

Лабораторная работа №3

Цель и задачи

- изучение создания системных служб Windows;
- создание службы для загрузки почты.

Описание задания

Создать windows-службу, которая через определенный момент времени, производит опрос сервера на наличие почты на нем (с помощью внешней библиотеки OpenPop).

При получении почты, информация каждого письма должна сохраняться в отдельный каталог в следующем виде:

- весь текст, который есть в сообщении необходимо записать в текстовый файл;
- прикрепленные к письму файлы также должны быть сохранены в каталоге письма, при этом тип сохраненного файла должен соответствовать действительности. (Например: музыкальный файл никак не должен быть с расширением docx).

Служба должна иметь 2 основных метода: `onStart()` и `onStop()` и они оба должны выполнять свою функцию в соответствии с названием.

Служба должна запускать опрос сервера с определенной периодичностью, периодичность должна быть не чаще 3 минут, задается с помощью конфигурационного файла (данную функцию реализовать с помощью класса `Timer`).

Для службы должен быть создан конфигурационный файл, позволяющий задать следующие параметры:

- адрес pop-сервера (imap-сервера);
- логин;
- пароль;
- периодичность опроса;
- указание базового каталога хранения сообщений.

При последующем опросе почтового сервера производить проверку на существование данного сообщения в директории:

- если сообщение уже существует, то переходить к следующему сообщению;
- если сообщения не существует, тогда необходимо его подгрузить;
- если существует сообщение, которого не было в списке сообщений с сервера, такое письмо следует удалить.

Технические требования

При выполнении задания должны быть использованы классы `Pop3Client` (библиотеки `OpenPop`) для установления соединения с pop-сервером. Класс `Timer` для запуска основного метода с определенным интервалом.

В итоге должна получиться скомпилированная служба, которую в итоге необходимо инициализировать и запустить, данные для подключения к pop-серверу должны браться из конфигурационного файла. После демонстрации выполненной работы службу необходимо деинсталлировать.

Теория и примеры реализации

Основной материал был найден на сайте Microsoft.

Создание службы

Создания проекта и описание класса службы

Создание проекта службы Windows (см. рисунок 3.1).

