Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет

информатики и радиоэлектроники»

Факультет информационных технологий и управления

Кафедра информационных технологий автоматизированных систем

Отчет по лабораторной работе №4 по дисциплине

ТЕХНОЛОГИИ ИНТЕРНЕТ-ПРОГРАММИРОВАНИЯ

на тему

«Запросы *HTTP*, параметры *URL* и формы *HTML*»

|  |  |
| --- | --- |
| Выполнил ст. гр. 820601  Проверил преп. каф. ИТАС | А.С. Хасан  А.Л. Гончаревич |

Минск 2022

СОДЕРЖАНИЕ

[Введение 3](#_Toc94186760)

[1 Постановка задачи 4](#_Toc94186761)

[2 Теоритическая часть 5](#_Toc94186762)

[3 Ход работы 6](#_Toc94186763)

[Заключение 8](#_Toc94186764)

# ВВЕДЕНИЕ

Протокол передачи гипертекста (*Hypertext Transfer Protocol* — *HTTP*) —это прикладной протокол для передачи гипертекстовых документов, таких как *HTML*. Он создан для связи между веб-браузерами и веб-серверами, хотя в принципе *HTTP* может использоваться и для других целей. Протокол следует классической клиент-серверной модели, когда клиент открывает соединение для создания запроса, а затем ждёт ответа. *HTTP* — это протокол без сохранения состояния, то есть сервер не сохраняет никаких данных (состояние) между двумя парами «запрос-ответ». Несмотря на то, что *HTTP* основан на *TCP*/*IP*, он также может использовать любой другой протокол транспортного уровня с гарантированной доставкой.

*PHP* (рекурсивный акроним словосочетания *PHP*: *Hypertext Preprocessor*) — это распространённый язык программирования общего назначения с открытым исходным кодом. *PHP* специально сконструирован для веб-разработок и его код может внедряться непосредственно в *HTML*.

# ПОСТАНОВКА ЗАДАЧИ

Ознакомиться с *POST* и *GET* запросами протокола *HTTP*, параметрами передаваемыми через *URL* и *HTML* формами.

# ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

*HTTP* — это протокол, позволяющий получать различные ресурсы, например *HTML*-документы. Протокол *HTTP* лежит в основе обмена данными в Интернете. *HTTP* является протоколом клиент-серверного взаимодействия, что означает инициирование запросов к серверу самим получателем, обычно веб-браузером (*web-browser*). Полученный итоговый документ может состоять из различных поддокументов, являющихся частью итогового документа: например, из отдельно полученного текста, описания структуры документа, изображений, видео-файлов, скриптов и многого другого.

Запрос *GET* передает данные в *URL* в виде пар «имя-значение» (другими словами, через ссылку), а запрос *POST* передает данные в теле запроса. Это различие определяет свойства методов и ситуации, подходящие для использования того или иного *HTTP* метода.

Страница, созданная методом *GET*, может быть открыта повторно множество раз. Такая страница может быть кэширована браузерами, проиндексирована поисковыми системами и добавлена в закладки пользователем. Из этого следует, что метод *GET* следует использовать для получения данных от сервера и не желательно в запросах, предполагающих внесений изменений в ресурс.

Запрос, выполненный методом *POST*, напротив следует использовать в случаях, когда нужно вносить изменение в ресурс (выполнить авторизацию, отправить форму оформления заказа, форму обратной связи, форму онлайн заявки). Повторный переход по конечной ссылке не вызовет повторную обработку запроса, так как не будет содержать переданных ранее параметров. Метод *POST* имеет большую степень защиты данных, чем *GET*: параметры запроса не видны пользователю без использования специального ПО, что дает методу преимущество при пересылке конфиденциальных данных, например в формах авторизации.

В *PHP* параметры, переданные через *URL*, хранятся в системной переменной $\_*GET*. Она представляет собой ассоциативный массив (или, по-другому, словарь). Ассоциативный массив (или словарь) – это такая структура данных, которая содержит пары ключ-значение. Ключи словаря $\_*GET* – это имена параметров. По аналогии с методом *GET*, параметры, переданные методом *POST*, содержатся в системной переменной $\_*POST*. Это также, как и $\_*GET*, ассоциативный массив. Ключами в массиве $\_*POST* являются значения атрибутов *name* у элементов на форме.

# 3 ХОД РАБОТЫ

Работа выполняется с помощью редактора кода – *Visual Studio Code*. Работа с программой начинается с создания *PHP* файла, в который будут помещаться скрипты.

В задании 4.1 для начала необходимо создать папку с изображениями и две страницы *index.php* и *photo.php*. На странице *photo.php* необходимо отобразить определённую картинку. Чтобы понять, какую именно картинку показывать, нужно считать определённый параметр – номер изображения – из *URL*-адреса. Страница *photo.php* показана на рисунке 3.1.

**

Рисунок 3.1 – Страница *photo.php*

На странице *index.php* находятся ссылки на все изображения. Страница *index.php* показана на рисунке 3.2.

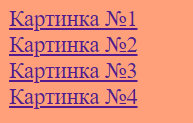
**

Рисунок 3.2 – Страница *index.php*

На странице *photo.php* запросы отправлялись *GET* методом. На следующей странице *sum.php* используется *POST* запрос. На этой странице реализован скрипт складывающий числа. Страница sum.php показана на рисунке 3.3.



Рисунок 3.3 – Страница *sum.php*

В задании 3.2 необходимо превратить получившийся сумматор из предыдущего задания в калькулятор с четырьмя операциями: сложение, вычитание, умножение, деление. Результат выполнения задания продемонстрирован на рисунке 3.4.



Рисунок 3.4 – Задание 3.2

В задании 3.3 необходимо создать калькулятор, который будет определять тип выбранной пользователем операции, ориентируясь на нажатую кнопку. Данные, введённые пользователем в поля, должны сохраняться и выводиться вместе с результатом вычисления.

Результат выполнения задания продемонстрирован на рисунке 3.*5*.

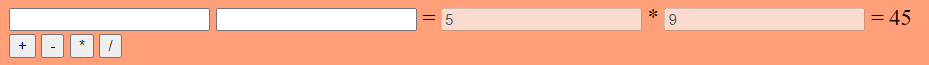


Рисунок 3.5 – Задание 3.3

# ЗАКЛЮЧЕНИЕ

*HTTP* — лёгкий в использовании расширяемый протокол. Структура клиент-сервера, вместе со способностью к простому добавлению заголовков, позволяет *HTTP* протоколу продвигаться вместе с расширяющимися возможностями Сети.

*PHP* позволяет создавать качественные *web*-приложения за очень короткие сроки, получая продукты, легко модифицируемые и поддерживаемые в будущем.

*PHP* прост для освоения, и вместе с тем способен удовлетворить запросы профессиональных программистов.

Язык *PHP* постоянно совершенствуется, и ему наверняка обеспечено долгое доминирование в области языков *web*-программирования, по крайней мере, в ближайшее время.

В ходе выполнения лабораторной работы я изучил *GET* и *POST* методы протокола *HTTP*, изучил как передавать параметры через *URL* и *HTML* форму.