**Лабораторная работа по курсу**

**«Использование библиотеки Windows Forms»**

**Тема: Использование расширенного текстового поля, дерева и списка ч.1**

**Цель:**

Закрепить у слушателей практические навыки и теоретические знания при взаимодействии с элементами управления. Научиться создавать приложения **Windows Forms**.

**Необходимые инструменты:** MS Visual Studio 2013

**Документация:** Конспект, Литература

**Ориентировочное время исполнения:** 2 часа.

**Задание 1.** Разработать приложение **«Статистика Email»**.

Цель: реализовать приложение, позволяющее организовать работу с SMTP сервером. Программа должна позволять осуществлять рассылку писем на выбранные e-mail-адреса по заранее составленному расписанию.

Для отправки письма используем код, представленный в приложении 1.

Технические требования к реализации приложения.

1. Для выбора пользователей, которым будут отправляться письма, необходимо воспользоваться элементом управления TreeView. Заранее с использованием свойства Nodes элемента управления TreeView создаются группы получателей (дерево может быть другое по содержанию, но минимальное число уровней – «3») (рисунок 1):

***«Получатели»***

***«Родственники»***

***«Близкие родственники»***

***«Дальние родственники»***

***«Сотрудники»***

***«Отдела»***

***«Фирмы»***

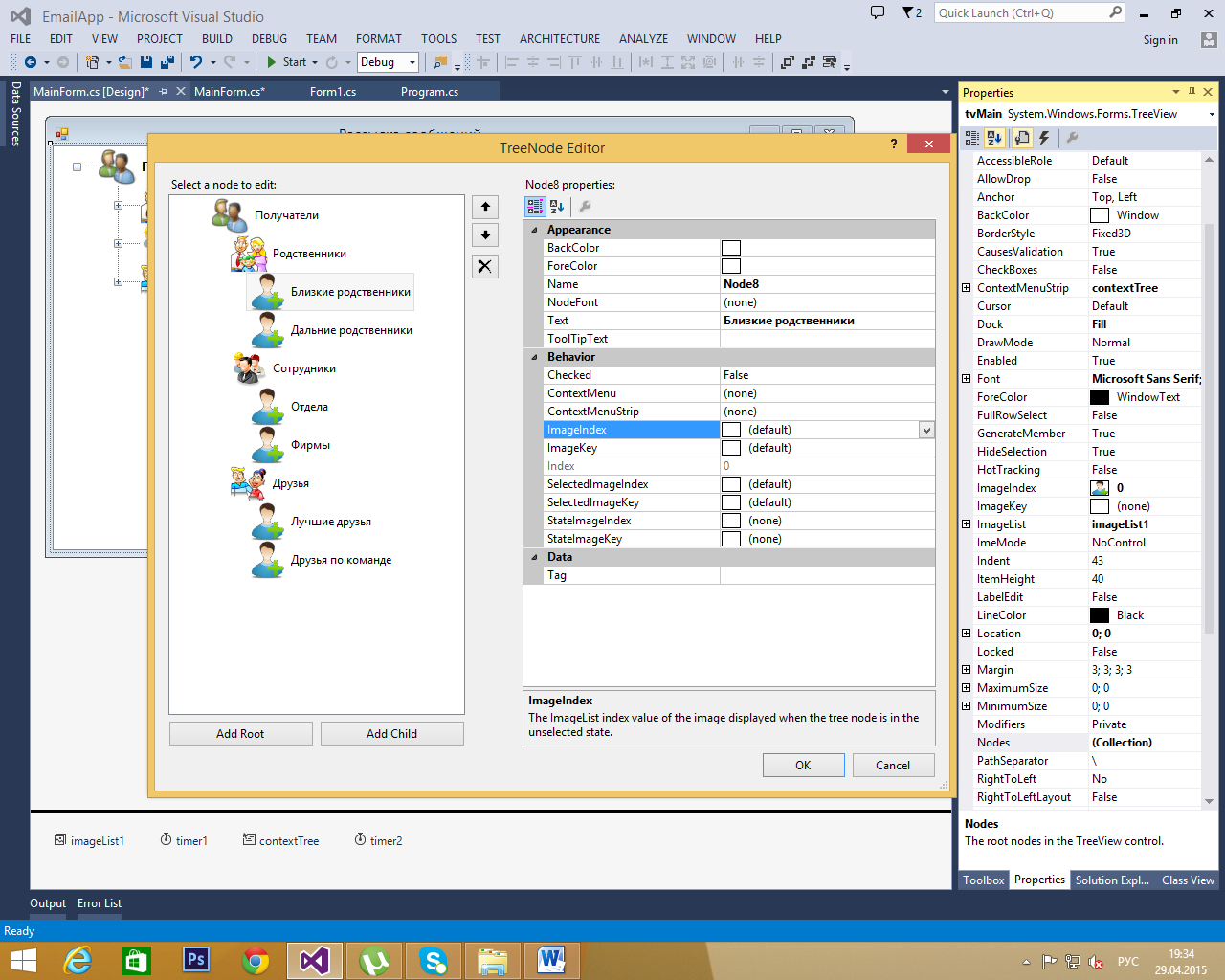
***«Друзья»***

***«Лучшие друзья»***

***«Друзья по команде»***

Для привлекательности работы с «деревом» необходимо настроить его параметры (настройка параметров по выбору разработчика).

