. Дударев
«__» _____ 2022 г.

УТВЕРЖДАЮ

Академический руководитель
образовательной программы
«Программная инженерия»
профессор департамента программной
инженерии, канд. техн. наук

_____ В.В. Шилов
«__» _____ 2022 г.

**ПРИЛОЖЕНИЕ ДЛЯ МОНИТОРИНГА
РАБОТЫ ASP.NET ПРИЛОЖЕНИЙ**

Пояснительная записка

ЛИСТ УТВЕРЖДЕНИЯ

RU.17701729.10.03-01 81 01-1-ЛУ

Исполнитель

студент группы БПИ203

_____ / Д. О. Соломонов /

«__» _____ 2022 г.

Подп. и дата	
Инв. № дубл.	
Взам. инв. №	
Подп. и дата	
Инв. № подл	

Москва 2022

УТВЕРЖДЕН
RU.17701729.10.03-01 81 01-1-ЛУ

**ПРИЛОЖЕНИЕ ДЛЯ МОНИТОРИНГА
РАБОТЫ ASP.NET ПРИЛОЖЕНИЙ**

Пояснительная записка

RU.17701729.10.03-01 81 01-1

Листов 25

<i>Инв. № подл</i>		<i>Подп. и дата</i>		<i>Взам. инв. №</i>		<i>Инв. № дубл.</i>		<i>Подп. и дата</i>	
--------------------	--	---------------------	--	---------------------	--	---------------------	--	---------------------	--

Москва 2022

СОДЕРЖАНИЕ

1. ВВЕДЕНИЕ	3
1.1. Наименование программы	3
1.2. Документы, на основании которых ведется разработка	3
2. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ.....	4
2.1. Назначение программы	4
2.2. Область применения программы	4
3. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ.....	5
3.1. Постановка задачи на разработку программы	5
3.2. Описание и обоснование метода входных и выходных данных	5
3.4. Описание выбора технических средств	6
3.5. Описание выбора программных средств	6
3.6. Описание архитектуры и функционирования программы	7
Схема структуры интерфейса программы	Error! Bookmark not defined.
4. ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ	10
4.1. Предполагаемая потребность.....	10
4.2 Экономические преимущества разработки по сравнению с отечественными и зарубежными аналогами	10
5. ИСТОЧНИКИ, ИСПОЛЬЗОВАННЫЕ ПРИ РАЗРАБОТКЕ	11
ПРИЛОЖЕНИЕ 1.....	12
ПРИЛОЖЕНИЕ 2.....	13
ПРИЛОЖЕНИЕ 3.....	16
ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ	25

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

1. ВВЕДЕНИЕ

1.1. Наименование программы

Наименование программы на русском языке: Приложение для мониторинга работы ASP.Net-приложений

Наименование программы на английском языке: Application for ASP.Net web applications monitoring

1.2. Документы, на основании которых ведется разработка

Программа выполнена в рамках задания на курсовую работу в соответствии с учебным планом подготовки бакалавров (НИУ ВШЭ, факультет компьютерных наук) по направлению 09.03.04 «Программная инженерия» и темой курсового проекта, утверждённой академическим руководителем программы.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

2. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ

2.1. Назначение программы

Приложение предназначено для отслеживания состояния работы ASP.Net приложений через счётчики производительности, доступные в ОС Windows, и оповещения пользователя при возникновении нештатных ситуаций.

