|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **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**  **Вариант № 3**  **Студент \_Брянская Е.В.\_\_\_\_\_\_\_\_**  **Группа \_ИУ7-72Б\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Оценка (баллы) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Преподаватель \_\_\_\_Рогозин Н.О.** |  |

Москва.

2021 г.

**Задание**

Написать smtp-клиент, который

1. В качестве входных данных (аргументы командной строки) получает:

адрес получателя, адрес отправителя, пароль.

1. Использует **один из открытых** smtp-серверов для доставки MIME-сообщений, включая приложения, если они есть, в соответствии с вариантом.
2. Дополнительная задача зависит от варианта.

I. **Доставка сообщений выполняется с регулярным интервалом. Интервал и тело сообщения, имя файла для прикрепления (опционально) вводятся с клавиатуры.**

II. В качестве дополнительного параметра задается ключевое слово. По данному ключевому слову выполняется поиск в текстовых файлах в папке клиента. При обнаружении слова файл прикрепляется к письму.

III. Доставка сообщений выполняется в определенное время. Время и тело сообщения, имя файла для прикрепления (опционально) вводятся с клавиатуры.

Задача решается в несколько этапов.

1. Получение данных от пользователя

def input\_data(data):  
 print("--- Input data for sending mail ---")  
 data["toUser"] = input("To: ")  
 data["fromUser"] = input("From: ")  
 data["password"] = input("Password: ")  
 data["interval"] = input("Time interval: ")  
  
 print("\n--- Mail ---")  
 data["title"] = input("Title: ")  
 data["body"] = input("Body: ")  
 data["filename"] = input("Filename (optional): ")

1. Формирование письма

def create\_msg(data):  
 msg = MIMEMultipart()  
 msg["From"] = data['fromUser']  
 msg["To"] = data['toUser']  
 msg["Date"] = formatdate(localtime=True)  
 msg["Subject"] = data['title']  
 msg.attach(MIMEText(data['body']))  
  
 attachment = MIMEBase('application', "octet-stream")  
  
 if data["filename"] != '':  
 try:  
 with open(data["filename"], "rb") as f:  
 tempData = f.read()  
  
 attachment.set\_payload(tempData)  
 encoders.encode\_base64(attachment)  
 attachment.add\_header('Content-Disposition', 'attachment; filename="%s"' % data["filename"])  
  
 msg.attach(attachment)  
 except Exception:  
 print("Error in open() file %s" % data["filename"])  
  
 return msg

1. Отправка

def send\_mail(msg, data):  
 smtp\_ssl\_host = 'smtp.yandex.ru'  
 smtp\_ssl\_port = 465  
  
 while True:  
 try:  
 server\_ssl = smtplib.SMTP\_SSL(smtp\_ssl\_host, smtp\_ssl\_port)  
 server\_ssl.login(data["fromUser"], data["password"])  
 server\_ssl.sendmail(data["fromUser"], data["toUser"], msg.as\_string())  
 server\_ssl.quit()  
  
 print('Email was send')  
 except Exception as exp:  
 print("Error in sending mail %s" % exp)  
  
 time.sleep(int(data["interval"]))

**Результаты**

Тест 1

|  |  |
| --- | --- |
|  |  |

Тест 2

|  |
| --- |
|  |
|  |

**Полный код программы**

|  |
| --- |
| from email.mime.multipart import MIMEMultipart from email.mime.text import MIMEText from email.mime.base import MIMEBase from email.utils import formatdate from email import encoders  import smtplib import time   def input\_data(data):  print("--- Input data for sending mail ---")  data["toUser"] = input("To: ")  data["fromUser"] = input("From: ")  data["password"] = input("Password: ")  data["interval"] = input("Time interval: ")   print("\n--- Mail ---")  data["title"] = input("Title: ")  data["body"] = input("Body: ")  data["filename"] = input("Filename (optional): ")  def create\_msg(data):  msg = MIMEMultipart()  msg["From"] = data['fromUser']  msg["To"] = data['toUser']  msg["Date"] = formatdate(localtime=True)  msg["Subject"] = data['title']  msg.attach(MIMEText(data['body']))   attachment = MIMEBase('application', "octet-stream")   if data["filename"] != '':  try:  with open(data["filename"], "rb") as f:  tempData = f.read()   attachment.set\_payload(tempData)  encoders.encode\_base64(attachment)  attachment.add\_header('Content-Disposition', 'attachment; filename="%s"' % data["filename"])   msg.attach(attachment)  except Exception:  print("Error in open() file %s" % data["filename"])  return msg  def send\_mail(msg, data):  smtp\_ssl\_host = 'smtp.yandex.ru'  smtp\_ssl\_port = 465   while True:  try:  server\_ssl = smtplib.SMTP\_SSL(smtp\_ssl\_host, smtp\_ssl\_port)  server\_ssl.login(data["fromUser"], data["password"])  server\_ssl.sendmail(data["fromUser"], data["toUser"], msg.as\_string())  server\_ssl.quit()   print('Email was send')  except Exception as exp:  print("Error in sending mail %s" % exp)   time.sleep(int(data["interval"]))   if \_\_name\_\_ == "\_\_main\_\_":  data = {}  input\_data(data)  msg = create\_msg(data)  send\_mail(msg, data) |