Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет информационных технологий и управления

Кафедра ИТАС

Отчет

по лабораторной работе №5

«*Cookies* и сессии *PHP*»

Вариант №17

|  |  |
| --- | --- |
| Выполнил:  ст. гр. 820603  Лаходанов А. Е. | Проверил:  А. Л. Гончаревич |

Минск 2022

# Цель работы

Изучение основных возможностей объектно-ориентированного языка *PHP* в работе с *cookies* и сессиями *PHP.*

# Проектирование

Для каждого задания из лабораторной работы №5 создам программы, содержащие их выполнение.

# Теоретические сведения

*PHP –* это язык программирования, специально разработанный для написания веб-приложений (сценариев) исполняющихся на стороне сервера.

Важным преимуществом языка *PHP* перед языками *Perl* и *C* является возможность создания *HTML*-документов с внедренными командами *PHP*. Конструкции *PHP*, вставленные в *HTML*-текст, выполняются сервером при каждом посещении страницы. Результат их обработки вместе с обычным *HTML*-текстом передается браузеру.

Значительным отличием *PHP* от других скриптов, выполняющихся на стороне клиента (например, *JavaScript*), является то, что скрипты *PHP* выполняются на стороне сервера. Можно сконфигурировать сервер так, чтобы *HTML*-документы обрабатывались процессором *PHP* таким образом, что пользователи даже не смогут определить, получают они обычную *HTML*-страницу либо результат выполнения кода *PHP*.

В *PHP* используется четыре основных (простейших) типа данных (литералов) и соответствующих им переменных: целые числа, числа с плавающей точкой, строчки, булевские значения.

Переменная – это имя (или идентификатор), присваиваемое ячейке памяти компьютера, которая хранит определенные литералы во время исполнения сценария *PHP*.

Если необходимо отобразить значение, присвоенное переменной, достаточно сослаться на нее c помощью оператора*echo*.

Строка – это набор любых символов. В *PHP* существует два способа задания строк в двойных, либо одинарных кавычках. Разница между ними заключается в том, что если внутри строки первого типа написать имя переменной, то интерпретатор *PHP* подставит значение этой переменной при результирующем выводе. В случае же со вторым типом строк (одинарные кавычки) подстановки не происходит.

Группу операторов составляют операции сравнения: «меньше» (<), «меньше или равно» (<=), «больше» (>) и «больше или равно» (>=). Все они сравнивают два заданных значения и возвращают логическое значение *true* или *false*.

Логические операторы проверяют булевы условия. Существует четыре главных булевых условия: «И» (*and* или &&), «ИЛИ» (*or* или ||), «НЕ» (!) и «исключающее ИЛИ» (*хоr или ^*).

Операция «И» возвращает истину только в том случае, если истинны оба выражения.

Операция «**ИЛИ»** возвращает истину в том случае, если истинно хотя бы одно выражение.

Операция «НЕ» переворачивает значение выражения.

Операция исключающее «ИЛИ» обозначается как *xor*. Он возвращает истину, если один и только один из операндов имеет истинное значение. Если оба операнда имеют истинное значение, оператор вернет ложное значение.

Ветвление – это выполнение определённой команды (набора команд) только при условии истинности некоторого логического выражения. По сути, ветвление – это разделение программы на варианты в зависимости от определённых условий.

Оператор *if* в *PHP* (как и *elseif*) используется для выполнения блока кода только в том случае, если указанное условие имеет значение *true*.

Оператор *if … else* позволяет выполнить один блок кода, если указанное условие оценивается как *true*, а другой блок кода, если *false*.

Оператор *switch* сравнивает значение условного выражения с несколькими значениями. Как правило, в качестве выражения используется переменная, в зависимости от значения которой должен быть исполнен тот или иной блок кода. Для сравнения в *switch* используется оператор равенства (==). Оператор *switch* берет значение, возвращаемое условным выражением, и начинает сравнивать его со всеми значениями, стоящими рядом с ключевым словом case (метка), в порядке их следования. Как только совпадение обнаружено, выполняется соответствующий блок кода. Если соответствие не найдено, не исполняется ни один из блоков. Если должен быть выполнен только один блок кода, соответствующий определенному значению, то в конце этого блока следует вставить ключевое слово *break*.

Функция – это блок кода, к которому можно обращаться из разных частей скрипта. Функции могут иметь входные и выходные параметры. Входные параметры могут использоваться в операциях, которые содержит функция. Выходные параметры устанавливаются функцией, а их значения используются после выполнения функции. Функция в *PHP* объявляется с помощью ключевого слова *function*. После указания параметров в фигурных скобках следует тело функции.

Функция может возвращать некоторое значение – число, строку и т.д., то есть некоторый результат. Для возвращения значения в функции применяется оператор *return*, после которого указывается возвращаемое значение.

