СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 4](#_Toc435604603)

[1 АНАЛИЗ ИСХОДНЫХ ДАННЫХ 5](#_Toc435604604)

[1.1 Анализ предметной области 5](#_Toc435604605)

[1.2 Выбор специального и общесистемного ПО 5](#_Toc435604605)

[2ПРОГРАММНОЕ ПРОЕКТИРОВАНИЕ](#_Toc435604604) 7

[2.1 Пользовательский интерфейссервера](#_Toc435604605) 7

[2.2 Пользовательский интерфейсклиента](#_Toc435604605) 7

[2.3 Функциональное обеспечение](#_Toc435604605) 8

[3 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ](#_Toc435604604) 9

[3.1 Детальная реализация функциональных частей ПО](#_Toc435604605) 9

[3.2 Сопроводительная документация ПО](#_Toc435604606) 10

[3.3 Анализ ПО](#_Toc435604607) 10

[3.4 Тестирование ПО](#_Toc435604608) 12

[ЗАКЛЮЧЕНИЕ](#_Toc435604604) 13

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ](#_Toc435604604) 14

[ПРИЛОЖЕНИЕ А](#_Toc435604612) 15

[ПРИЛОЖЕНИЕ Б](#_Toc435604612) 16

[ПРИЛОЖЕНИЕ В](#_Toc435604612) 17

# ВВЕДЕНИЕ

# При разработке программ, мы не только изучаем новые языки программирования, а также их синтаксис. Мы также изучаем историю их появления и развития: изучаем типы данных, учимся создавать приложения, игры, базы данных.

# Современные языки программирования предоставляют широкие возможности в программировании. Существует множество видов: процедурные, структурные, логические, объектно-ориентированные, функциональные и т.д.

# В данном курсовом проекте реализовано приложение, которое осуществляет сбор информации о ПК.К этой информации относится: информация о «железе» ПК, о сетевых адаптерах, об установленном ПО, запущенных процессов и служб.Сбор информации обычно используются в местах, где находится много компьютеров(клиентов) и среди них имеется главный(сервер).С помощью данного приложения мы можем получить нужную нам информацию о компьютере, с возможностью сохранения некоторой информации. Для сбора информации на клиент-компьютере запускается клиент-приложение, которое подключается к серверу.

# Существует широкий спектр доступных аналогов данного приложения, в том числе программы с открытым исходным кодом. По сравнению с другими программами у данного приложения есть некоторые преимущества, а именно: простой и понятный пользователю интерфейс, минимальная нагрузка на процессор, жесткий диск и оперативную память, а также удобство просмотра полученном информации.

# 

# АНАЛИЗ ИСХОДНЫХ ДАННЫХ

**1.1 Анализ предметной области**

Предметной областью проектируемого приложения является сбор информации о ПК в операционной системе Windows.Программа-сервер делает запрос на ту или иную информации программе-клиенту, а та выдаёт ответ в виде запрашиваемой информации. Также имеются некоторые функции по частичному управлению клиентом (блокировка, разблокировка, отключение).

В разрабатываемой программе основным видом формализованных данных является сама информация о ПК.

Обязательно необходимо реализовать три основных действия для работы с данными данной программы:

* Запрос информации у клиента;
* Обработка полученной информации и вывод пользователю;
* Частичное управление клиентом;

## Выбор специального и общесистемного программного обеспечения

# При написании данного курсового проекта использовался язык программирования «Cи#». «C#»(произносится сишарп) —объектно-ориентированныйязык программирования. C# относится к семье языков с C-подобным синтаксисом, из них его синтаксис наиболее близок к C++ и [Java](https://ru.wikipedia.org/wiki/Java). Язык имеет статическую типизацию, поддерживает полиморфизм, перегрузку операторов (в том числе операторов явного и неявного приведения типа), делегаты, атрибуты, события, свойства, обобщённые типы и методы, итераторы, анонимные функции с поддержкой замыканий, LINQ, исключения, комментарии в формате XML.

# «Cи#»широко используется для разработки программного обеспечения, являясьодним из самых популярных языков программирования. Область его применения включает разработку системного программного обеспечения, разработку прикладного программного обеспечения,, приложений для встраиваемых систем, высокопроизводительных серверов, а также развлекательных приложений (игр).

