**Практическая работа 2**

**Цель практическое работы:**

Создать приложение по отправки email писем из приложения разработанного на Python.

**Задачи:**

* Доработать приложение по отправки email писем
* Необходимо сделать отправку с любого ящика
* Сделать добавление файлов в письмо

**Ход работы**

В данной практической работы, нужно будет доработать приложение.

Для начала доработаем форму. форма должна содержать следующие элементы (Рис.1.)

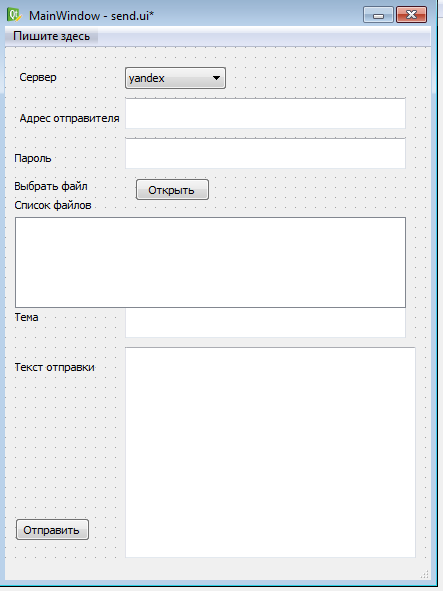


Рис.1 – Форма отправки email писем

Для того, чтобы была возможность отправлять email письма с любого email адреса(домена), поместим на форму «comboBox» и вставим в comboBox значения по умолчанию (рис 2.)

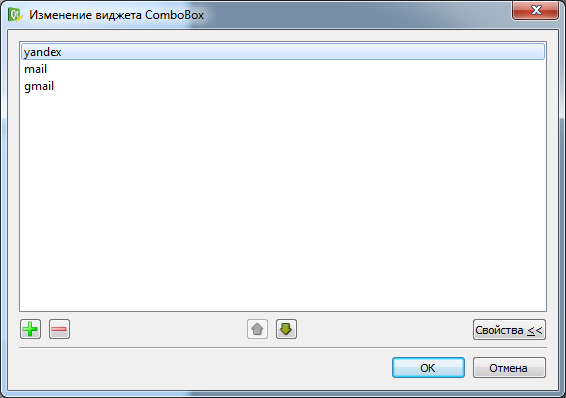


Рисунок 2. – значения по умолчанию.

Теперь, вставим в функцию отправления обработчик события, какая позиция выбрана, те настройки и применяем:

*# Создаем объект SMTP*

**if** win.ui.comboBox.currentIndex()==0:  
 server = smtplib.SMTP\_SSL(**'smtp.yandex.ru'**, 465) **elif** win.ui.comboBox.currentIndex()==1:  
 server = smtplib.SMTP\_SSL(**'smtp.mail.ru'**, 25)  
**elif** win.ui.comboBox.currentIndex() == 2:  
 server = smtplib.SMTP\_SSL(**'smtp.gmail.com'**, 587)

Здесь **win –** объект класса Win(QtGui.QMainWindow),**ui** – соответственно, наша форма.

Для отправки email писем с вложением, добавим элементы на форму: **Button,Listview.**

Необходимо создать коннект и событие по нажатию на **Button,** для открытия диалогового окна с возможностью выбора файлов.

Ниже приведен пример:

self.connect(  
 self.ui.open\_file,  
 QtCore.SIGNAL(**'clicked()'**),  
 open\_file)

«open\_file» – Внешняя функция.

Описание функции «open\_file»:

**def** open\_file():  
 filename= QtGui.QFileDialog.getOpenFileName(  
 parent=win,  
 caption=**u'Открыть файл'**,  
 directory=PATH,  
 filter=**'AllFiles (\*)'** )  
 **if** filename:  
 win.path=filename  
 win.ui.label\_4.setText(filename)

Здесь необходимо доработать функцию. Функция должна вставлять имена файлов в **Listview.**

Для добавления файлов в письмо, необходимо написать функцию по обработке списка, добавляемых к сообщению файлов и функцию по добавлению конкретного файла к сообщению.

По нажатию кнопки «Отправить» необходимо вставлять файлы в письмо. Для этого напишем функцию прикрепления файлов:

**def** process\_attachement(msg, files): *# Функция по обработке списка, добавляемых к сообщению файлов* **for** f **in** files:  
 **if** os.path.isfile(f): *# Если файл существует* attach\_file(msg,f) *# Добавляем файл к сообщению* **elif** os.path.exists(f): *# Если путь не файл и существует, значит - папка* dir = os.listdir(f) *# Получаем список файлов в папке* **for** file **in** dir: *# Перебираем все файлы и...* attach\_file(msg,f+**"/"**+file) *# ...добавляем каждый файл к сообщению***def** attach\_file(msg, filepath): *# Функция по добавлению конкретного файла к сообщению* filename = os.path.basename(filepath) *# Получаем только имя файла* ctype, encoding = mimetypes.guess\_type(filepath) *# Определяем тип файла на основе его расширения* **if** ctype **is** None **or** encoding **is not** None: *# Если тип файла не определяется* ctype = **'application/octet-stream'** *# Будем использовать общий тип* maintype, subtype = ctype.split(**'/'**, 1) *# Получаем тип и подтип* **if** maintype == **'text'**: *# Если текстовый файл* **with** open(filepath) **as** fp: *# Открываем файл для чтения* file = MIMEText(fp.read(), \_subtype=subtype) *# Используем тип MIMEText* fp.close() *# После использования файл обязательно нужно закрыть* **elif** maintype == **'image'**: *# Если изображение* **with** open(filepath, **'rb'**) **as** fp:  
 file = MIMEImage(fp.read(), \_subtype=subtype)  
 fp.close()  
 **elif** maintype == **'audio'**: *# Если аудио* **with** open(filepath, **'rb'**) **as** fp:  
 file = MIMEAudio(fp.read(), \_subtype=subtype)  
 fp.close()  
 **else**: *# Неизвестный тип файла* **with** open(filepath, **'rb'**) **as** fp:  
 file = MIMEBase(maintype, subtype) *# Используем общий MIME-тип* file.set\_payload(fp.read()) *# Добавляем содержимое общего типа (полезную нагрузку)* fp.close()  
 encoders.encode\_base64(file) *# Содержимое должно кодироваться как Base64* file.add\_header(**'Content-Disposition'**, **'attachment'**, filename=filename) *# Добавляем заголовки* msg.attach(file) *# Присоединяем файл к сообщению*

Теперь, достаточно передать параметры(объект MIMEMultipart(), массив файлов) в функцию **process\_attachement.**

**Заключение**

В ходе практической работы были закреплены знания языка Python, а так же была освоена технология отправки писем с вложениями. Было написано полностью рабочее приложение по отправки писем.