Цикл – это конструкция языка, которая позволяет выполнить блок кода больше одного раза.

Цикл с предусловием *while* вычисляет значение логического выражения. Если оно истинно, выполняется тело цикла, иначе переходит в следующий за телом цикла оператор.

Цикл с постусловием *do…while* в отличие от цикла *while*, проверяет значение выражения не до, а после каждого прохода (итерации). Таким образом, тело цикла выполняется хотя бы один раз.

Цикл со счетчиком *for* используется для выполнения тела цикла определенное число раз. С помощью цикла *for* можно создавать конструкции, которые будут выполнять действия совсем не такие тривиальные, как простая переборка значения счетчика.

В *PHP4* появился еще один специальный тип цикла – *foreach*. Данный цикл предназначен специально для перебора массивов. Цикл *foreach* оперирует не исходным массивом, а его копией. Это означает, что любые изменения, которые вносятся в массив, не могут быть «видны» из тела цикла. Что позволяет, например, в качестве массива использовать не только переменную, но и результат работы какой-нибудь функции, возвращающей массив.

Массив – это ещё один тип данных, вроде числа или строки. Главное отличие массива от остальных типов данных заключается в его способности хранить в переменной больше одного значения. Внутри массива у каждого значения есть адрес, по которому к нему можно обратиться. Такой адрес называется индексом. Индекс – это просто порядковый номер значения внутри массива. Массив может быть создан языковой конструкцией *array().*

*HTTP* – это протокол передачи данных между сервером и клиентом.

Cуществует два способа, с помощью которых клиенты-браузеры могут отправлять информацию на веб-сервер – это метод *GET* и метод *POST*.

Метод *GET* создает длинную строку, которая отображает в логах сервера и в адресной сторке браузера.

Метод *POST* передает информацию через *HTTP*-заголовки.

*URL* – единообразный локатор (определитель местонахождения) ресурса. *URL* – это стандартизированный способ записи адреса ресурса в сети Интернет.

*Cookies* – это небольшой кусок текстовой информации, которую сервер сохраняет в браузере пользователя.

Для установки куки существует специальная функция, которая называется *setcookie()*. Она может принимать различное количество параметров

Сессия – это возможность сохранять данные при переходе между страницами. Общая идея заключается в следующем: для того, чтобы однозначно идентифицировать клиента, достаточно одной метки, а вся информация теперь хранится на сервере в специальном файле.

Для создания сессии существует специальная функция *session\_start()*.

*Session\_destroy* – Уничтожает все данные сессии

Всю работу по записи и чтению информации из файла PHP выполняет автоматически.

# Ход работы

Страница, которая будет неавторизованных пользователей отправлять на страницу авторизации, а авторизованных на ту страницу, которую они посещали последний раз:

*<?php*

*session\_start();*

*if(!empty($\_SESSION['autorisation']) and $\_SESSION['autorisation']){*

*header("Location: ". $\_COOKIE['lastPage']);*

*}*

*else {*

*//Иначе нужен редирект на страницу авторизации*

*header("Location: autorisation.php");*

*}*

*?>*

Код страницы, используемой для авторизации пользователя:

*<?php*

*header("Content-type: text/html; charset=utf-8");*

*if(isset($\_COOKIE['style']))*

*$s = $\_COOKIE['style'];*

*else*

*$s = 'first';*

*if(isset($\_POST['submit'])){*

*session\_start();*

*$name = $\_POST['name'];*

*setcookie('name', $name);*

*$\_SESSION['autorisation'] = true;*

*if(isset($\_COOKIE['lastPage']))*

*header("Location: ". $\_COOKIE['lastPage']);*

*else*

*header("Location: choose.php");*

*}?>*

*<head><link rel="stylesheet" href=<?php echo "$s.css" ?></head>*

*<h3>Cтраница авторизации</h3>*

*<form action="" method="post">*

*Логин: <input type="text" name="name"><br>*

*<input type="submit" name="submit" value="Войти"><br>*

*</form>*

Результат страницы авторизациипродемонстрирован на рисунке 1.



Рисунок 1 – Авторизация

Страница, на которую осуществляется переход после введения логина нажатия кнопки «Войти»:

*<?php*

*header("Content-type: text/html; charset=utf-8");*

*session\_start();*

*//Запись в cookie адреса страницы*

*$lastPage = $\_SERVER['REQUEST\_URI'];*

*setcookie('lastPage', $lastPage);*

*if(isset($\_COOKIE['style']))*

*$s = $\_COOKIE['style'];*

*else*

*$s = 'first';*

*//Если сессия создана, то поприветствовать пользователя*

*if(!empty($\_SESSION['autorisation']) and $\_SESSION['autorisation'])*

*echo "Добро пожаловать, ". $\_COOKIE['name'] .". Вы авторизованы. <br>";*

*else*

*header("Location: autorisation.php");*

*?>*

*<head><link rel="stylesheet" href=<?php echo "$s.css" ?>></head>*

*<h2>CHOOSE</h2>*

*На страницу <a href="A.php"><"A.php"></a><br><br>*

*На страницу <a href="B.php"><"B.php"></a><br><br>*

Результат страницы перехода продемонстрирован на рисунке 2.