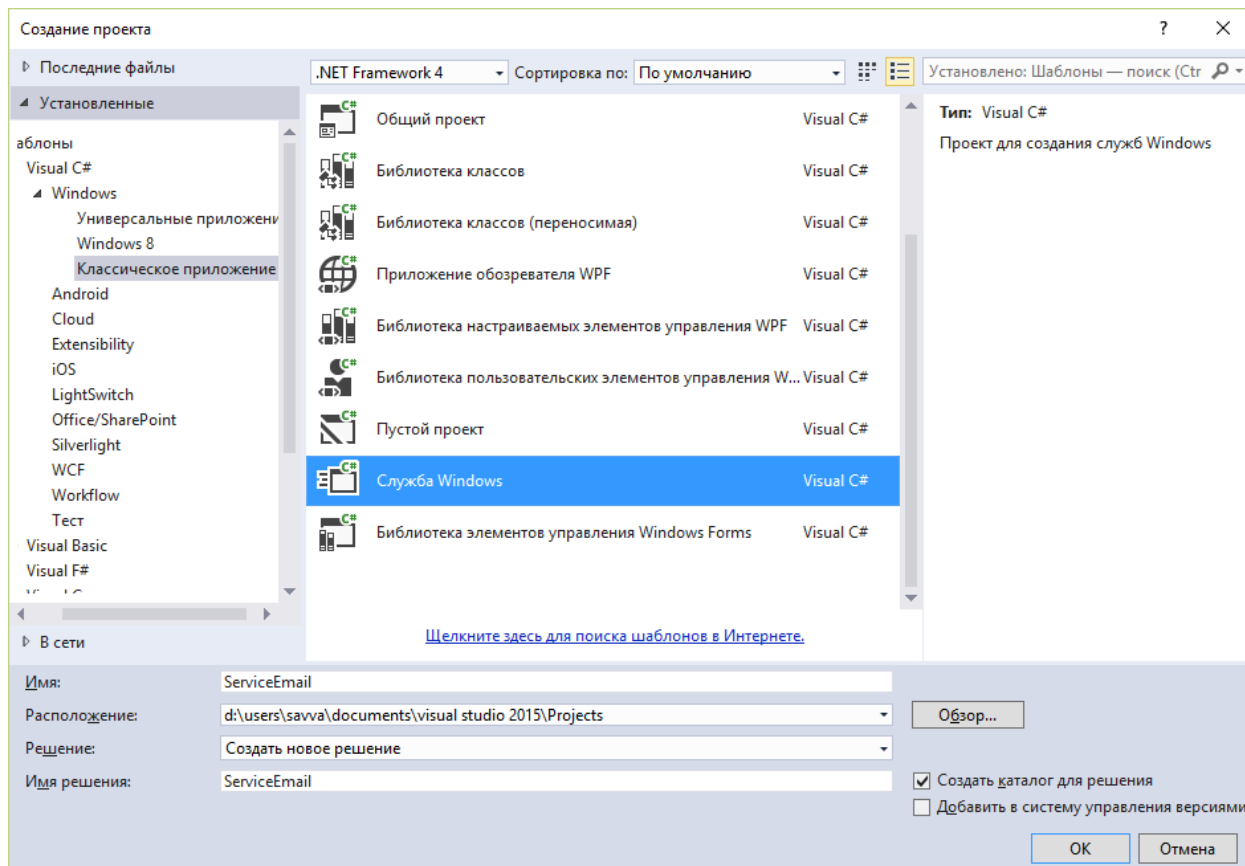


Рисунок 3.1 – Форма создания «Службы Windows»

В обозревателе решений открываем файл Service1.cs. В открывшейся вкладке необходимо нажать на ссылку «щелкните здесь для перехода в режим отображения кодов».

Инициализируем в методе OnStart() – таймер, для того чтобы событие работало не постоянно а с некоторым интервалом ожидания (см. рисунок 3.2).

```
//объявление глобальной переменной типа Timer
Timer tr = null;
//Событие запуска сервиса
ссылка 0
protected override void OnStart(string[] args)
{
    File.AppendAllText("MyFirstService.log",
        Environment.NewLine + "Start service");

    //Создание экземпляра класса Timer
    tr = new System.Timers.Timer();
    //Инициализация параметра "интерва"
    tr.Interval = 10000 * 10; //10000 - 1с.
    //работает без перерывов
    tr.AutoReset = true;
    //инициализация нашего события OnTimedEvent
    tr.Elapsed += new ElapsedEventHandler(OnTimedEvent);
}
//событие которое будет производить запись в файл
ссылка 1
private void OnTimedEvent(object source, ElapsedEventArgs e)
{
    //запись в конец файла
    File.AppendAllText("MyFirstService.log",
        Environment.NewLine + " " + DateTime.Now.ToString("dd/MM/yyyy hh:mm:ss"));
}
//Событие остановки сервиса
ссылка 0
protected override void OnStop()
{
    File.AppendAllText("MyFirstService.log",
        Environment.NewLine + "Stop service");
}
```

Рисунок 3.2 – листинг службы

При запуске службы будет создавать и стартовать *непрерывный* таймер с инициализацией параметров. Таймер в свою очередь, будет запускать метод OnTimedEvent по истечению заданного интервала (то есть 10 секунд).

При остановке службы, таймер будет останавливать свою работу.

Класс Timer:

Свойство:

- AutoReset – дает указание работы класса Timer, false – timer прорабатывает один раз, true – timer будет вызывать событие неоднократно.
- Interval – задает значение частоты вызова события Elapsed. В миллисекундах.

Методы:

- Start – начало вызова события Elapsed.
- Stop – прекращает вызов события Elapsed.

События:

- Elapsed – событие которое будет вызываться при запуске таймера.

Добавление установщиков в службу

Переходим в конструктор класса службы. В *Обозревателе решений* выделить файл в котором описана работа службы, нажимаем правой кнопки мышки (ПКМ) и выбираем пункт меню «Открыть в конструкторе».

В открывшейся вкладке, в центре поля нажать правую кнопку мышки (ПКМ) и выбрать пункт меню «Добавить установщик».

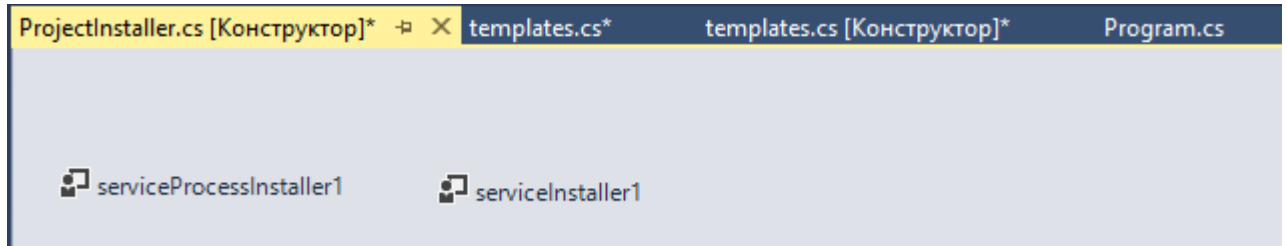


Рисунок 3.3 – вкладка установщика сервиса

Дальше устанавливаем свойства.