# Также для написания курсового проекта были использованы следующие сторонние программные продукты:

# Visual Studio 2015 Enterprise Edition [1] – написаниеприложения.VisualStudio2015 содержит интегрированную среду разработки программного обеспечения и ряд других инструментальных средств для повышения производительности разработчика, дополнительные инструменты с поддержкой разработки приложений для операционных систем Windows, веб-разработки, а также улучшения средств отладки и оптимизации исполняемого и управляемого кода.

# MicrosoftOfficeVisio 2013 [2] – проектирование программы и составление диаграмм.Данный продукт позиционируется как редактор диаграмм и блок-схемдля Windows.

# Платформа, на которой реализовано данное приложение – Windows8.

# ПРОГРАММНОЕ ПРОЕКТИРОВАНИЕ

## Пользовательский интерфейссервера

При запуске приложения-серверана экран выводится основное окно, макет которого показан на рисунке 2.1.

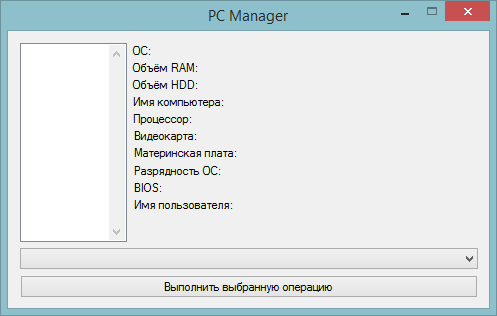


Рисунок 2.1 – Макет основного окна приложения-сервера

## 2.2 Пользовательский интерфейсклиента

При запуске приложения-клиентана экран выводится основное окно, макет которого показан на рисунке 2.2.

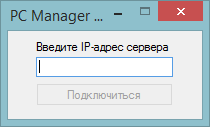


Рисунок 2.2 – Макет основного окна приложения-клиента

Интерфейс приложения реализован с помощью редактора ресурсов VisualStudio 2015.В интерфейсе использованы элементы: EditBox,ComboBox, Button, Static.

## Функциональное обеспечение

Диаграмма вариантов использования, как правило, является исходной моделью, с которой начинается процесс моделирования на языке UML. Она описывает функциональное назначение системы в самом общем виде с точки зрения всех ее пользователей и заинтересованных лиц.

Диаграмма вариантов использования представляет собой диаграмму, на которой изображаются варианты использования проектируемой системы, как правило, заключенные в границу субъекта и внешние актеры, а также определенные взаимоотношения между актерами и вариантами использования.

Основными элементами (предметами языка UML), отображаемыми на диаграмме

вариантов использования, являются: вариант использования, актер, системная граница, примечание.

Актёром в данном случае является пользователь. Актёр на диаграмме использования обозначается фигуркой «человечка».

Отдельный вариант использования обозначается на диаграмме эллипсом, внутри которого содержится его краткое название.

Отношение ассоциации на диаграмме вариантов использования предназначено только для обозначения взаимодействия актера и варианта использования.

Между вариантами использования определены отношение обобщения и две разновидности отношения зависимости – включения и расширения. Отношение включения означает, что базовый элемент явно включает поведение другого элемента. Отношение расширения означает, что базовый элемент неявно включает поведение другого элемента; это отношение применяется для моделирования выбираемого поведения программы[4].

При запуске приложения-серверапользователю требуется:

* Подождать подключения клиентов;
* Выбрать из списка нужного клиента;
* Выбрать действие из возможных;
* Нажать кнопку «Выполнить выбранную операцию»;

При запуске приложения-клиента пользователю требуется:

* Ввести IP-адрес сервера;
* Нажать кнопку «Подключиться (Отключиться)»;

Общая схема действия сервера и клиента представлена в виде диаграммы вариантов использования, представленной на диаграмме А(сервер), диаграмме Б(клиент).

# 3 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ

* 1. **Детальная реализация функциональных частей ПО**

В листинге 1 представлен цикл*,* который определяет архитектуру системы (x64 илиx86), а так же отправляет её серверу.