Основные свойства для работы с элементов TreeView приведены в приложении 2.



***Рисунок 1.***

Далее для последних узлов добавляем контекстное меню, которое обеспечит добавление с использованием контекстного меню в выбранный узел дерева элемента «имя получателя - email» (см. рисунок 2).

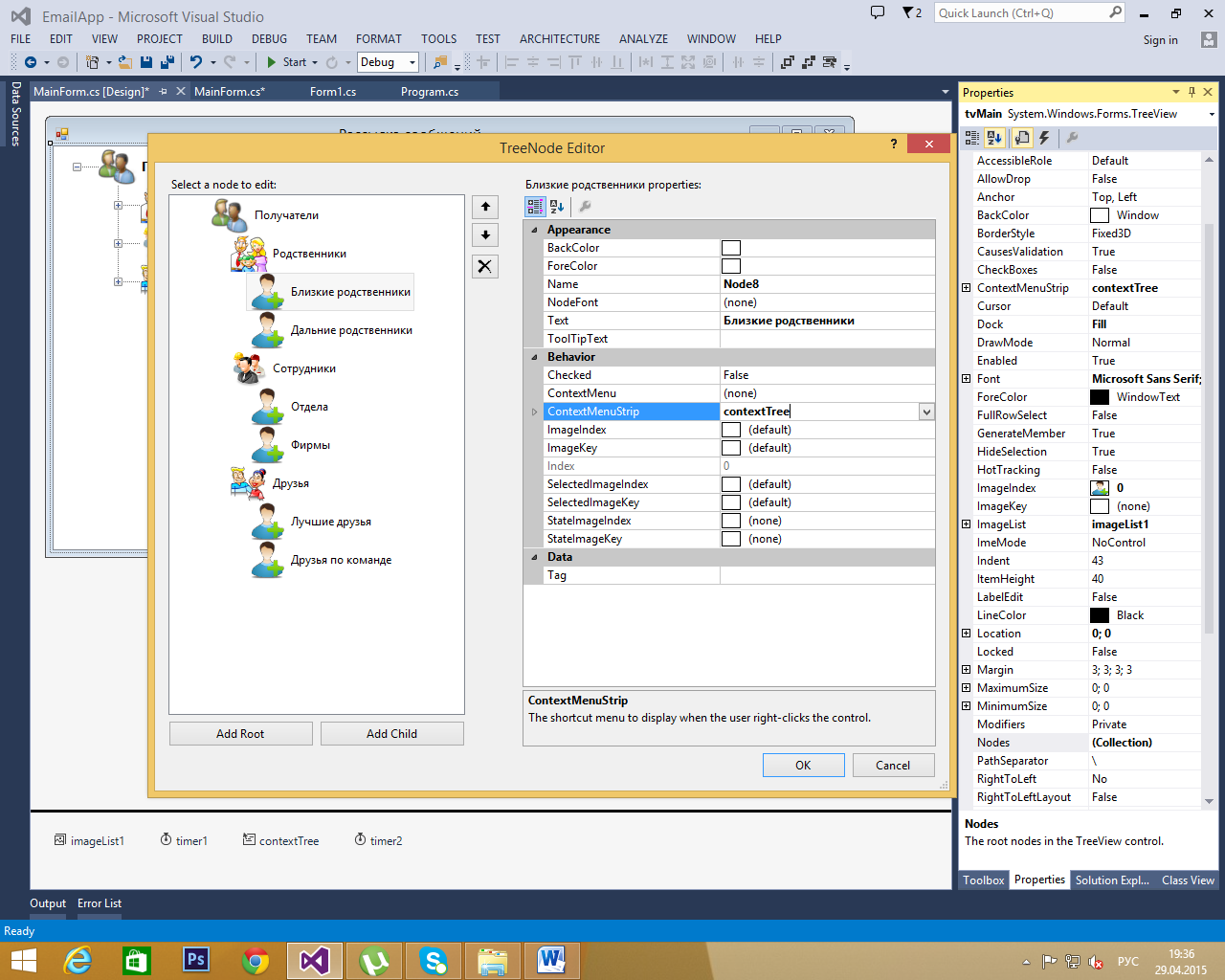


Рисунок 2.