1. Выделяем объект `serviceInstaller1` ПКМ -> Свойства.

Нужно изменить несколько свойств (см. рисунок 3.4):

- `ServiceName` – имя службы;
- `Description` – описание службы, информация также отображается в окне управления службами, и помогает пользователям идентифицировать службы;
- `DisplayName` – текст, который будет отображаться в окне «Службы» в столбце *Имя*.
- `StartType` – тип запуска службы: автоматический, ручной, отключен. Обычно устанавливают автоматический.

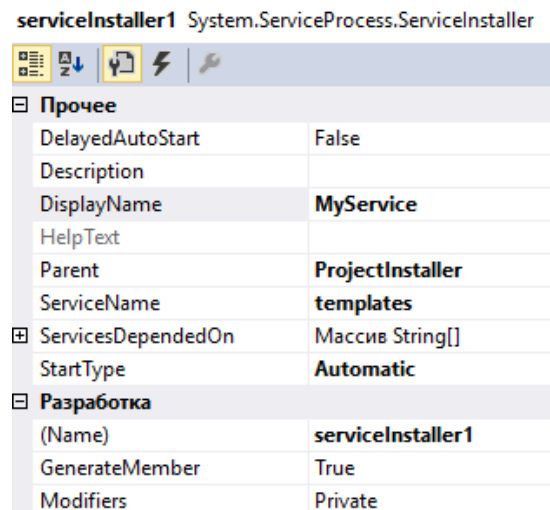


Рисунок 3.4 – пример заполнения свойств инсталлера

2. Выделяем объекта `serviceProcessInstaller1` ПКМ -> Свойства.

Нужно изменить только одно свойство: `Account` присвоить ему значение **LocalSystem**. Получает или задает тип учетной записи, под управлением которой должно запускаться данное служебное приложение.

Построение проекта службы

Для того чтобы построить проект службы необходимо выполнить следующие действия:

1. В *Обозревателе решений* открыть контекстное меню для проекта – выбрать пункт меню *Свойства*.
2. На вкладке *Приложения* в списке *Автоматически запускаемый объект* нужно выбрать класс программ, например, *Service1.Program*.
3. В *Обозревателе решений* открыть контекстное меню проекта, и произвести *Сборку* проекта.

Установка службы

Для того чтобы установить службу нужно сделать ряд действий:

1. Открыть *Командная строка разработки* в разделе *Инструменты Visual Studio* в меню *Пуск*.
2. В окне *Командная строка* откройте папку, содержащую выходные данные проекта. Например, в папке *Мои документы* перейдите в папку "Visual Studio 2013\Projects\MyNewService\bin\Debug".
3. Введите следующую команду:

```
installutil.exe MyNewService.exe
```

Если служба установлена успешно, *installutil.exe* сообщит об успешном завершении работы. Если система не может найти файл *InstallUtil.exe*, убедитесь, что он существует на вашем компьютере.

Запуск и выполнение службы

1. В Windows откройте экран *Пуск* или меню *Пуск* и введите *services.msc*.

В окне *Службы* должна появиться служба с именем *MyNewService*.

2. В окне *Службы* откройте контекстное меню службы и нажмите *Пуск*.
3. Откройте контекстное меню службы и щелкните *Остановить*.
4. (Необязательно) Для запуска и остановки службы в командной строке можно использовать команды *net start ServiceName* и *net stop ServiceName*.

Удаление службы Windows

1. Откройте командную строку разработчика с учетными данными администратора.
2. В окне "Командная строка" откройте папку, содержащую выходные данные проекта. Например, в папке "Мои документы" перейдите в папку "Visual Studio 2013\Projects\MyNewService\bin\Debug".
3. Введите следующую команду:

```
installutil.exe /u MyNewService.exe
```

Работа с почтой

Для чтения почты с сервера можно использовать библиотеку *OpenPop*.

Подключение библиотеки *OpenPop*

В *Обозревателе решений* в ветке проекта в контекстном меню объекта *References* выбрать «Управление пакетами NuGet...» (см. рисунок 3.5).