**Листинг 1** – Определения архитектуры системы

1. if (Environment.Is64BitOperatingSystem)
2. {
3. message += "64-битная ОС/";
4. else message += "32-битная ОС/";
5. query = new ManagementObjectSearcher("root\\CIMV2", "SELECT \* FROM Win32\_BIOS");
6. foreach (ManagementObjectqueryObj in query.Get())
7. {
8. message += queryObj["Manufacturer"] + " " + queryObj["Version"] + "/";
9. break; }
10. message += Environment.UserName;
11. for (int i = 0; i < 2; i++)
12. socket.Send(Encoding.Default.GetBytes(message));
13. break;
14. }

В листинге 2 представлено блокировка управления клиентской программой, а так же отключение диспетчера задач.

**Листинг 2** – Блокировка клиента

1. BeginInvoke(new MethodInvoker(delegate
2. {
3. this.Enabled = false;
4. RegistryKey key = Registry.CurrentUser.CreateSubKey(@"Software\Microsoft\Windows\CurrentVersion\Policies\System");
5. key.SetValue("DisableTaskMgr", "1");
6. foreach (Process p in Process.GetProcesses())
7. if (p.ProcessName == "Taskmgr")
8. { p.Kill(); break; }
9. key.Close();
10. }));

Результат блокировки показан на рисунке 3.1 и рисунке 3.2.

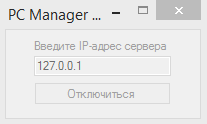


Рисунок 3.1 – Блокировка управления клиентской программой

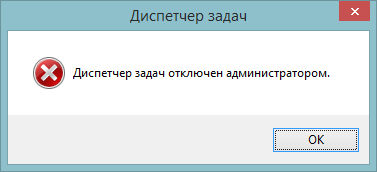


Рисунок 3.2 – Отключение диспетчера задач

В листинге 3 представлена функция IsRunAsAdmin(), выполняющая проверку наличия прав администратора у клиентской программы.

**Листинг 3** – Функция IsRunAsAdmin().

1. internal bool IsRunAsAdmin()
2. {
3. WindowsIdentity id = WindowsIdentity.GetCurrent();
4. WindowsPrincipal principal = new WindowsPrincipal(id);
5. return principal.IsInRole(WindowsBuiltInRole.Administrator);
6. }

В листинге 4 представлено выполнение ключевой команды переданной от сервера к клиенту.

**Листинг 4** – Выполнение ключевой команды

1. if (recv.Contains("keyword"))
2. {
3. string[] pair = recv.Split('$');
4. string command = FindCommand(pair[1]);
5. if (command == "shutdown") Process.Start("C:\\Windows\\System32\\shutdown.exe", "-s -t 1");
6. if (command != "not found")
7. { try { Process.Start(command); }
8. catch(Exception) {} } continue; }