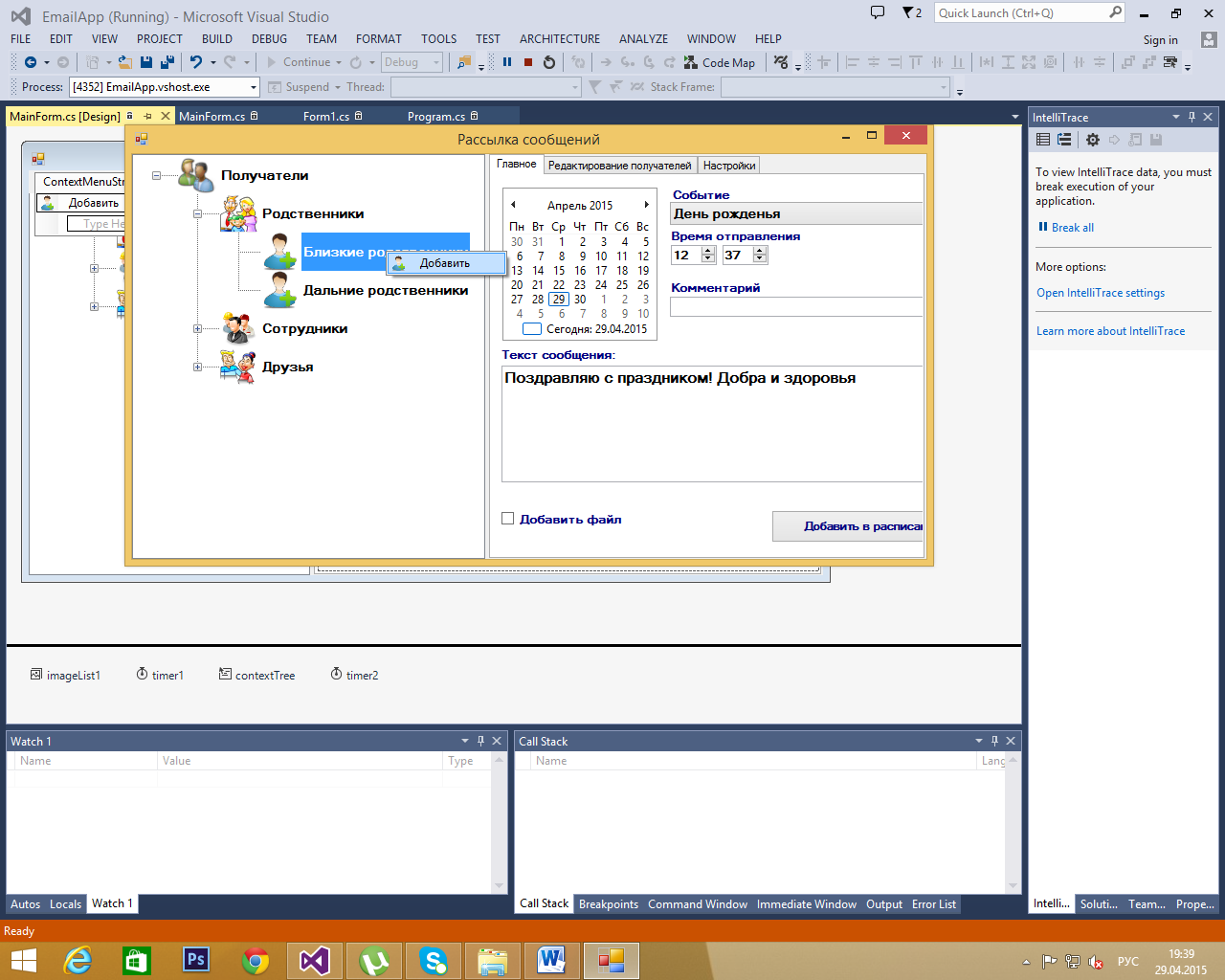


Рисунок 3 Отображение меню.

По нажатию меню «Добавить» открывается форма для добавления нового элемента в дерево (рисунок 4).