2.2. Область применения программы

Программа предполагает использование в области мониторинга работы ASP.Net приложений.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

3. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

3.1. Постановка задачи на разработку программы

В процессе выполнения курсовой работы необходимо создать удобное desktop-приложение для мониторинга текущего состояния работы ASP.Net-приложений через доступные в ОС Windows счётчики производительности, и имеющее следующие функции:

- Наличие списка счётчиков производительности, доступных в ОС Windows;
- Наличие меню для переходов между окнами приложения;
- Возможность изменения статуса отслеживания выбранных счётчиков производительности;
- Наличие графика, отображающего изменения последних показаний активных счётчиков за отслеживаемый период;
- Наличие таблицы, отображающее средние, максимальные, и минимальные значения активных счётчиков производительности за отслеживаемый период;
- Возможность создания триггера, который будет приведён в активное состояние в случае, если показания активного счётчика в течение всего периода времени будут превышать критическое значение;
- Возможность выбора активного счётчика, конфигурирование критического значения, периода времени, и условия при создании триггера;
- Возможность объединения триггеров в группы;
- Наличие списка групп триггеров;
- Возможность удаления триггеров из групп и самих групп;
- Возможность получения уведомления на email от группы триггеров, когда все триггеры в группе находятся в активном состоянии;
- Возможность привязки email адреса для получения уведомлений;
- Возможность изменения статуса получения уведомлений;
- Возможность изменения подключенного компьютера.

3.2. Описание и обоснование метода входных и выходных данных

Входные данные обусловлены доступными WMI классами в ОС Windows и действиями пользователя. Из списка доступных счётчиков производительности, которые предварительно получены из WMI, пользователь выбирает нужные ему показатели. Имеется возможность ввода адреса электронной почты для дальнейшего получения уведомлений, возможность ввода имени компьютера. Далее у пользователя есть возможность задать имя создаваемой группе для триггеров. Также программа предоставляет возможность добавления триггеров с выбором необходимого активного свойства для отслеживания, конфигурируемым периодом вне нормы, критическим значением, условием (больше или меньше критического значения).

Выходными данными являются отчёты-уведомления о нештатных ситуациях, приходящие на выбранный email.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

3.4. Описание выбора технических средств

Техническое средство для использования продукта должно быть представлено персональным компьютером, обеспечивающим бесперебойную работу операционной системы и имеющим средства для ввода и отображения информации в программе. Он должен включать процессор с тактовой частотой не менее 1 ГГц, свободную память не менее 512 мб, не менее 2 гб оперативной памяти, клавиатуру, монитор с минимальным разрешением 1280x1024, указательное устройство ввода (мышка или сенсорная панель).

3.5. Описание выбора программных средств

Программа разрабатывалась для операционной системы Windows. Выбор обусловлен особенностями разрабатываемого проекта, а именно необходимый доступ к счётчикам производительности в Windows. Выбор WPF в качестве системы для построения клиентского приложения был обусловлен вхождением системы в платформу .NET, которая в свою очередь связана с ОС Windows.

Во время разработки использовался язык программирования C# версии 9.0, который был разработан как язык для платформы Microsoft .Net. Помимо основных библиотек языка использовалась библиотека LiveCharts для отображения информации в графическом представлении. При написании программного кода использовалась интегрированная среда разработки JetBrains Rider. В качестве системы контроля версий использовалась программа Git.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