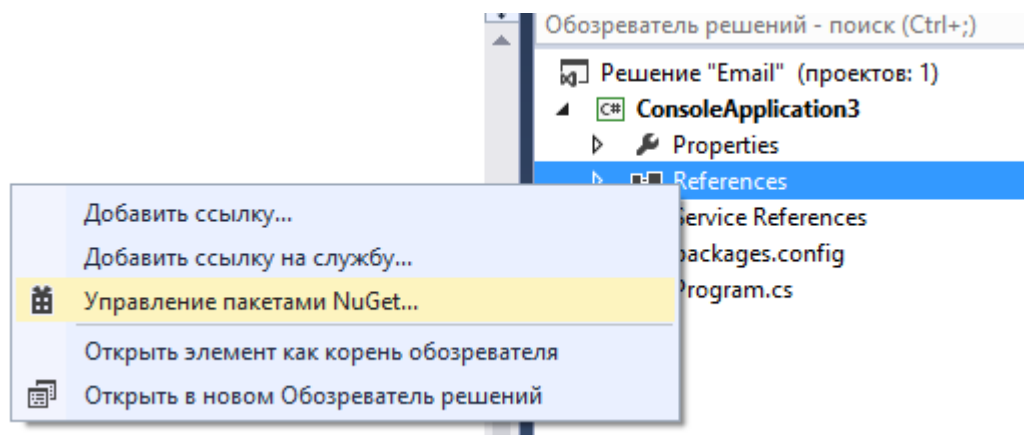


Рисунок 3.5 – Управление пакетами NuGet

Далее в открывшемся окне в поле *поиск* прописываем OpenPop, должен появиться список библиотек, для данной лабораторной работы нужна библиотека под названием *OpenPop POP3 Mail Library* (см. рисунок 3.6), устанавливаем ее. Если она не установлена установить.

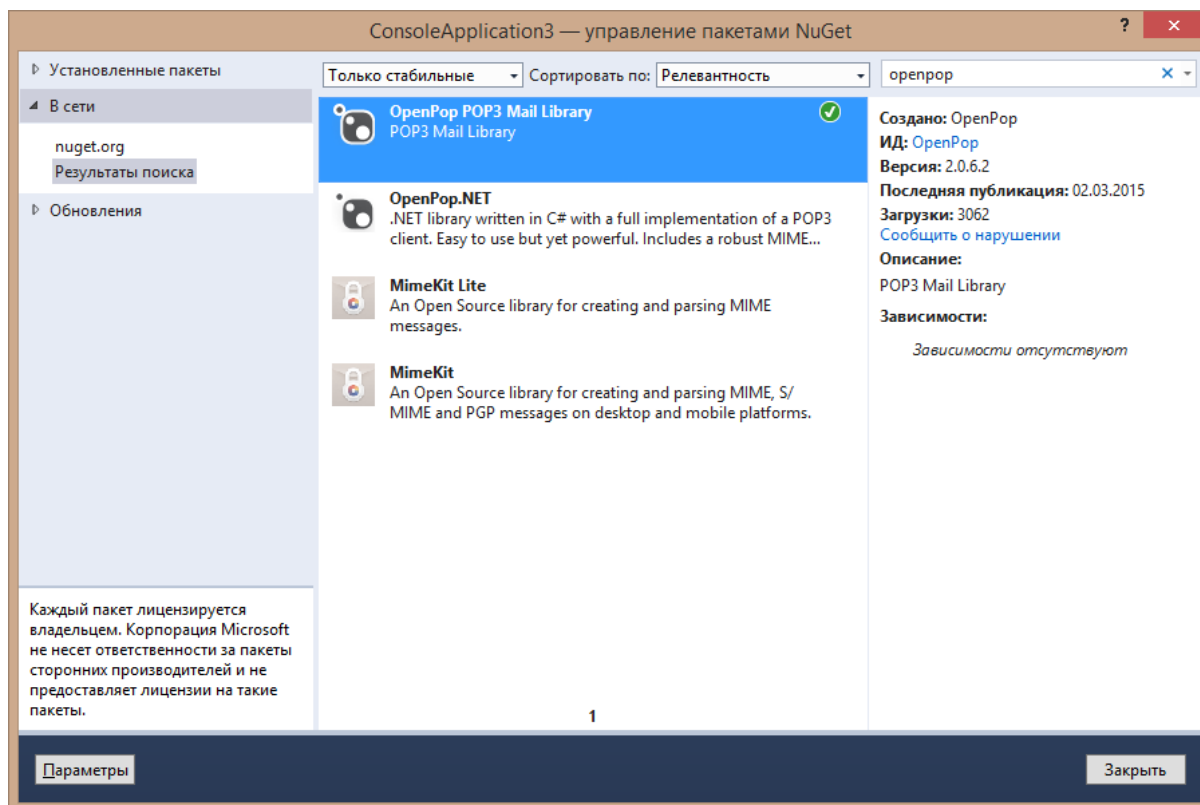


Рисунок 3.6 – Библиотека OpenPop

Работа с библиотекой OpenPop

Работа с библиотекой OpenPop хорошо рассмотрена на примерах на сайте: <http://hpop.sourceforge.net/examples.php>.