Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет информационных технологий и управления

Кафедра ИТАС

Отчет

по лабораторной работе №4

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

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 запросами *HTTP*, параметры *URL* и формы *HTML.*

# Проектирование

Для каждого задания из лабораторной работы №4 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.

# Ход работы

## Работа с изображениями

Страница с ссылками на изображения:

*<html>*

*<head>*

*<title>Галерея изображений</title>*

*</head>*

*<body>*

*<h2>Какое-то описание картинки</h2>*

*<?php*

*for($i = 1; $i <= 4; $i++)*

*echo "<a href = photo.php?id=$i >Картинка №$i</a><br/>";*

*?>*

*</body>*

*</html>*

Результат страницы с картинкамипродемонстрирован на рисунке 1.

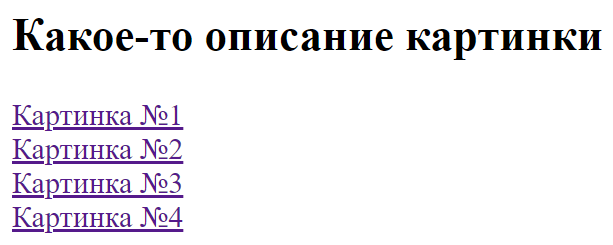


Рисунок 1 – Страница с ссылками на картинки

После нажатия на любую из представленных ссылок, откроется страница с изображением как на рисунке 2.

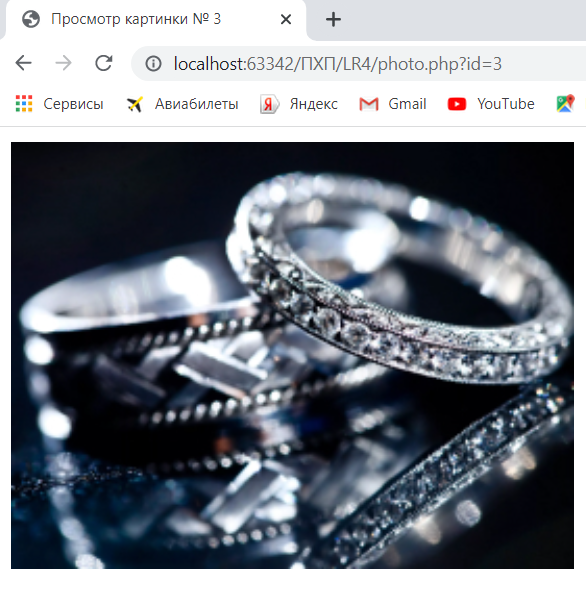


Рисунок 2 – Выбранное изображение

## Добавление калькулятора

Код страницы с формой для сложения введенных чисел:

*<?php*

*if(isset($\_POST['a']) && isset($\_POST['b']))*

*$result = $\_POST['a'] + $\_POST['b']; else*

*$result = "";*

*?>*

*<html>*

*<head>*

*<title>Галерея изображений</title>*

*</head>*

*<body>*

*<form method="post">*

*<input type="text" name="a" />*

*+*

*<input type="text" name="b" />*

*<input type="submit" value="=" />*

*<?php echo $result; ?>*

*</form>*

*</body>*

*</html>*

После запуска программы получен результат как на рисунке 3.

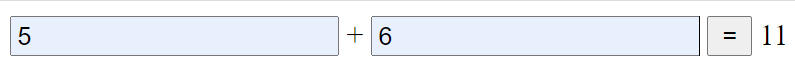


Рисунок 3 – Сложение чисел

Калькулятор, в котором арифметические операции выбираются с помощью выпадающего списка:

*<?PHP*

*if(isset($\_POST['a']) && isset($\_POST['b']) && isset($\_POST["calculator"])) {*

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

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

*$calc = $\_POST["calculator"];*

*if($calc == "+")*

*$result = $a + $b;*

*if($calc == "-")*

*$result = $a - $b;*

*if($calc == "\*")*

*$result = $a \* $b;*

*if($calc == "/") {*

*if($b !== '0') {*

*$result = $a / $b;*

*}*

*else {*

*$result = "Деление на ноль";*

*}*

*}*

*}*

*else $result = "";*

*?>*

*<html>*

*<head>*

*<meta charset="utf-8">*

*</head>*

*<body>*

*<form method="post">*

*<input type="text" name="a"/>*

*<select name="calculator">*

*<option selected value="+">+</option>*

*<option value="-">-</option>*

*<option value="\*">\*</option>*

*<option value="/">/</option>*

*</select>*

*<input type="text" name="b"/>*

*<input type="submit" value="="/>*

*<?php echo $result; ?>*

*</form>*

*</body>*

*</html>*

Результат выполнения программы показан на рисунке 4.

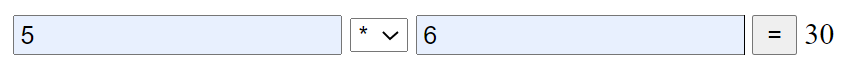


Рисунок 4 – Калькулятор с выбором из списка арифметических операций

Страница с калькулятором, в котором тип выбранной арифметической операции выбирается нажатием на кнопку:

*<?PHP*

*$result = '';*

*$a = isset($\_POST['a']) ? $\_POST['a'] : '';*

*$b = isset($\_POST['b']) ? $\_POST['b'] : '';*

*$calc = isset($\_POST['calc']) ? $\_POST['calc'] : '';*

*if($calc == "+")*

*$result = $a + $b;*

*if($calc == "-")*

*$result = $a - $b;*

*if($calc == "\*")*

*$result = $a \* $b;*

*if($calc == "/") {*

*if($b !== '0') {*

*$result = $a / $b;*

*}*

*else {*

*$result = "Деление на ноль";*

*}*

*}*

*?>*

*<html>*

*<head>*

*<meta charset="utf-8">*

*</head>*

*<body>*

*<form method="post">*

*<input type="text" name="a" value="<?php echo $a ?>"/>*

*<span id = "op"><?php echo $calc; ?></span>*

*<input type="text" name="b" value="<?php echo $b ?>"/>*

*=*

*<?php echo "<b>$result<b>"; ?>*

*<br>*

*<button name="calc" value="+">+</button>*

*<button name="calc" value="-">-</button>*

*<button name="calc" value="\*">\*</button>*

*<button name="calc" value="/">/</button>*

*</form>*

*</body>*

*</html>*

После запуска программы получен результат как на рисунке 5.

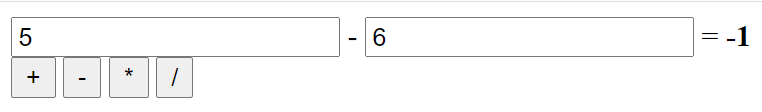


Рисунок 5 – Калькулятор с выбором арифметической операции нажатием на кнопку

Заключение

В результате выполнения лабораторной работы №4 мною разработаны программы, иллюстрирующие выполнение заданий, с использованием такого языка программирования как *PHP*.