3.6. Описание архитектуры и функционирования программы

Расположение графических окон в приложении

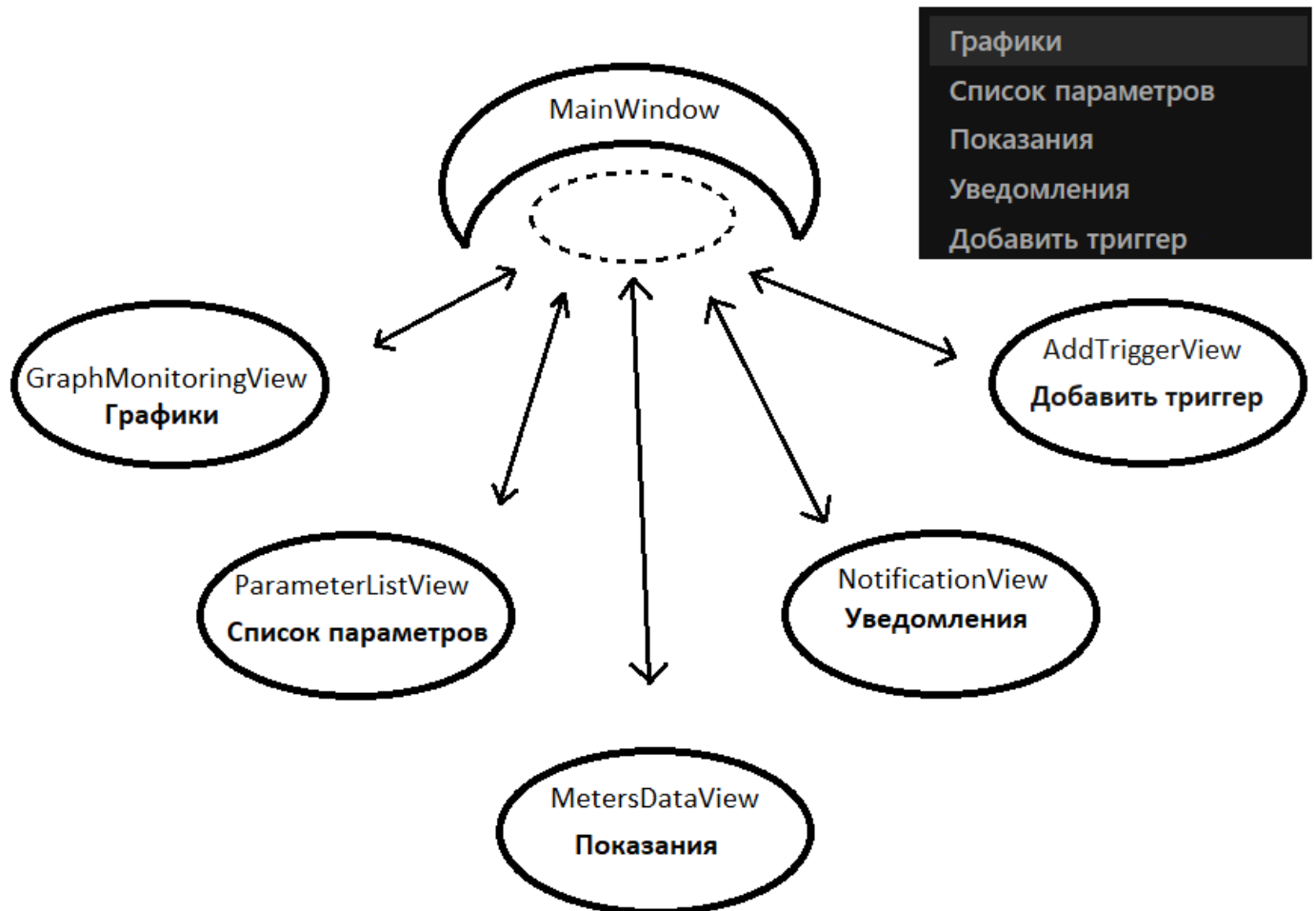


Рис. 1 – Расположение графических окон в приложении.

Общая архитектура приложения задаётся паттерном проектирования Model-View-ViewModel. Бизнес-логика программы определена набором Model классов.

Взаимодействие моделей программы реализовано в таком виде:

- Главным классом для получения данных счётчиков производительности, т. е. для взаимодействия с Windows Management Instrumentation является класс CimConnection.cs. Каждый объект класса CimConnection.cs представляет соединение с локальным или удалённым компьютером, к которому с помощью метода MakeQuery можно делать запросы на языке WQL (WMI Query Language), с помощью метода GetPropertyValue можно получать конкретное числовое значение счётчика.
- Каждый объект класса ObservedDataChanges.cs представляет из себя обновляемый набор показаний выбранного счётчика. Статический таймер каждую секунду

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

обновляет показания всех подписанных на него объектов класса `ObservedDataChanges.cs`, запрашивая через объект класса `CimConnection.cs` значение выбранного счётчика производительности, и сохраняя показания поочерёдно в массиве данных каждого объекта.