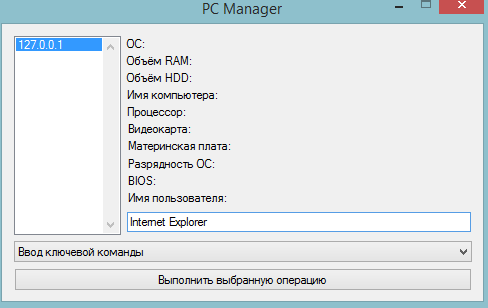


Рисунок 3.3 – Открытие браузера Internet Explorer

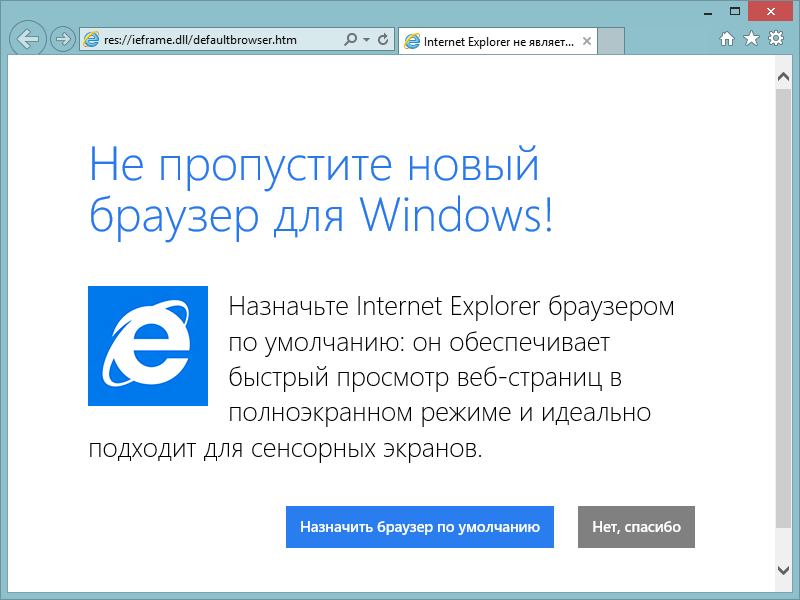
**

Рисунок 3.4 – Internet Explorer

В листинге 5 представлена функция DeleteSocketFormListBox,осуществляющая удаление сокета из общего массива вместе с IP-адресом.

**Листинг 5** – Функция DeleteSocketFormListBox

1. private void DeleteSocketFormListBox(Socket socket)
2. {
3. for(int i = 0; i < listBox1.Items.Count; i++)
4. if (listBox1.Items[i].ToString() == GetSocketRemoteIp(socket))
5. {
6. BeginInvoke(new MethodInvoker(delegate
7. {
8. listBox1.Items.RemoveAt(i);
9. }));
10. return;
11. }
12. }
    1. **Сопроводительная документация программного обеспечения**

Правила использования приложений просты: пользователю-клиенту требуется запустить приложение-клиент, ввести ip-адрес сервера и нажать кнопку «Подключиться», а пользователю-серверу нужно выбрать клиента и производить нужные ему операции.

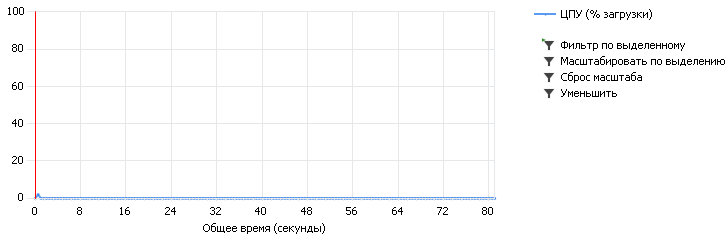
Данные приложения не требуют установки, их необходимо только запустить.

Для запуска приложений необходимо только дважды кликнуть по ярлыку и немного подождать.

В качестве сопроводительной документации программного обеспечения представлено руководство пользователя в приложении В.

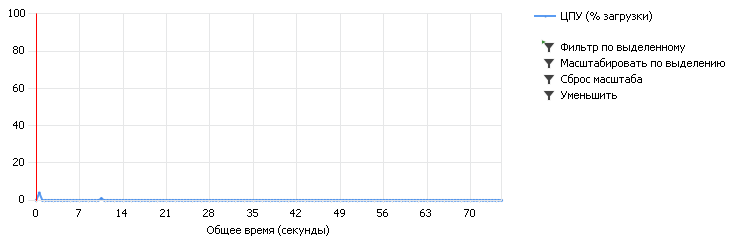
* 1. **Анализ программного обеспечения**

Анализ производительности данного приложения был проведен с помощью встроенного профилировщикаVisualStudio. На клиенте было произведено подключение к серверу и был сделан запрос от сервера на получение списка установленного ПО.Отчёт профилировщика показан в виде графика (рисунок 3.1). К серверу был подключён клиент и был сделан запрос клиенту на список установленного на нём ПО. Отчёт профилировщика показан в виде графика (рисунок 3.2).