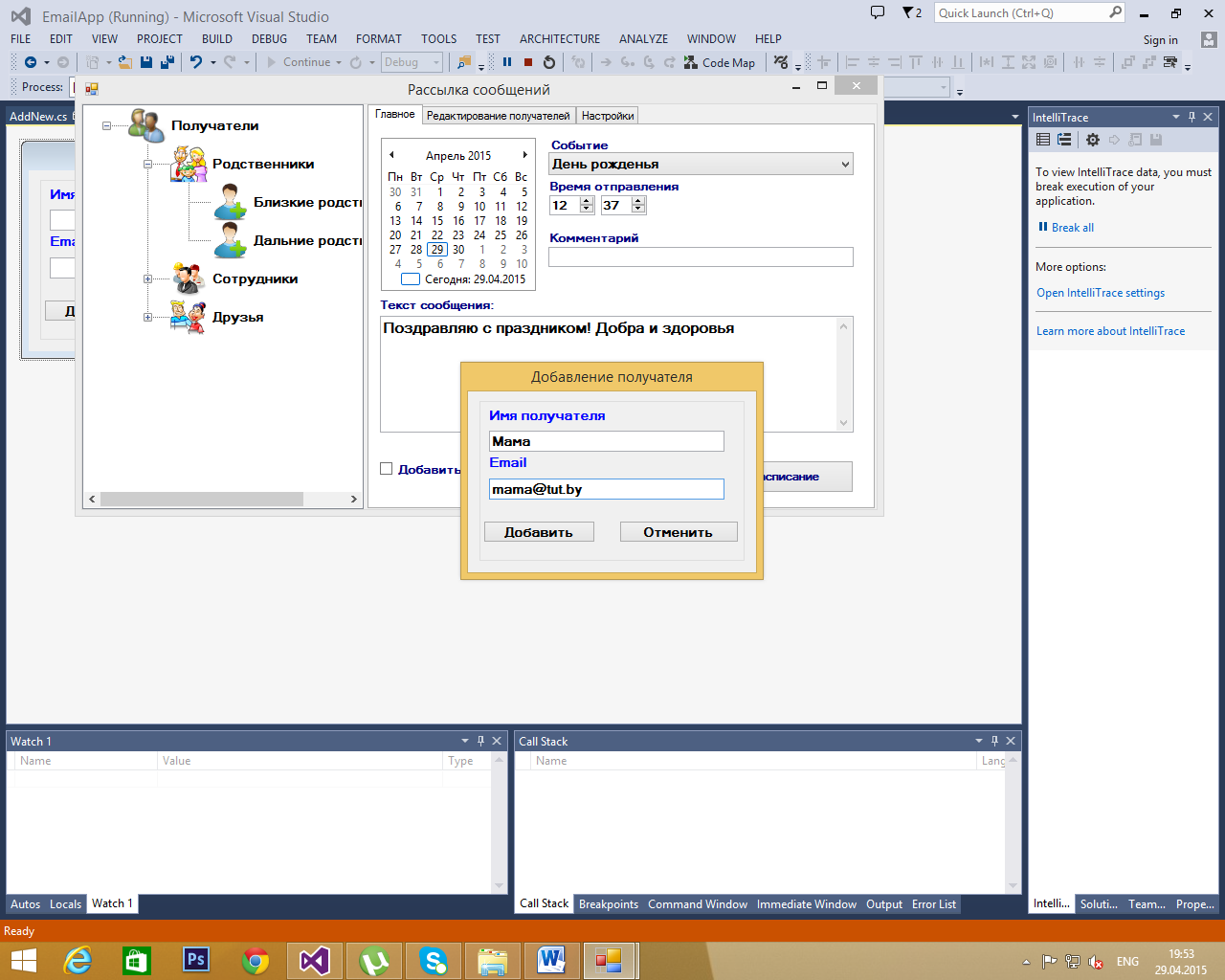


Рисунок 4. Окно ввода параметров нового пользователя.

После нажатия на кнопку «Добавить» в дереве должен появится новый получатель (рисунок 5)

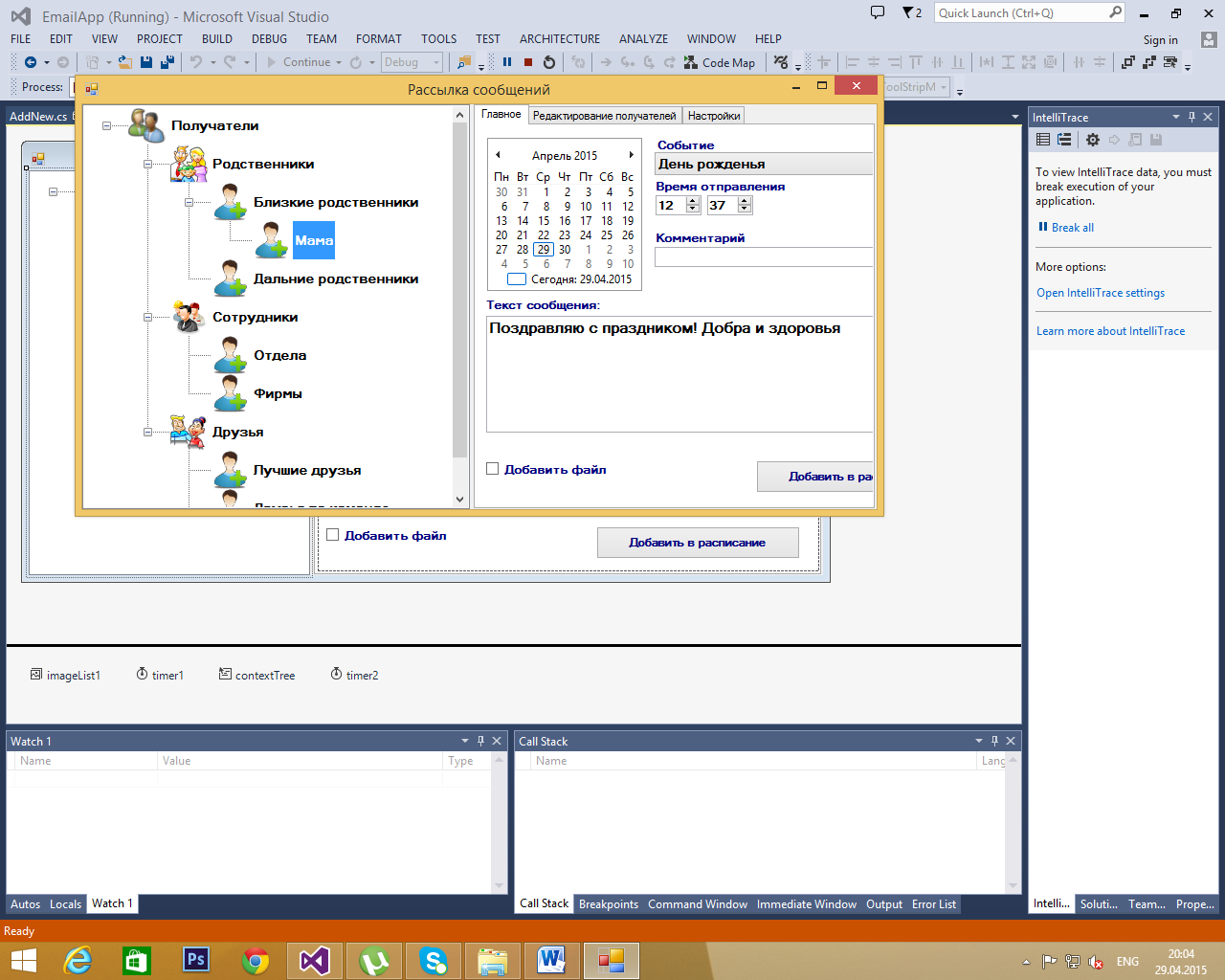


Рисунок 5.