- `DataCenter.cs` – класс, содержащий все выбранные счётчики производительности, т.е. все объекты класса `ObservedDataChanges.cs`. Основной класс, к которому осуществляется привязка в `ViewModel` программы для получения данных о счётчиках.
- `ParameterList.cs` – класс с заполняющимся при инициализации списком доступных в компьютере счётчиков производительности. Из доступного списка происходит выбор пользователем интересующих его счётчиков в интерфейсе программы.
- Объекты класса `MonitoringTrigger.cs` являются конфигурируемыми при их создании триггерами, которые становятся активны при соблюдении заданного пользователем условия, т. е. при превышении показаниями выбранного счётчика критического значения на протяжении определённого периода времени. Имеется возможность привязки и отвязки одного объекта класса `ObservedDataChanges.cs` для наблюдения, т. к. триггер не удаляется при изменении статуса отслеживания выбранного свойства.
- Класс `TriggerGroup.cs` содержит в себе список добавленных пользователем объектов класса `MonitoringTrigger.cs` и с помощью события `event EventHandler GroupTriggered` сообщает всем подписчикам о событии, когда все триггеры в группе станут активны одновременно. В классе имеется функционал для добавления и удаления триггеров из списка.
- Класс `EmailNotification.cs` реализует интерфейс `INotificationService.cs`, позволяет привязать email адрес и отправлять сообщения.
- В объекте класса `NotificationModel.cs` содержится список добавленных групп с триггерами, сервис для отправки уведомлений. Объект подписывается на событие добавляемых в список групп триггеров и отправляет сообщение при сигнале от какой-либо из групп.

Воздействие пользователя на изменение состояния моделей предполагается следующим образом:

- Всё взаимодействие пользователя осуществляется благодаря классам, реализующим интерфейс `ICommand` (например, `AddAllPropertiesCommand.cs`, `AddDeletePropertyCommand.cs`, `SetEmailCommand.cs` и т. д.). С помощью инструментов привязки (`Data binding in WPF`) взаимодействие упрощается до нажатия нужных кнопок, каждая из которых привязана к соответствующему классу, реализующему интерфейс `ICommand`.
- В программе реализовано пять основных окон для взаимодействия и отображения информации (см рис.1). Переход между ними доступен в любой момент в меню доступных окон, расположенном слева. Переключение между ними осуществляется с помощью класса `Navigator.cs`, где, благодаря привязке заранее инициализированных кнопок в меню к классу `NavigateCommand.cs`, изменяется отображаемое окно.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

- Каждое окно в приложении представлено отдельным файлом с декларативным языком разметки XAML (View) и соответствующим классом для привязки данных (ViewModel).
- В окне, за которое отвечает GraphMonitoringView, пользователь наблюдает изменение показаний добавленных счётчиков производительности на графике в определённом временном промежутке. В соответствующем классе GraphMonitoringViewModel.cs с помощью инструментов привязки отображаются данные из объекта класса DataCenter.cs.
- В окне ParameterListView с помощью привязки отображается список, доступный в объекте класса ParameterList.cs со всеми имеющимися счётчиками производительности. У каждого счётчика имеется кнопка для изменения его статуса отслеживания, т. е. для добавления (удаления) его в отслеживаемый список объекта класса DataCenter.cs.
- Окно MetersDataView отображает список объекта класса DataCenter.cs, где с каждым счётчиком отображается минимальное, максимальное и среднее значение показаний счётчика за наблюдаемый период. Привязка осуществляется через объект класса MetersDataViewModel.
- NotificationView отвечает за отображение списка групп триггеров из объекта класса NotificationModel.cs. У каждого триггера и группы триггеров имеется кнопка для удаления, привязанные к объектам классов, реализующих интерфейс ICommand DeleteTriggerCommand.cs и DeleteTriggerGroupCommand.cs. Также на элементе расположено текстовое поле для ввода email с кнопкой подтверждения, переключатель для включения\отключения уведомлений, текстовое поле для ввода имени компьютера и кнопка подтверждения.
- Окно AddTriggerView представляет из себя меню для добавки триггеров и группы для триггеров в общий список. У пользователя есть возможность выбрать счётчик для отслеживания из списка активных, критическое значение в диапазоне от 5 до 4294967295, условие (> или <критического значения), необходимый период вне нормы в диапазоне от 5 секунд до 10 минут. Также есть возможность добавить триггер в существующую группу или создать и добавить в новую группу. С помощью класса AddTriggerCommand.cs созданный триггер добавляется в список групп триггеров в объект класса NotificationModel.cs.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

4. ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ

4.1. Предполагаемая потребность

Решение, созданное в рамках курсовой работы, будет полезно в области администрирования и отслеживания состояния ASP.Net приложений, продукт может быть востребован для целей мониторинга работоспособности веб-сервисов, диагностирования и последующего устранения проблем.

4.2 Экономические преимущества разработки по сравнению с отечественными и зарубежными аналогами

Основным аналогом является функция Azure Monitor - Application Insights, она интегрируется в само приложение и имеет больше возможностей для конфигурации. Это платный аналог, который доступен по бесплатной версии подписки с ограничением мощности. Преимуществом разработанной программы является её бесплатное использование и распространение, user-friendly интерфейс.

Программа мониторинга системы Performance Monitor Windows позволяет отслеживать изменение счётчиков производительности и её использование бесплатно. В разработанной программе, помимо этого, имеется возможность оповещения пользователя о возникновении нестандартных ситуаций на почту.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

5. ИСТОЧНИКИ, ИСПОЛЬЗОВАННЫЕ ПРИ РАЗРАБОТКЕ

1. ГОСТ 19.404-79 Пояснительная записка. Требования к содержанию и оформлению. – М.: Стандартинформ, 2010.
2. ГОСТ 19.101-77 Виды программ и программных документов. – М.: Стандартинформ, 2010.
3. Библиотека для работы с визуальным представлением данных в виде графиков URL <https://lvcharts.net/>
4. Информация о инструментарии управления Windows URL <https://docs.microsoft.com/ru-ru/windows/win32/wmisdk/wmi-start-page>
5. Диагностика проблем с производительностью приложений с помощью счётчиков производительности Windows URL <https://docs.microsoft.com/ru-ru/windows-server/remote/remote-desktop-services/rds-rdsh-performance-counters>
6. Мэтью Мак-Дональд. WPF: Windows Presentation Foundation в .NET 4.5 с примерами на C# 5.0 для профессионалов. 4 издание. Изд-во Вильямс, 2013 г. – 1024 с.
7. Подбельский Вадим Валериевич. Язык декларативного программирования XAML. Изд-во ДМК Пресс, 2018 г. – 336 с.
8. Instrument and Monitor Your ASP.NET Apps Using WMI URL <https://docs.microsoft.com/en-us/archive/msdn-magazine/2005/september/instrument-and-monitor-your-asp-net-apps-using-wmi-and-mom-2005>
9. Классы WMI URL <https://docs.microsoft.com/ru-ru/windows/win32/wmisdk/wmi-classes>
10. Счётчики производительности Windows URL <https://docs.microsoft.com/ru-ru/windows/win32/perfctrs/performance-counters-portal>
11. ASP.NET Core Документация для разработки приложений URL <https://docs.microsoft.com/ru-ru/aspnet/core/?view=aspnetcore-6.0>
12. Настройка отправки email через SMTP Gmail сервер URL <https://docs.microsoft.com/en-us/answers/questions/189401/c-send-email-via-smtp.html>
13. Информация о построении WQL (WMI Query Language) запросов URL <https://docs.microsoft.com/ru-ru/windows/win32/wmisdk/wql-sql-for-wmi>
14. Предоставление подключения к локальному или удалённому компьютеру объектом сеанса модели CIM URL https://docs.microsoft.com/ru-ru/powershell/module/microsoft.powershell.core/about/about_cimsession?view=powershell-7.2

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

ИСПОЛЬЗУЕМЫЕ ПОНЯТИЯ И ОПРЕДЕЛЕНИЯ

MVVM (*Model-View-ViewModel*) — шаблон проектирования архитектуры приложения.