****

****

Рисунок3.1–Отчёт производительности на клиенте



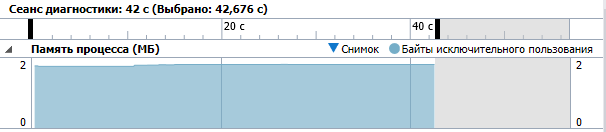


Рисунок 3.2– Отчёт производительности на сервере

Проведя анализ данного отчета можно сказать, что максимальная нагрузка на процессор составляла 5% от его максимальной частоты.Отличие начального и конечного объема ОЗУ, занятого приложением – 0,1 мегабайт. Размер ОЗУ увеличивается т.к. при получениисписка ПО, в оперативной памяти выделяется буфердля данных. Данные показатели не являются критичными для изменения кода программы.

* 1. **Тестирование программного обеспечения**

Модульное тестирование проводилось по мере выполнения данного проекта. Тестирование проходило в основном на граничных значениях, а также учитывались вероятные ошибки пользователя.

Все основные функции приложения были протестированы. Результаты тестирования программного обеспечения представлены в таблице 5.1.

Из-за неправильного формата записи в файл данных о ПК, было некорректное отображение информации в файле. Был изменён формат записи.

**Таблица 5.1**– Результаты тестирования программы

| **Тест** | **Реакция программы** | **Статус** |
| --- | --- | --- |
| Запуск приложения на компьютерах с различными ОС семейства Windows | Программа запущена успешно. | Выполнено |
| Запуск клиентского приложения без прав администратора | Приложение автоматически открывает нового клиента с правами администратора, исходный клиент закрывается. | Выполнено |
| Подключение клиента к серверу | Ввод IP-адреса и корректное подключение к серверу | Выполнено |
| Запрос данных о ПК | Принятие данных от выбранного клиента и отображение их на экране в элементах Static. | Выполнено |
| Принудительное отключение клиента | Выбранный IP-адрес должен отключиться от программы, а на клиенте появляется сообщение, что его отключили принудительно. | Выполнено |
| Получение списка запущенных процессов | В новом окне откроются список запущенных процессов. | Выполнено |
| Получение списка служб | В новом окне откроются список служб. | Выполнено |
| Получение списка сетевых адаптеров | В новом окне откроется список служб. | Выполнено |
| Ввод ключевой команды | Выполнение введённой команды, в случае ввода несуществующей команды выводится предупреждение корректности ввода. | Выполнено |
| Команда открытия изображения | Изображение открывается во весь экран автоматически программой открытия изображений, установленной по умолчанию. | Выполнено |
| Команда открытия аудио видео файла | Аудио или видео файл открывается во весь экран автоматически программой открытия аудио видео файлов, установленной по умолчанию. | Выполнено |
| Команда открытия приложения формата .exe а так же их ярлыков формата .lnk | Запуск приложения. | Выполнено |
| Блокировка клиента | Управление клиентской программой недоступно, диспетчер задач отключён. | Выполнено |
| Разблокировка клиента | Управление клиентской программой стало доступно, диспетчер задач включён. | Выполнено |
| Повторная разблокировка | Выводится сообщение о недоступности этой команды, так как клиент не заблокирован. | Выполнено |
| Блокировка и разблокировка клиента при утере клиентской программы прав администратора | Выводится сообщение о невозможности блокировки клиента. | Выполнено |
| Отключение клиента от сервера | Выбранный IP-адрес должен отключиться от программы, а на клиенте появляется сообщение, что его отключили. | Выполнено |
| Разрыв сети | Сервер удаляет IP-адрес сети, клиент должен отключиться от программы-сервера, а на клиенте появляется сообщение о временной блокировки. Клиент пытается подключится к серверу и при подключении вернёт управление. | Выполнено |
| Подключение нового клиента | Клиент подключён к серверу и отображается в виде IP-адреса клиента. | Выполнено |

# ЗАКЛЮЧЕНИЕ

В рамках курсовой работы были решены поставленные задачи. Спроектировано и разработано приложение для сбора информации о ПК, при помощи языка «Си#» с использованием редактора ресурсов. Приложение корректно работает, отвечает требованиям, поставленным в задачах курсовой работы.

Удалось реализовать программу, которая способна запрашивать у ПК нужные ей сведения, такие как список установленного ПО, запущенных процессов, информацию о «железе» ПК и т.д.

