## Низкоуровневая работа с веб

### Цель работы

Освоить основные навыки обращения c Web из программы на Python, средства парсинга веб-страниц, соответствующие библиотеки.

### Задания для выполнения

1. Написать простейший веб-клиент.
2. Программа должна принимать на вход URL-адрес веб-страницы.
3. Программа делает запрос к серверу средствами библиотеки urllib и сохраняет полученную веб-страницу как текстовый файл в локальную директорию.
4. Модифицируйте программу таким образом, чтобы скачивание производилось в отдельном потоке.

### Дополнительные задания

1. Модифицируйте вашу программу так, чтобы при скачивании HTML документа программа распознавала все ссылки на другие файлы и скачивала их тоже.
2. (\*) Реализуйте простейший текстовый веб-браузер.
3. Создайте робота-скачивателя веб-сайтов с настраиваемой глубиной скачки по ссылкам.
4. Реализуйте простейший веб-сервер. Ваша программа должна принимать входящие соединения и выдавать запрошенные пользователем файлы из домашнего каталога веб-сервера.
   1. Основные требования:
      1. Клиент при установке соединения сразу передает название файла (возможно с подпапками).
      2. Сервер ищет данный файл в домашней папке и передает в сокет его содержимое, если файл найден.
      3. Если файл не найден, сервер передает в сокет специальный код ошибки - 404.
      4. Сервер должен работать в многопоточном режиме.
      5. Сервер должен вести логи в следующем формате: Дата запроса. IP-адрес клиента, имя запрошенного файла, код ошибки.
   2. Дополнительные задания:
      1. Модифицируйте ваш сервер так, чтобы он поддерживал запросы по протоколу HTTP. Используйте минимальное возможное подмножество протокола.
      2. Добавьте возможность запрашивать только определенные типы файлов (.html, .css, .gif и так далее). При запросе неразрешенного типа, верните ошибку 403