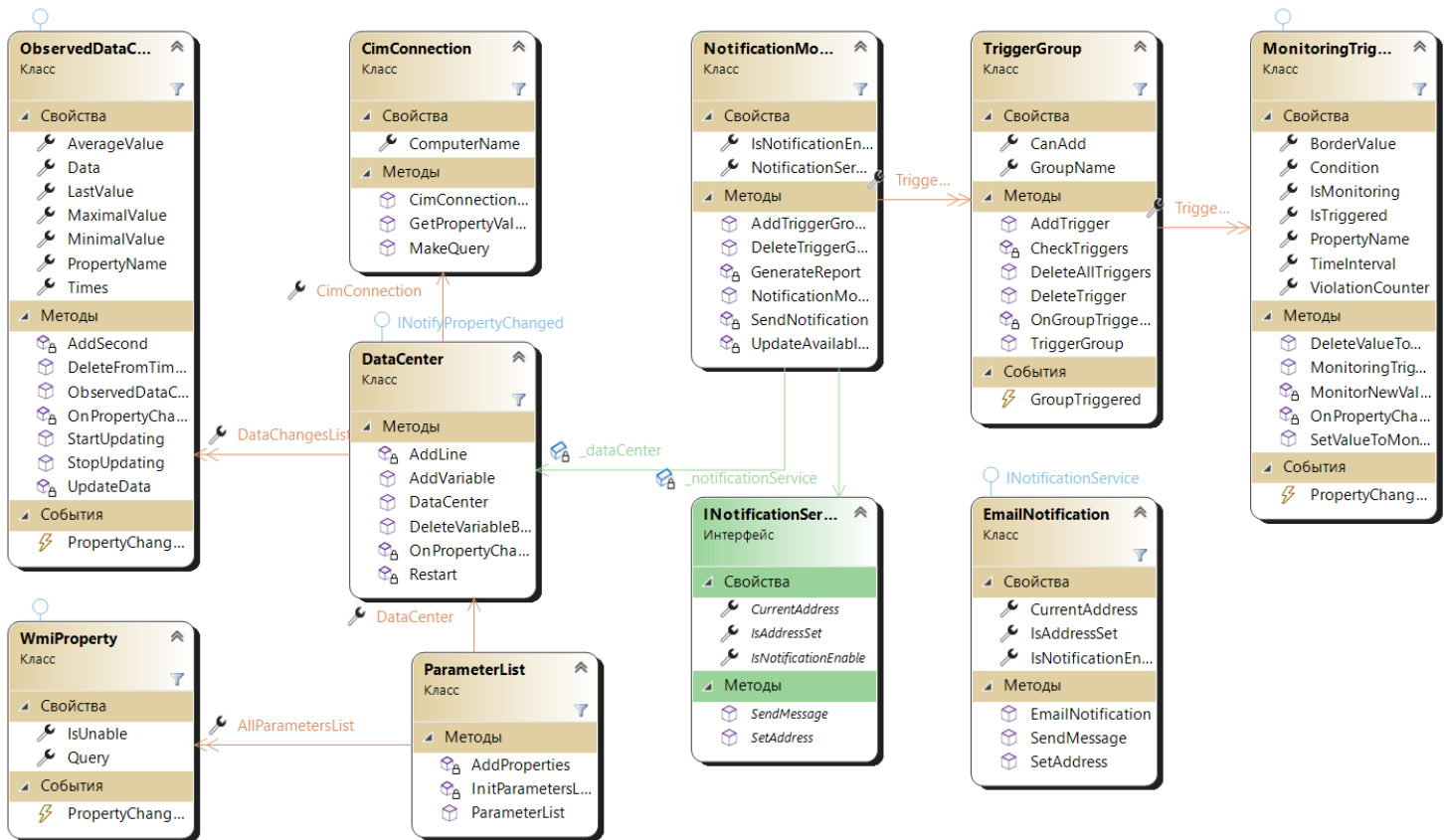
WMI (*Windows Management Instrumentation*)— инструментарий управления Windows, одна из базовых технологий для централизованного управления и слежения за работой различных частей компьютерной инфраструктуры под управлением платформы Windows.

