

Лабораторная работа №6

Тема: «Сетевые приложения на Python»

Требования к выполнению лабораторной работы №6

1. Изучите теоретическую часть к четвёртой лабораторной работе:
 - a. Теоретическая часть к шестой лабораторной работе (notebook_6.ipynb).
 - b. Лекция №6.
2. Создайте новый проект.
3. Запустите примеры из лабораторной работы.
4. Выполните блок «Подготовительные задания».
5. Выполните задание и отправьте решение в ОРИОКС.

Формат защиты лабораторных работ:

1. Продемонстрируйте выполненные задания.
2. Ответьте на вопросы по вашему коду.
3. При необходимости выполните дополнительное (*дополнительные*) задание от преподавателя.
4. Ответьте (*устно*) преподавателю на контрольные вопросы.

Список вопросов

1. Что такое сокет? Для чего используются?
2. Принцип работы метода bind.
3. Принцип работы метода listen.
4. Принцип работы метода accept.
5. Принцип работы метода recv.
6. Принцип работы метода send.
7. Протокол SMTP.
8. Протокол IMAP.

Подготовительные задания

Задание 1. Для выполнения задания вам необходимо создать электронную почту в одном из следующих сервисов:

1. Yandex
2. Gmail
3. Mail.ru

Задание 2. Настройте двухфакторную авторизацию.

Yandex – [инструкция](#)

Gmail – [инструкция](#)

Mail.ru – [инструкция](#)

Задание 3. Создайте пароль приложения.

Yandex – <https://yandex.ru/support/id/ru/authorization/app-passwords>

Gmail – <https://support.google.com/mail/answer/185833?hl=ru>

Mail.ru – <https://help.mail.ru/mail/security/protection/external/>

Задание 4. Настройте работу почтового сервера для работы с приложением.

Yandex – <https://yandex.ru/support/mail/mail-clients/others.html>

Gmail – <https://support.google.com/a/answer/9003945?hl=ru>

Mail.ru – <https://help.mail.ru/mail/mailler/popsmt/>

Общее задание

Написать приложение **Mailer**

Структура приложения:

Файл	Пояснение
<i>client.py</i>	Скрипт, который принимает от пользователя email-адрес и текст сообщения (<i>ввод данных через клавиатуру</i>), отправляет данную информацию (<i>используя сокеты</i>) на сервер (<i>server.py</i>). Если сервер отвечает «ОК», то скрипт завершает работу, иначе выводит пользователю информацию об ошибке и предлагает ввести данные заново.
<i>server.py</i>	<p>Скрипт, который «слушает» сообщения от <i>client.py</i>. При получении <i>email пользователя</i> и текста сообщения проводит валидацию входных данных и в случае ошибки отправляет информацию об ошибке в <i>client.py</i>.</p> <p>Если данные корректны, то скрипт отправляет текст сообщения на email пользователя (<i>отправитель – email администратора</i>) и email администратора.</p> <p>Заголовок письма в формате: «[Ticket #12345] Mailer», где 12345 – уникальный ID, формируемый скриптом <i>server.py</i>.</p> <p>В случае ошибки возвращает <i>client.py</i> соответствующую информацию, иначе отправляет «ОК».</p>
<i>success_request.log</i>	Лог-файл, в котором хранится информация об успешно обработанных сообщениях от <i>collector.py</i>
<i>error_request.log</i>	Лог-файл, в котором хранится информация об ошибочных сообщениях от <i>collector.py</i>
<i>collector.py</i>	Скрипт, который с заданной частотой проверяет новые сообщения, поступающие на email администратора. Если тема письма имеет формат «[Ticket #12345] Mailer», то дописывает в файл <i>success_request.log</i> ID и текст полученного сообщения, иначе дописывает в файл <i>error_request.log</i> текст сообщения.

Файл	Пояснение
<i>config.ini</i> (самостоятельно заполнить из примера)	<p>Структурированный текстовый файл:</p> <p>https://docs.python.org/3/library/configparser.html#supported-ini-file-structure</p> <p>В файле должна храниться следующая информация:</p> <ol style="list-style-type: none"> 1. Email логин администратора 2. Email пароль администратора 3. Хост почтового сервера (чтение сообщений) 4. Порт почтового сервера (чтение сообщений) 5. Хост почтового сервера (отправка сообщений) 6. Порт почтового сервера (отправка сообщений) 7. Частота проверки новых сообщений (в секундах) <p>Пример:</p> <pre>[EMAIL] EMAIL_LOGIN=example@ya.ru EMAIL_PASSWORD=my_password IMAP_HOST=imap.yandex.ru IMAP_PORT=993 SMTP_HOST=smtp.yandex.ru SMTP_PORT=465 PERIOD_CHECK=60</pre>