



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н. Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н. Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

Лабораторная работа № 5

Дисциплина: Компьютерные сети

Тема: smtp-клиент

Студент: Платонова О. С.

Группа: ИУ7-75Б

Вариант: 10

Оценка (баллы): _____

Преподаватель: Рогозин Н. О.

Москва, 2021 г.

SMTP (Simple Mail Transfer Protocol) — это широко используемый сетевой протокол, предназначенный для передачи электронной почты в сетях TCP/IP.

SMTP-клиент - это программа, обеспечивающая отправку электронных писем по SMTP протоколу на SMTP-сервер(ы).

`class email.mime.multipart.MIMEMultipart ()`

Функция создания экземпляра класса MIME. Если сообщение имеет multipart тип контента, то это означает, что оно состоит из нескольких сообщений, и каждое из них определяет свой собственный тип контента. Составные сообщения в Python представлены классом MIMEMultipart.

`MIMEText(_text, _subtype='plain', _charset=None, *, policy=compat32)`

класс MIMEText используется для создания объектов MIME основного типа text. _text — строка для полезной нагрузки. _subtype — второстепенный тип, по умолчанию — plain.

Функция формирования сообщения:

```
def form_message(mail_to, mail_from, mail_text):
    msg = MIMEMultipart()
    msg['From'] = mail_from
    msg['To'] = mail_to
    msg['Subject'] = "Test message"
    msg.attach(MIMEText(mail_text, 'plain'))
    return msg
```

Функция добавления файла:

```
def add_file(msg, filename):
    if (filename != '0'):
        with open(filename, 'rb') as file:
            part = MIMEApplication(file.read(), Name=basename(filename))
            part['Content-Disposition'] = 'attachment; filename="%s"' % basename(filename)
            msg.attach(part)

    return msg
```

Функция отправки сообщения:

```
def send_message(smtphost, login, password, msg):
    server = smtplib.SMTP(smtphost[0], smtphost[1]) #открываем подключение
    server.starttls() #включаем шифрование
    server.login(login, password)
    res = server.sendmail(msg['From'], msg['To'], msg.as_string())
    server.quit() #закрываем подключение
```

Результат работы

1. Отправка сообщений с интервалом 5 секунд, без файла.

```
Message:
  To (mail): networktest1@mail.ru
  From (mail): 1platosha@mail.ru
  Password: [REDACTED]
Time interval: 5
Message text: 5 time
Filename (0 - without file): 0
Message sent to networktest1@mail.ru!
```

1

1platosha@mail.ru

5 time

2. Отправка сообщений с интервалом 3 секунды, с файлом.

```
Message:
  To (mail): networktest1@mail.ru
  From (mail): 1platosha@mail.ru
  Password: [REDACTED]
Time interval: 3
Message text: 3 time
Filename (0 - without file): test.txt
Message sent to networktest1@mail.ru!
Message sent to networktest1@mail.ru!
```

Test message

- 1 1platosha@mail.ru Сегодня, 1:25
Кому: вам



1 файл [Скачать \(7 байт\)](#) [Сохранить в Облако](#)

3 time

Листинг программы

```
end = False
def signal_handling(signum, frame):
    global end
    end = True

def input_args1():
    print("Message:")
    mail_to = input("\tTo (mail): ")
    mail_from = input("\tFrom (mail): ")
    password = input("\tPassword: ")
    return mail_to, mail_from, password

def input_args2():
    time_interval = int(input("Time interval: "))
    mail_text = input("Message text: ")
    filename = input("Filename (0 - without file): ")
    return time_interval, mail_text, filename

def form_message(mail_to, mail_from, mail_text):
    msg = MIME multipart()
    msg['From'] = mail_from
    msg['To'] = mail_to
    msg['Subject'] = "Test message"
    msg.attach(MIMEText(mail_text, 'plain'))
    return msg

def add_file(msg, filename):
    if (filename != '0'):
        with open(filename, 'rb') as file:
            part = MIMEApplication(file.read(), Name=basename(filename))
            part['Content-Disposition'] = 'attachment; filename="%s"' % basename(filename)
            msg.attach(part)

    return msg

def send_message(smtp_host, login, password, msg):
    server = smtplib.SMTP(smtp_host[0], smtp_host[1])
    server.starttls()
    server.login(login, password)
    res = server.sendmail(msg['From'], msg['To'], msg.as_string())
    server.quit()

def main():
    mail_to, mail_from, password = input_args1()
    time_interval, mail_text, filename = input_args2()

    msg = form_message(mail_to, mail_from, mail_text)
    msg = add_file(msg, filename)

    smtp_host = ["smtp.mail.ru", 25]

    signal.signal(signal.SIGINT, signal_handling)
    while (not end):
        send_message(smtp_host, mail_from, password, msg);
        print("Message sent to %s!" % (mail_to))
        time.sleep(time_interval)

if __name__ == '__main__':
    main()
```