Триггер — это хранимая процедура особого типа, которую пользователь не вызывает непосредственно, а исполнение которой обусловлено действием по модификации данных.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Таблицы с описанием классов и интерфейсов

Таблица 1. Схема зависимости и взаимодействия классов моделей проекта ASP.NetMonitoring



Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Таблица 2. Назначение классов моделей проекта ASP.NetMonitoring

Класс	Назначение
CimConnection.cs	Класс для создания соединения с компьютером.
DataCenter.cs	Класс, содержащий все активные счётчики производительности.
EmailNotification.cs	Класс для отправки сообщений на заданный email адрес.
INotificationService.cs	Интерфейс для отправки уведомлений по адресу.
MonitoringTrigger.cs	Класс, каждый объект которого представляет триггер с заданными условиями.
NotificationModel.cs	Класс для хранения всех созданных групп триггеров и отправки оповещений, при сигнале от какого-либо триггера.
ObservedDataChanges.cs	Класс, который представляет обновляемые изменения счётчика производительности с течением времени.
ParameterList.cs	Класс, хранящий список всех доступных счётчиков.
TriggerGroup.cs	Класс для хранения триггеров и отслеживания их сигналов.
WmiProperty.cs	Класс, который хранит специальный запрос

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Таблица 3. Назначение классов команд проекта ASP.NetMonitoring

Класс команды	Назначение
AddAllPropertiesCommand.cs	Команда для добавления всех выбранных счётчиков в список для отслеживания.
AddDeletePropertyCommand.cs	Команда добавляет (удаляет) выбранный счётчик в список для отслеживания.
AddTriggerCommand.cs	Команда для добавления триггера в группу.
ChangeConnectionCommand.cs	Команда для изменения соединения с заданным компьютером.
ChangeNotificationStatusCommand.cs	Команда для изменения включения\отключения уведомлений.
DeleteAllPropertiesCommand.cs	Команда для удаления всех выбранных счётчиков из списка для отслеживания.
DeleteTriggerCommand.cs	Команда для удаления триггера из группы.
DeleteTriggerGroupCommand.cs	Команда для удаления группы триггеров.
NavigateCommand.cs	Команда для изменения текущего отображаемого окна.
SetEmailCommand.cs	Команда на привязки email адреса.
SwitchMoreOrLessCommand.cs	Команда для смены условия срабатывания триггера (больше или меньше критического значения).

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

ПРИЛОЖЕНИЕ 3

Таблицы с методами классов моделей проекта ASP.NetMonitoring с описанием

Таблица 4. Методы класса CimConnection.cs с описанием

Методы				
Имя	Мод. Доступа	Тип	Аргументы	Назначение
MakeQuery	public	IEnumerable<CimInstance>	string	Метод делает запрос для получения нужного объекта WMI.
GetPropertyValue	public	double	string, string	Метод получает текущее значение счётчика производительности из свойства класса WMI.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Таблица 5. Методы класса DataCenter.cs с описанием

Методы				
Имя	Мод. Доступа	Тип	Аргументы	Назначение
OnPropertyChanged	public	void	string	Метод уведомляет о изменении переданного свойства в объекте.
AddVariable	public	void	string	Метод создаёт и добавляет счётчик в список активных счётчиков.
DeleteVariableByName	public	void	string	Метод удаляет счётчик из отслеживаемого списка, если он был добавлен заранее.
AddLine	private	void	ObservedDataChanges	Метод добавляет в список графиков график, соответствующий переданному счётчику.
Restart	private	void		Метод очищает список всех активных счётчиков и графиков.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Таблица 6. Методы класса EmailNotification.cs с описанием

