## 3.4 Разработка алгоритма работы программы

Для реализации поставленной задачи был разработан алгоритм работы программы, схема которого изображена на рисунке 3.4.



А

B

Рисунок 3.4 – Обобщенный алгоритм работы службы (начало)



Рисунок 3.5 – Обобщенный алгоритм работы службы (конец)

B

А



Рисунок 3.6 – Алгоритм работы метода GetAttachments ()

## 3.5 Разработка методов классов

Для реализации задуманной архитектуры проекта были использованы методы и функции, которые описаны в таблице 3.2.

Таблица 3.2 – Описание методов класса MailClient

|  |  |
| --- | --- |
| Метод | Описание |
| OnStart(string[] args) | Метод, который вызывается при событии запуска службы. В данном методе осуществляется подключение к почтовому серверу, авторизация пользователя и настройка таймера службы.  − уровень доступа – protected;  − возвращаемое значение – void;  – args – параметры командной стшроки. |
| OnStop() | Метод, который вызывается при событии остановки службы. В данном методе осуществляется освобождение ресурсов.  − уровень доступа – protected;  − возвращаемое значение – void; |
| loopService(object sender, System.Timers.ElapsedEventArgs e) | Метод, который вызывается по событию тика таймера. Данный метод является основным рабочим циклом программы. В данном методе осуществляется выбор новых писем, сохранения их в нужные каталоги, выбор критерия размещения письма.  − уровень доступа – private;  − возвращаемое значение – void;  – sender – объект, который вызвал данный метод. |
| GetAttachments(ImapX.Message message, string path) | Метод предназначен для извлечения вложений из письма (или сохранения текста письма в файл), реализует извлечение файлов архива в подкаталог или в корневой каталог письма, в зависимости от конфигураций.  − уровень доступа – private;  − возвращаемое значение – void;  – message – объект, письма;  – path – путь к подкаталогу письма. |
| fileName(string dateTime) | Метод предназначен для формирования имени файла куда будет сохраняться текст письма, в котором нет вложений  − уровень доступа – private;  − возвращаемое значение – string: строка сформированная из MessageData;  – dateTime – текущая дата-время в формате MessageData. |
| GetConfInfo() | Метод предназначен для извлечения настроек из конфигурационного файла.  − уровень доступа – private;  − возвращаемое значение – void; |

Таблица 3.3 – Описание методов класса WorkWithConfig

|  |  |
| --- | --- |
| Метод | Описание |
| SetConfigValue(string name, string value) | Метод, для записи данных в конфигурационный файл.  − уровень доступа – public;  − возвращаемое значение – bool: результат записи;  – name – ключ;  – value – значение. |
| GetConfigValue(string name, string filename) | Метод, для извлечения данных из конфигурационного файла.  − уровень доступа – public;  − возвращаемое значение – string: значение запрашиваемого поля;  – name – ключ;  – filename – путь к файлу конфигураций. |

Таблица 3.4 – Описание методов класса Configuration

|  |  |
| --- | --- |
| Метод | Описание |
| Initialisation(object sender, EventArgs e) | Метод, для извлечения данных из конфигурационного файла.  − уровень доступа – private;  − возвращаемое значение – void; |
| SaveToolStripMenuItem\_Click(object sender, EventArgs e) | Метод, для добавления выбранных настроек в конфигурационный файл.  − уровень доступа – private;  − возвращаемое значение – void; |
| RestartService() | Метод, перезапуска службы после внесения изменений в конфигурационный файл.  − уровень доступа – private;  − возвращаемое значение – void; |

Таблица 3.5 – Описание переменных класса MailClient

|  |  |
| --- | --- |
| Переменная | Описание |
| private StreamWriter file | Поток для записи в файл (для ведения лога работы службы) |
| private StreamWriter error | Поток для записи в файл (для ведения лога ошибок) |
| private System.Timers.Timer timer1 | Таймер для управлением службой |
| private ImapX.ImapClient client | Объект почтового сервера |
| private const string pathToLog | Путь к лог-файлу |
| Process process | Переменная для получения реального расположения установленной службы |
| FileInfo finfo | Переменная для получения реального расположения установленной службы |
| private bool isLogined | Переменная для проверки аутентификации пользователя |
| private string criteries | Конфигурация: критерии для размещения писем (шаблон >> каталог) |
| private string rootPath | Конфигурация: путь к коревому каталогу |
| private string otherDir | Конфигурация: название подпапки, куда помещать письма, которые не подошли ни к одному критерию |
| private string login | Конфигурация: логин пользователя от почтового ящика |
| private string password | Конфигурация: пароль пользователя от почтового ящика |
| private string server | Конфигурация: почтовый сервер |
| private string name | Конфигурация: название лог файла |
| private int frequency | Конфигурация: частота обращения к почтовому серверу |
| private int port | Конфигурация: порт почтового сервера |
| private int safety | Конфигурация: протокол безопасности почтового сервера |
| private int maxSize | Конфигурация: максимальный размер лог-файла |
| private int saveTime | Конфигурация: время хранения лог-файла на диске |
| private bool autoMode | Конфигурация: настройка, отвечающая за автоматическое извлечение файлов из архива |
| private bool mkNewDir | Конфигурация: настройка, отвечающая за создание дополнительного подкаталога для извлечения содержимого архива |