Для хранения введенного email можно выбрать использовать свойство Tag узла (элемента TreeNode).

После добавления в дерево получателей и их email необходимо реализовать следующий функционал: выбираем добавленного получателя, его имя и email отображаются на вкладке «Главное» в полях «Кому» и «Email»(рисунок 6)

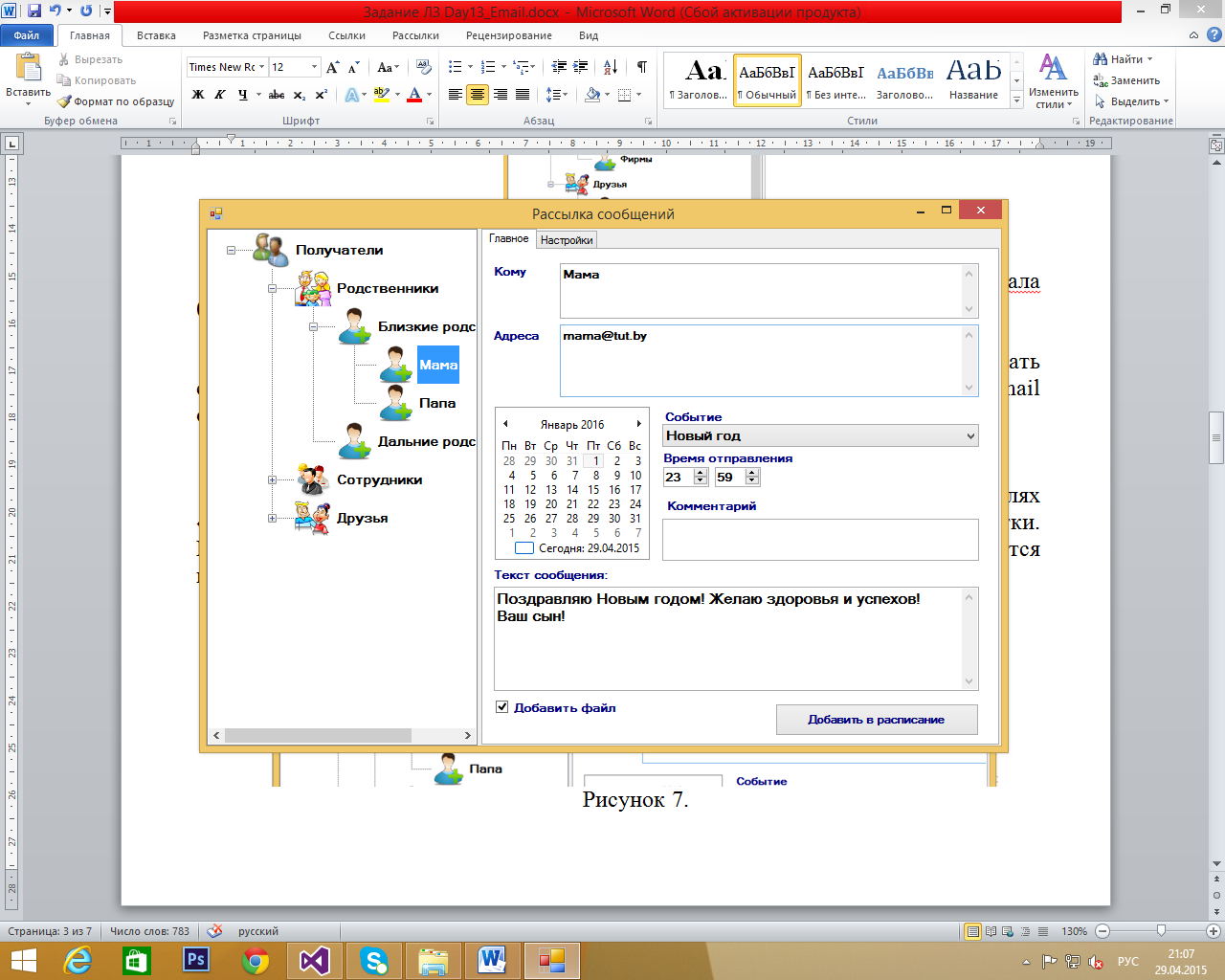


Рисунок 6.

При этом необходимо реализовать, чтобы при выделении группы в полях «Кому» и «Email» отображались имена и email последних узлов выбранной ветки. Например, после выбора ветки «Близкие родственники» в полях отображается информация, представленная на рисунке 7.