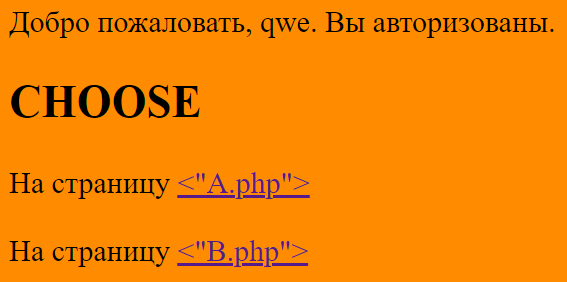


Рисунок 2 – Страница перехода

После выбора файла «*A.php*», осуществляется перенаправление на страницу, как на рисунке 3.

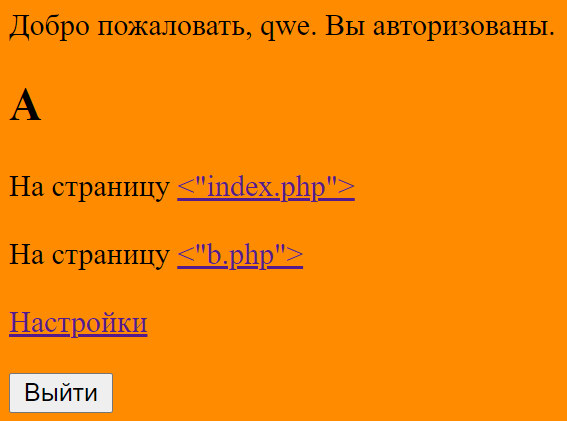


Рисунок 3 – Выбранная страница

Код программы, в которой находится настройка стилей:

*<?php*

*header("Content-type: text/html; charset=utf-8");*

*session\_start();*

*//Если сессия создана, то поприветствовать пользователя*

*if(!empty($\_SESSION['autorisation']) and $\_SESSION['autorisation']){*

*if(isset($\_POST['submit'])) {*

*$s = $\_POST['style'];*

*setcookie('style', $s);*

*}*

*elseif (isset($\_COOKIE['style']))*

*$s = $\_COOKIE['style'];*

*else*

*$s = 'first';*

*}*

*else*

*header("Location: autorisation.php");*

*?>*

*<head>*

*<link rel="stylesheet" href="<?php echo "$s.css"; ?>">*

*</head>*

*<h3>Выбор оформления</h3>*

*<form action="" method="post">*

*<p><input name="style" type="radio" value="first"> Первое</p>*

*<p><input name="style" type="radio" value="second"> Второе</p>*

*<p><input name="style" type="radio" value="third" > Третье</p>*

*<p><input name="submit" type="submit" value="Выбрать"></p>*

*</form>*

*На страницу <a href="A.php"><"A.php"></a><br><br>*

*На страницу <a href="B.php"><"B.php"></a><br><br>*

Результат страницы настроек проиллюстрирован на рисунке 4.

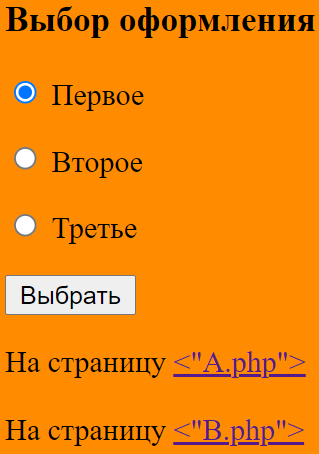


Рисунок 4 – Страница настроек

При выборе оформления и нажатия кнопки «Выбрать», изменяется цвет фона страниц, как на рисунке 5.

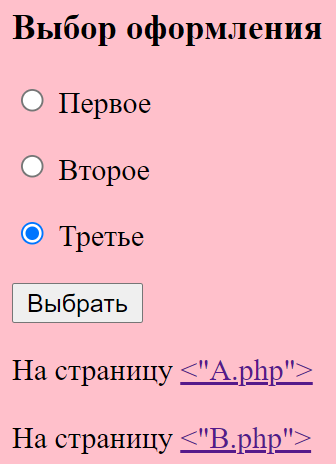


Рисунок 5 – Измененное оформление

Заключение

В результате выполнения лабораторной работы №5 мною разработаны программы, иллюстрирующие выполнение заданий, с использованием такого языка программирования как *PHP*.