Интерфейс и алгоритм программы интуитивно понятны любому пользователю, который будет использовать данный проект, даже без специальных знаний. Всё это позволит работать с данной программой большому количеству людей и использовать ее в различных организациях без предварительного обучения методике работы с нею. Для использования программы будет достаточным наличие базовых навыков работы с персональным компьютером.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Сайт программы EnterpriseArchitect [Электронный ресурс]. – Режим доступа:

# <http://www.sparxsystems.com.au/>.– Дата доступа: 03.12.2016

1. Центры разработчика Microsoft [Электронный ресурс]. – Режим доступа:

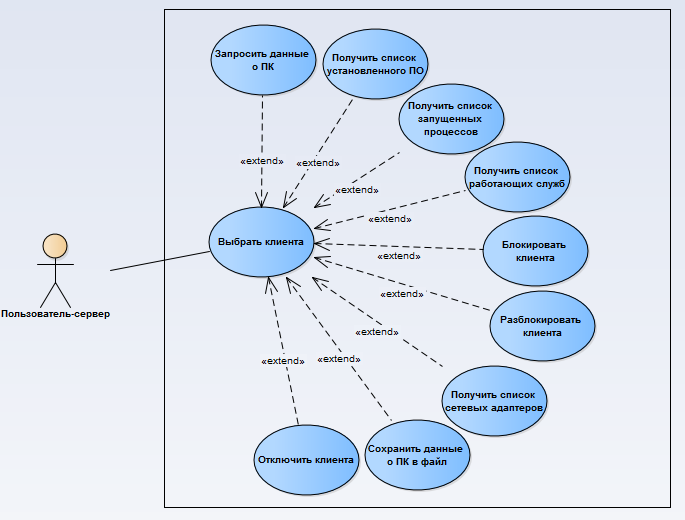
# <https://msdn.microsoft.com/developer-centers-msdn>. – Дата доступа: 02.10.2016-02.12.2016

1. Основы языка C# [Электронный ресурс]. – Режим доступа: https://msdn.microsoft.com/ru-ru/library/zkxk2fwf%28v=vs.90%29.aspx/. – Дата доступа: 02.10.2016-02.12.2016
2. Сайт «Мастерская DimDim». – Режим доступа <http://www.info-system.ru/designing/methodology/uml/theory/use_case_diagram_theory.html> - Дата доступа: 25.11.2016

**ПРИЛОЖЕНИЕ А**

(обязательное)

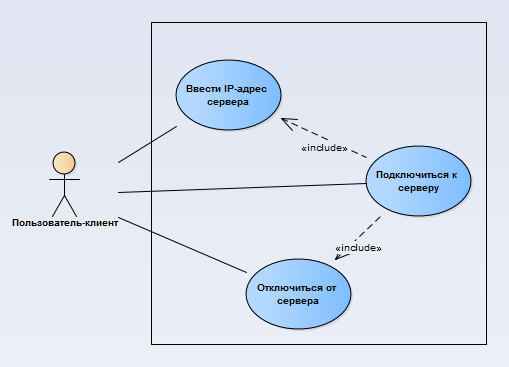
**Диаграмма вариантов использования сервера**

****

**ПРИЛОЖЕНИЕ Б**

(обязательное)

**Диаграмма вариантов использования клиента**

****

**ПРИЛОЖЕНИЕ В**

**Руководство пользователя**

**В1 НАЗНАЧЕНИЕ ПРОГРАММЫ**

# Приложение «PCManager» представляет собой программное средство, которое предоставляет возможностьполучать информацию о ПК.

**В2 УСЛОВИЯ ВЫПОЛНЕНИЯ ПРОГРАММЫ**

Для работы программы требуются:

* Версия WindowsXPи выше.
* На жёстком диске: сервер – 332 Кб, клиент – 309Кб.
* 7.6Мб оперативной памяти.
* Набор библиотек VisualC++ 2015 (VCRedist).

**В3 ВЫПОЛНЕНИЕ ПРОГРАММЫСЕРВЕРА**

После запуска приложения-сервера, на экране появится главное окно программы (рисунок В3.1).

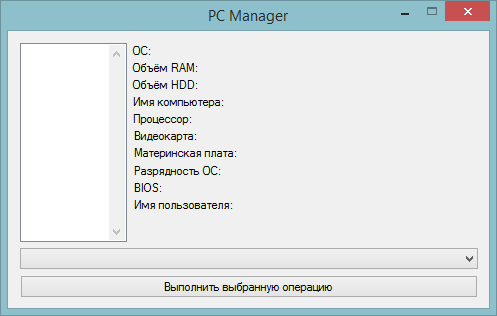


Рисунок В3.1 – Главное окно программы-сервера

Далее ожидаем подключение клиента. После подключения клиента, выбираем его IP-адрес в списке клиентов, затем выбираем операцию, которую будем производить с клиентом (рисунок В3.2).

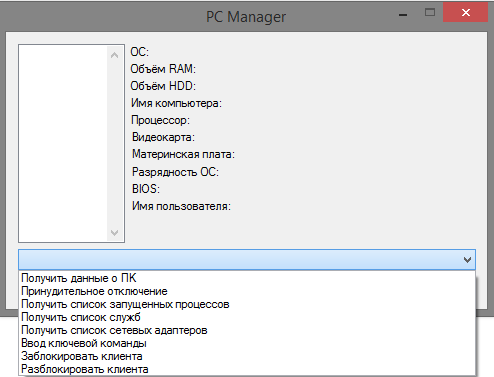


Рисунок В3.2 – Выбор операции над клиентом

Выберем «Получить данные о ПК». Нажимаем «Выполнить выбранную операцию». После получения информации она отобразится в Static-ах рядом с ListBox-ом (рисунок В3.3).

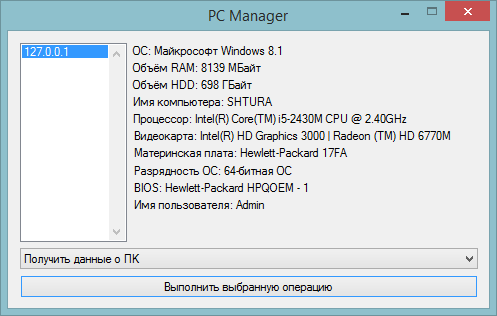


Рисунок В3.3 – Выбор пункта меню «Добавить файл»

При выполнении получение какого-либо списка(например, запущенных процессов) информация будет представлена в модальном диалоговом окне (рисунок В3.4).

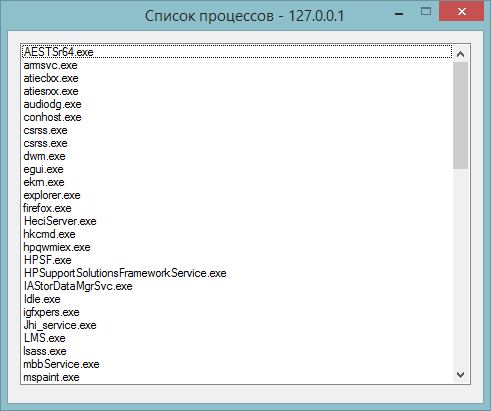


Рисунок В3.4 – Результат операции «Получить список запущенных процессов»

**В4 ВЫПОЛНЕНИЕ ПРОГРАММЫКЛИЕНТА**

После запуска приложения-клиента, на экране появится главное окно программы (рисунок В4.1).

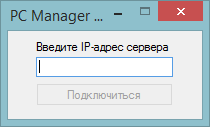


Рисунок В4.1 – Главное окно программы-клиента

Далее вводим IP- адрес сервера и нажимаем «Подключиться» (рисунок В4.2).

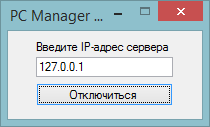


Рисунок В4.2– Подключение к серверу

Единственное оставшееся действие – это отключение от сервера.

**В5 СООБЩЕНИЕ СИСТЕМНОМУ ПРОГРАММИСТУ**

Перед установкой данного программного продукта необходимо сравнить системные требования приложения «PCManager» с конфигурацией ПК (версия операционной системы, наличие свободного места на диске, имеется ли необходимый объем ОЗУ).

Далее необходимо запустить приложение.