Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет информационных технологий и управления

Кафедра ИТАС

Отчет

по лабораторной работе №2

«Массивы и циклы»

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 массивами и циклами*.*

# Проектирование

Для каждого задания из лабораторной работы №3 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.

Внутри блока кода функций можно пользоваться абсолютно любыми средствами языка *PHP* также, как и вне функций. В том числе, существует возможность вызова функцией самой себя. Эта операция называется рекурсией.

Функция *date()* выводит текущие дату и время в заданном формате.

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().*

Конструкция *mb\_substr()* Возвращает не более *length* символов, начиная с позиции *start* исходной строки *str*.

Функция *strtr* – преобразует заданные символы или заменяет подстроки.

Оператор «=>» позволяет сопоставить ключ с определенным значением.

# Ход работы

## Использование цикла *while*

Программа, в которой при помощи цикла *while* выводятся числа числа в промежутке от 0 до 100, которые делятся на 3 без остатка:

*<?php*

*$n = 100;*

*$i = 0;*

*while ($i < $n)*

*{*

*if ($i % 3 == 0)*

*echo "$i<br/>";*

*$i++;*

*}*

*?>*

Результат вывода чиселпродемонстрирован на рисунке 1.

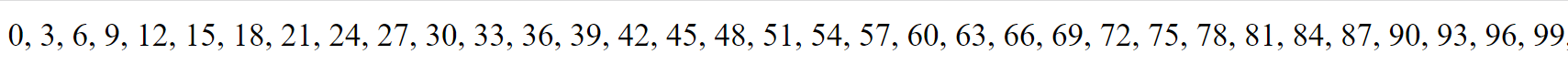


Рисунок 1 – Вывод последовательности чисел

## Работа с циклом *do…while*

Функция вывода чисел от 0 до 10:

*<?php*

*$n = 10;*

*$i = 0;*

*do*

*{*

*if ($i == 0)*

*echo "$i - это ноль <br/>";*

*elseif ($i % 2 == 0)*

*echo "$i - это четное число <br/>";*

*else*

*echo "$i - это нечетное число <br/>";*

*$i++;*

*} while ($i <= $n)*

*?>*

После запуска программы получен результат как на рисунке 2.

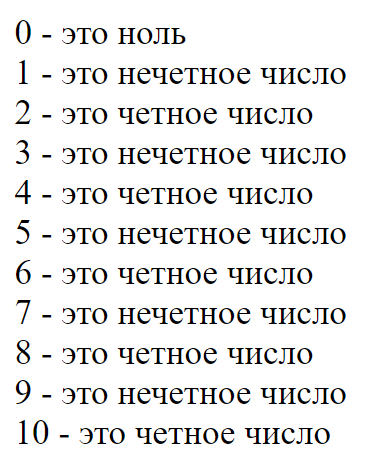


Рисунок 2 – Числа от 0 до 10

## Использование цикла *for*

Код страницы, в которой реализован вывод чисел от 0 до 9 без использования тела цикла *for*:

*<?php*

*for($i = 0; print "$i</br>", $i++ < 9;)*

*?>*

Результат вывода последовательности проиллюстрирован на рисунке 3.

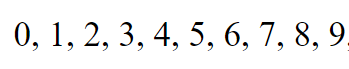


Рисунок 3 – Последовательность чисел

## Работа с массивами

Массив, где в качестве ключей будут использоваться названия областей, а в качестве значений – массивы с названиями городов из соответствующей области:

*<?php*

*$ob = array(*

*"Московская область: "*

*=> array("Москва", "Зеленоград", "Клин") ,*

*"Ленинградская область: " =>*

*array("Санкт-Петербург", "Всеволожск", "Павловск", "Кронштадт"),*

*"Рязанская область :" =>*

*array("Рязань", "Касимов", "Михайлов", "Ряжск", "Скопин"));*

*foreach ($ob as $naz => $items)*

*{*

*echo "</br>*

*$naz*

*</br>";*

*foreach ($items as $value)*

*{*

*echo "$value*

*<br>";*

*}*

*}*

*?>*

После запуска программы получен результат как на рисунке 4.

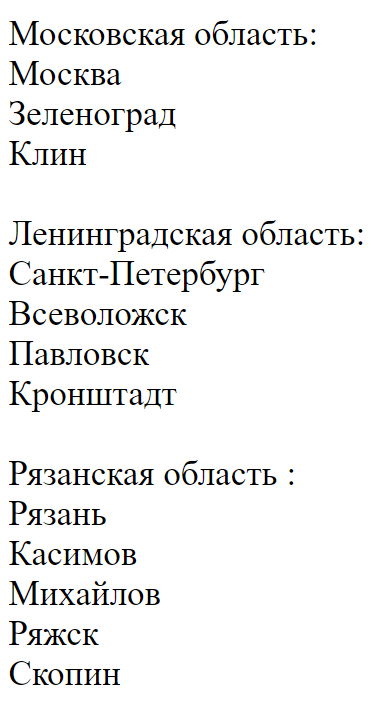


Рисунок 4 – Массив областей с городами

Код программы, выводящей на экран города, начинающиеся на букву «К»:

*<?php*

*$ob = array(*

*"Московская область: "=> array("Москва", "Зеленоград", "Клин", "Красногорск") ,*

*"Ленинградская область: " => array("Санкт-Петербург", "Всеволожск", "Павловск", "Кронштадт"),*

*"Рязанская область :" => array("Рязань", "Касимов", "Михайлов", "Ряжск", "Скопин"));*

*foreach ($ob as $naz => $items)*

*{*

*echo "</br>$naz </br>";*

*foreach ($items as $value)*

*{*

*if(mb\_substr($value, 0, 1) == 'К')*

*echo "$value <br>";*

*}*

*}*

*?>*

Результат вывода городов продемонстрирован на рисунке 5.

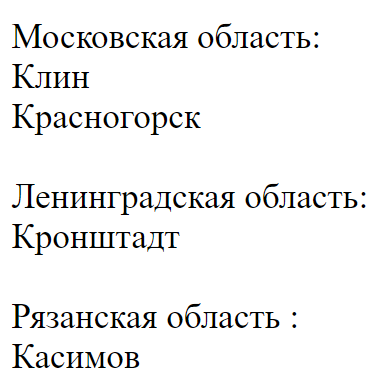


Рисунок 5 – Города на букву «К»

Программа, заменяющая буквы русского алфавита латинскими:

*<?php*

*function translate ($text) {*

*$letters = array(*

*"а"=>"a",*

*"б"=>"b",*

*"в"=>"v",*

*"г"=>"g",*

*"д"=>"d",*

*"е"=>"e",*

*"ё"=>"e",*

*"ж"=>"zh",*

*"з"=>"z",*

*"и"=>"i",*

*"й"=>"i",*

*"к"=>"k",*

*"л"=>"l",*

*"м"=>"m",*

*"н"=>"n",*

*"о"=>"o",*

*"п"=>"p",*

*"р"=>"r",*

*"с"=>"s",*

*"т"=>"t",*

*"у"=>"u",*

*"ф"=>"f",*

*"х"=>"kh",*

*"ц"=>"tc",*

*"ч"=>"ch",*

*"ш"=>"sh",*

*"щ"=>"shch",*

*"ы"=>"y",*

*"э"=>"e",*

*"ю"=>"iu",*

*"я"=>"ia",*

*"ь"=>""*

*);*

*return strtr($text, $letters);*

*}*

*echo translate ("Тестовый текст");*

*?>*

После запуска программы, на экран выводится результат как на рисунке 6.

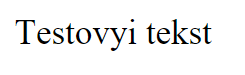


Рисунок 6 – Замена букв в тексте

Код функции, заменяющий в строке пробелы на подчеркивания:

*<?php*

*function test ($str) {*

*$text = array (*

*" " => "\_",*

*);*

*return strtr($str, $text);*

*}*

*echo test("раз два три");*

*?>*

Результат замены проиллюстрирован на рисунке 7.

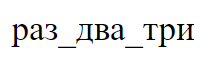


Рисунок 7 – Замена пробелов на подчеркивания

Код страницы с функцией, получающей строку на русском языке и производящей транслитерацию с заменой пробелов на подчеркивания:

*<?php*

*function translate ($text) {*

*$letters = array(*

*"а"=>"a",*

*"б"=>"b",*

*"в"=>"v",*

*"г"=>"g",*

*"д"=>"d",*

*"е"=>"e",*

*"ё"=>"e",*

*"ж"=>"zh",*

*"з"=>"z",*

*"и"=>"i",*

*"й"=>"i",*

*"к"=>"k",*

*"л"=>"l",*

*"м"=>"m",*

*"н"=>"n",*

*"о"=>"o",*

*"п"=>"p",*

*"р"=>"r",*

*"с"=>"s",*

*"т"=>"t",*

*"у"=>"u",*

*"ф"=>"f",*

*"х"=>"kh",*

*"ц"=>"tc",*

*"ч"=>"ch",*

*"ш"=>"sh",*

*"щ"=>"shch",*

*"ы"=>"y",*

*"э"=>"e",*

*"ю"=>"iu",*

*"я"=>"ia",*

*" " => "\_"*

*);*

*return strtr($text, $letters);*

*}*

*echo translate ("Тестовый текст тут");*

*?>*

После запуска программы получен результат как на рисунке 8.

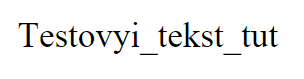


Рисунок 8 – Транслитерация с заменой пробелов

Заключение

В результате выполнения лабораторной работы №3 мною разработаны программы, иллюстрирующие выполнение заданий, с использованием такого языка программирования как *PHP*.