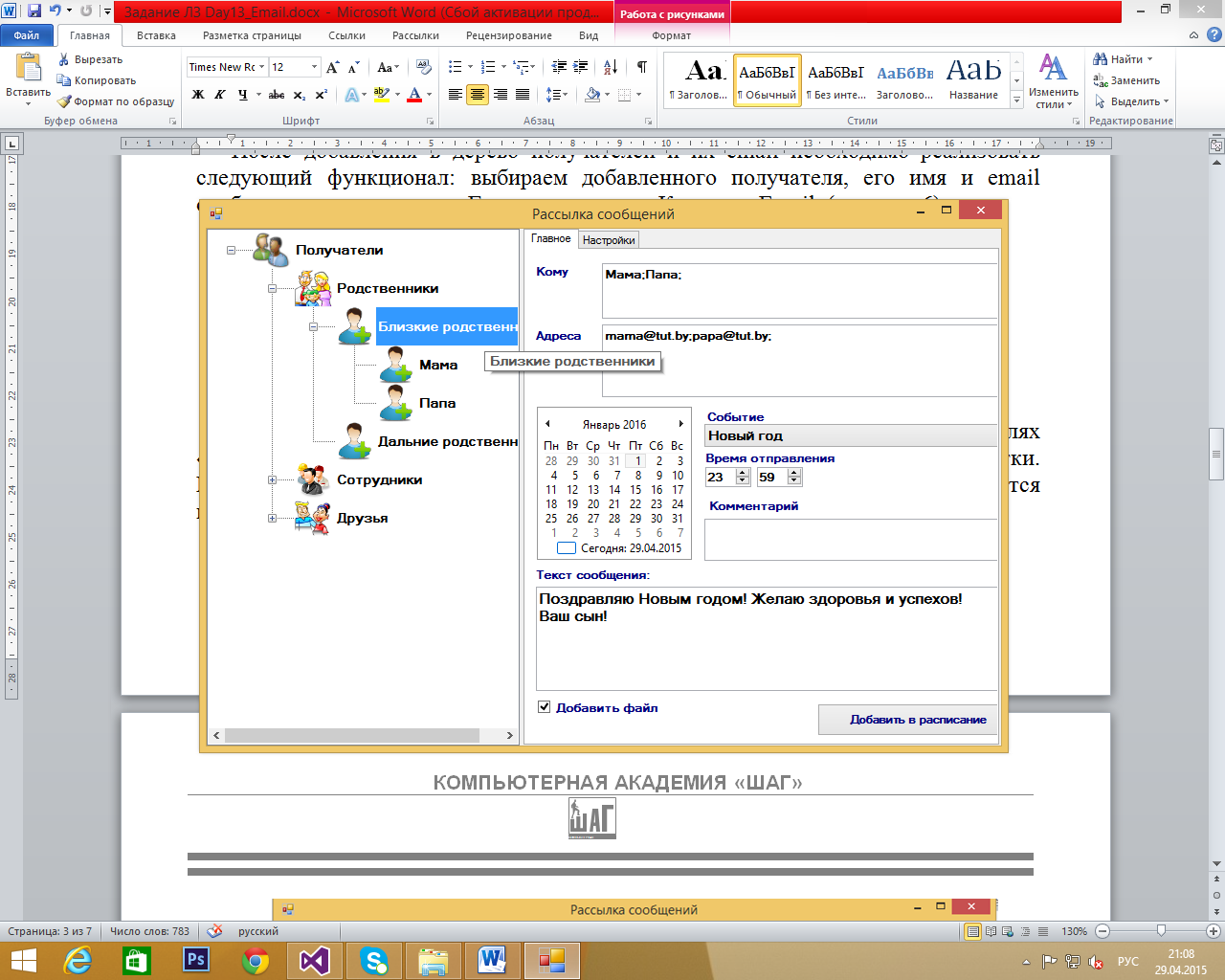


Рисунок 7.

Для реализации функционала, представленного на рисунке 7, необходимо реализовать проход по всем дочерним узлам выбранного узла. Далее представлен пример рекурсивной функции проходящей по всем узлам