Методы				
Имя	Мод. Доступа	Тип	Аргументы	Назначение
SendMessage	public	void	string	Метод отправляет сгенерированный репорт о появившейся нештатной ситуации на почту.
SetAddress	public	bool	string	Метод устанавливает почту для отправки уведомлений.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Таблица 7. Методы интерфейса INotificationService.cs с описанием

Методы				
Имя	Мод. Доступа	Тип	Аргументы	Назначение
SendMessage	public	void	string	Метод отправляет сообщение на текущий адресс.
SetAddress	public	bool	string	Метод устанавливает почту для отправки уведомлений.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Таблица 8. Методы класса MonitoringTrigger.cs с описанием

Методы				
Имя	Мод. Доступа	Тип	Аргументы	Назначение
SetValueToMonitoring	public	void	ObservedDataChanges	Метод устанавливает объект счётчика производительности для отслеживания изменений.
DeleteValueToMonitoring	public	bool	String	Метод удаляет объект счётчика производительности для отслеживания изменений.
MonitorNewValue	private	void	object, PropertyChangedEventArgs	Метод обрабатывает каждое изменение свойства объекта счётчика.
OnPropertyChanged	public	void	string	Метод уведомляет о изменении переданного свойства в объекте.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Таблица 9. Методы класса NotificationModel.cs с описанием

Методы				
Имя	Мод. Доступа	Тип	Аргументы	Назначение
GenerateReport	private	String	TriggerGroup	Метод генерирует отчёт о нештатной ситуации в группе триггеров.
UpdateAvailableProperties	private	void	object, NotifyCollectionChangedEventArgs	Метод обновляет статусы отслеживания триггеров за счётчиками при добавлении\удалении счётчиков в отслеживающий список.
SendNotification	private	void	object, EventArgs	Метод отправляет уведомление.
DeleteTriggerGroup	public	void	TriggerGroup	Метод удаляет группу триггеров из общего списка.
AddTriggerGroup	public	void	TriggerGroup	Метод добавляет группу триггеров в общий список.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Таблица 10. Методы класса ObservedDataChanges.cs с описанием

Методы				
Имя	Мод. Доступа	Тип	Аргументы	Назначение
UpdateData	private	void	object, ElapsedEventArgs	Метод запрашивает значение счётчика и добавляет полученное значение в список. Также обновляет максимальное, минимальное и среднее значение.
AddSecond	private	void	object, ElapsedEventArgs	Метод добавляет новую секунду в список со временем.
DeleteFromTimerList	Public	void	ObservedDataChanges	Метод удаляет свойство из статического таймера для обновления.
StartUpdating	public	void		Метод останавливает обновление всех объектов класса.
StopUpdating	public	void		Метод возобновляет обновление всех объектов класса.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Таблица 11. Методы класса ParameterList.CS с описанием

Методы				
Имя	Мод. Доступа	Тип	Аргументы	Назначение
AddProperties	private	void	string	Метод добавляет все свойства WMI класса, т. е. все счётчики.
InitParameterList	private	bool	String	Метод инициализирует доступный список счётчиков свойствами WMI классов.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Таблица 12. Методы класса TriggerGroup.cs с описанием

Методы				
Имя	Мод. Доступа	Тип	Аргументы	Назначение
DeleteAllTriggers	public	void		Метод удаляет все триггеры из группы.
DeleteTrigger	public	void	MonitoringTrigger	Метод удаляет переданный триггер из группы.
CheckTriggers	private	void	object, PropertyChangedEventArgs	Метод проверяет активное состояние всех триггеров в группе и вызывает событие срабатывания группы триггеров.
AddTrigger	public	void	MonitoringTrigger	Метод добавляет триггер в группу.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

[illegible]

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.02.07-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата