*Цель работы*: изучение протокола HTTP и структуры сообщений запросов и ответов взаимодействующих клиентских и серверных Web‑приложений.

**Постановка задачи**

**Вариант 1.**

Создать Web‑сервер, способный принимать на уровне транспортного протокола TCP[[1]](#footnote-1) HTTP‑запросы от одного или одновременно от нескольких клиентов, формировать HTTP‑ответы и отсылать их по протоколу TCP обратно клиентам.

Сервер должен:

1. Принимать GET запросы, выделять в строке заголовка запроса URI и формировать ответ, содержащий в теле HTTP‑сообщения объект в не зашифрованном виде, идентифицированный по URI.
2. В передаваемом ответе заполнять поля Server, Location, Content-Length, Connection и Content-Type.
3. Если запрос успешно обработан, то в строке состояния передавать код 200, иначе возвращать код ошибки 400, 404, 405, 501, 505 в соответствующих ошибкам случаях.
4. Отвечать на запросы OPTIONS, в том числе, если в них Requrest‑URI равно "\*".

Правильность работы Web‑сервера проверить клиентскими приложениями Opera, Firefox, Chrome и разработанными программами по варианту 2 данной лабораторной работы.

**Вариант 2.**

Создать Web‑клиент, способный формировать HTTP‑запрос, отсылать его серверу по протоколу транспортного уровня TCP1 и обрабатывать ответ сервера, полученный по протоколу TCP, после чего возможно формирование следующего запроса без перезапуска клиента.

Клиент должен:

1. Формировать HTTP‑запрос по методу GET, HEAD, TRACE или OPTIONS в зависимости от выбора пользователя.
2. Передавать в качестве Request‑URI абсолютный путь на сервере, а адрес самого сервера передавать в поле Host, при этом должна производиться валидация передаваемого пути и адреса с точки зрения правильности синтаксиса.
3. Передавать заполненными поля заголовка запроса Accept, User-Agent, Accept-Language, Accept-Charset (предоставить пользователю возможность самостоятельного заполнения полей или выбора из нескольких возможных вариантов значений).
4. Анализировать полученный от сервера ответ, осуществляя вывод отдельно версии протокола, кода статуса, поясняющей фразы, каждого из полей заголовка и полученного тела сообщения.
5. Анализировать полученный объект, выводя текстовые документы и картинки на экран, а остальные объекты предлагая сохранить.

Правильность работы Web‑клиента проверить на 3‑х произвольно выбранных Web‑серверах, находящихся в сети Internet, а также разработанными программами по варианту 1 данной лабораторной работы.

1. При выполнении задания нельзя использовать компоненты и решения, реализующие более высокий уровень модели TCP/IP, чем транспортный. В частности это подразумевает запрет на использование в среде Delphi компонента Indy HTTP, хотя не запрещает использования Indy TCPClient и Indy TCPServer. [↑](#footnote-ref-1)