**string GetAllNames(TreeNode selTN)**

**{**

**for (int i = 0; i < selTN.Nodes.Count; i++)**

**{ GetAllNames(selTN.Nodes[i]);**

**if (selTN.Nodes[i].Level >= 3)**

**{ selNames += selTN.Nodes[i].Text+";";**

**}**

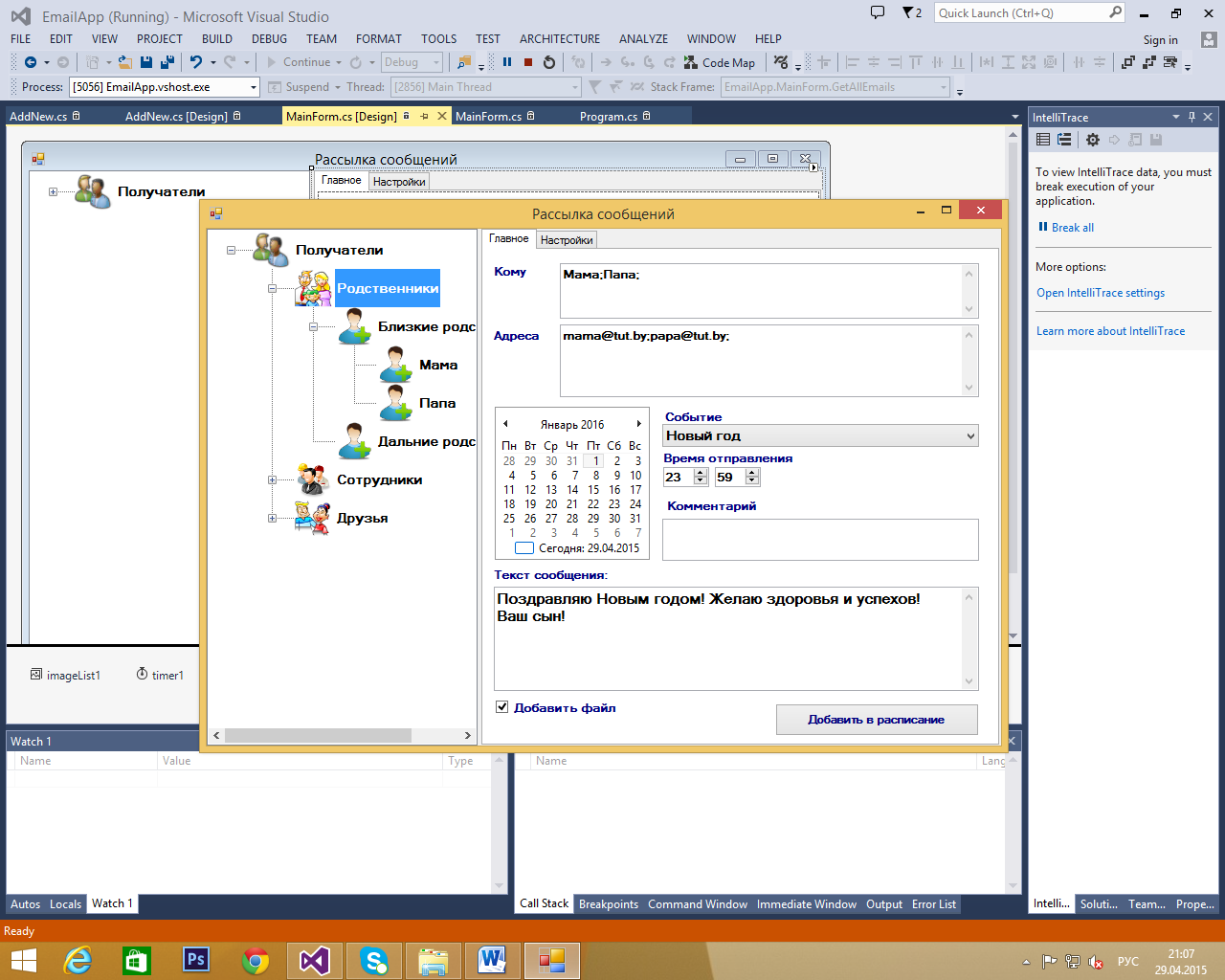
**}**

**return selNames;**

**}**

Далее при получении необходимой информации пользователь может добавить в расписание данное событие. Целесообразно для описания события создать соответствующий класс, а все события хранить в коллекции.

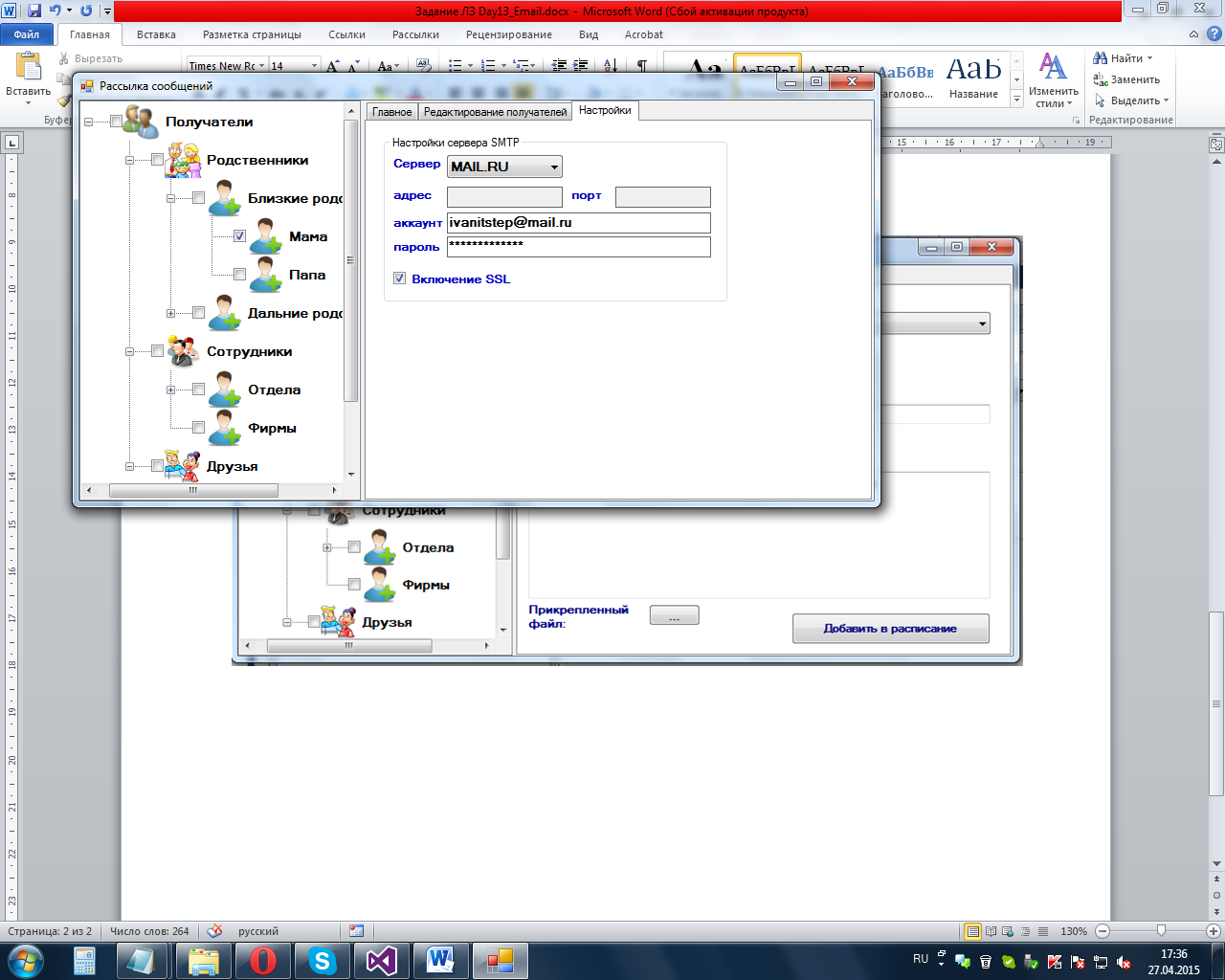
В приложении постоянно работает таймер, который с периодичностью – «одна минута» проверяет если событие на текущее время, если такое событие имеется, то формируется письмо и отправляется на заданный email. Необходимо предусмотреть, если для события имеются несколько получателей, то письма отправляются каждому. Также необходимо обеспечить отправку поздравительной открытки вместе с письмом.



Дополнительно:

- предусмотреть сохранение текущего дерева в файл и восстановление его из файла;

- предусмотреть в настройках программы установку параметров SMTP клиента и сервера.



- предусмотреть добавление и удаление информации в дереве с использованием различных наборов контекстных меню;

- отправку письма реализовать с асинхронном режиме с использованием элемента управления BackgroundWorker.

**ПРИЛОЖЕНИЕ 1**

Пример кода для отправки email.

Для реализации отправки письма используется SMTP. **SMTP (Simple Mail Transfer Protocol)** – это сетевой протокол, предназначенный для передачи электронной почты в сетях TCP/IP. SMTP используется для отправки почты от пользователей к серверам и между серверами для дальнейшей пересылки к получателю. Для приёма почты, почтовый клиент должен использовать протоколы  POP3  или  IMAP. Работа с SMTP происходит непосредственно на сервере получателя. Поддерживает функции: установление соединения, аутентификация, передача данных.

Список SMTP-серверов:

* **Yandex** - SMTP-сервер:  **smtp.yandex.ru; Порт: 25**
* **Gmail** - SMTP-сервер: **smtp.gmail.com; Порт: 58**
* **Mail**.Ru - SMTP-сервер: **smtp.mail.ru; Порт: 25**

**//Создание SMTP клиента (выбираем один из SMTP-серверов)**

**SmtpClient Smtp = new SmtpClient("smtp.mail.ru", 25);**

**// авторизируем подключение, указываем email и пароль,**

**// которые были ранее зарегистрированы (можно воспользоваться указанными!!!)**

**Smtp.Credentials = new NetworkCredential("ivanitstep@mail.ru", "ivan123456789");**

**//включаем поддержку SSL (некоторые сервера это требуют)**

**Smtp.EnableSsl = true;**

**//Формирование письма**

**MailMessage message = new MailMessage();**

**//указываем адрес отправителя!**

**message.From = new MailAddress("ivanitstep@mail.ru");**

**//указываем адрес получателя!**

**message.To.Add(new MailAddress("<e-mail>"));**

**//указываем заголовок письма**

**message.Subject = "Заголовок письма ";**

**//указываем ТЕКСТ письма**

**message.Body = "Текст письма!";**

**// Прикрепление файла**

**Attachment attach = new Attachment(<имя файла>, MediaTypeNames.Application.Pdf);**

**// Добавляем информацию для файла (дополнительно!)**

**ContentDisposition disposition = attach.ContentDisposition;**

**// указываем дату создания**

**disposition.CreationDate = System.IO.File.GetCreationTime(file);**

**//добавление вложения**

**message.Attachments.Add(attach);**

**//отправка письма**

**Smtp.Send(message);**

**ПРИЛОЖЕНИЕ 2**

Основные конструкции по работе с TreeView

1. Получение выделенного в дереве узла

**TreeNode selTr = tvMain.SelectedNode;**

1. Получение уровня выделенного в дереве узла

**TreeNode selTN = tvMain.SelectedNode;**

**Level = selTN.Level;**

1. Добавление нового узла в выделенный узел

**TreeNode newTN = new TreeNode();**

**newTN.Text = addNew.NameUser;**

**TreeNode selTr = tvMain.SelectedNode;**

**selTr.Nodes.Add(newTN);**

1. Добавление дополнительной информации в узле

**TreeNode newTN = new TreeNode();**

**newTN.Text = addNew.NameUser;**

**newTN.Tag = addNew.EmailUser;**

1. Отображение подсказок в TreeView

**tvMain.ShowNodeToolTips = true;**

**TreeNode newTN = new TreeNode();**

**newTN.ToolTipText = addNew.EmailUser;**