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 №3 по дисциплине

ТЕХНОЛОГИИ ИНТЕРНЕТ-ПРОГРАММИРОВАНИЯ

на тему

«Циклы и массивы»

|  |  |
| --- | --- |
| 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 7](#_Toc94186763)

[Заключение 11](#_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 циклами и массивами в языке *PHP*. Изучить основные функции для работы с массивами.

# ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Следующими по частоте использования, после конструкций условий (условных операторов), находятся циклы. Циклы позволяют повторять определенное (и даже неопределенное, когда работа цикла зависит от условия) количество раз различные операторы. Данные операторы называются телом цикла. Проход цикла называется итерацией. *PHP* поддерживает три вида циклов: *while*, *do*..*while*, *for*, *foreach*.

При использовании циклов есть возможность использования операторов *break* и *continue*. *Break* прерывает работу всего цикла, а *continue* только текущей итерации.

Циклы *while* являются простейшим видом циклов в *PHP*. Простейшей формой цикла *while* является следующее выражение:

*while (expr)*

*{*

*statement*

*{*

Смысл выражения *while* очень прост. Оно указывает *PHP* выполнять вложенные выражения повторно до тех пор, пока выражение в самом while является *true*. Значение выражения *expr* проверяется каждый раз перед началом цикла, поэтому даже если значение выражения изменится в процессе выполнения вложенных выражений в цикле, выполнение не прекратится до конца итерации (каждый раз, когда *PHP* выполняет выражения в цикле – это одна итерация). Если выражение *while* равно *false* с самого начала, вложенные выражения ни разу не будут выполнены.

Цикл *do-while* очень похож на цикл *while*, с тем отличием, что истинность выражения проверяется в конце итерации, а не в начале. Главное отличие от обычного цикла *while* в том, что начальная итерация цикла *do-while* гарантированно выполнится (истинность выражения проверяется в конце итерации), тогда как она может не выполниться в обычном цикле *while* (истинность выражения которого проверяется в начале выполнения каждой итерации, и если изначально имеет значение *false*, то выполнение цикла будет прервано сразу).

Цикл *for* самый сложный цикл в *PHP*. Синтаксис цикла *for* следующий:

*for (expr1; expr2; expr3){*

*statement*

*}*

Выражение *expr*1 всегда вычисляется (выполняется) только один раз в начале цикла. В начале каждой итерации оценивается выражение *expr*2. Если оно принимает значение *true*, то цикл продолжается и выполняются вложенные операторы. Если оно принимает значение *false*, выполнение цикла заканчивается. В конце каждой итерации выражение *expr*3 вычисляется (выполняется). Каждое из выражений может быть пустым или содержать несколько выражений, разделённых запятыми. В *expr*2 все выражения, разделённые запятыми, вычисляются, но результат берётся из последнего. Если выражение *expr*2 отсутствует, это означает, что цикл будет выполняться бесконечно.

Конструкция *foreach* предоставляет простой способ перебора массивов. *foreach* работает только с массивами и объектами, и будет генерировать ошибку при попытке использования с переменными других типов или неинициализированными переменными. Существует два вида синтаксиса:

*foreach (iterable\_expression as $value)*

*statement*

*foreach (iterable\_expression as $key => $value)*

*statement*

На самом деле массив в PHP – это упорядоченное отображение, которое устанавливает соответствие между значением и ключом. Этот тип оптимизирован в нескольких направлениях, поэтому можно использовать его как собственно массив, список (вектор), хеш-таблицу (являющуюся реализацией карты), словарь, коллекцию, стек, очередь. Так как значением массива может быть другой массив PHP, можно также создавать деревья и многомерные массивы.

Массив (тип *array*) может быть создан языковой конструкцией *array*(). В качестве параметров она принимает любое количество разделённых запятыми пар *key* => *value* (ключ => значение).

*array(*

*key => value,*

*key2 => value2,*

*key3 => value3,*

*...*

*)*

# 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 3.1 необходимо было с помощью цикла *while* вывести все числа в интервале [0..100], которые делятся на три без остатка. Результат выполнения задания показан на рисунке 3.1.

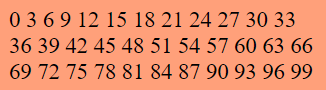
**

Рисунок 3.1 – Задание 3.1

В задании 3.2 необходимо было c помощью цикла *do*…*while* написать функцию для вывода чисел от нуля до 10. Результат выполнения задания продемонстрирован на рисунке 3.2.

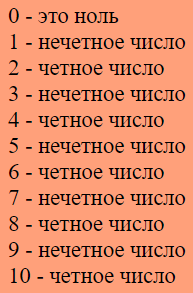


Рисунок 3.2 – Задание 3.2

В задании 3.3 необходимо было вывести с помощью цикла *for* числа от нуля до девяти, не используя тело цикла.

Скрипт задания:

*<?php*

*for ($i=0; $i < 10; print $i++){}*

*?>*

Результат выполнения задания продемонстрирован на рисунке 3.3.



Рисунок 3.3 – Задание 3.3

В задании 3.4 необходимо объявить массив, где в качестве ключей будут использоваться названия областей, а в качестве значений – массивы с названиями городов из соответствующей области. Вывести в цикле значения массива.

Результат выполнения задания показан на рисунке 3.4.

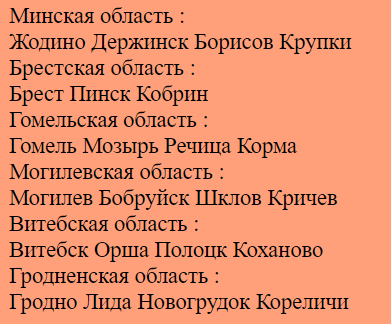


Рисунок 3.4 – Задание 3.4

В задании 3.5 необходимо повторите предыдущее задание, но вывести на экран только города, начинающиеся с буквы «К». Результат выполнения задания показан на рисунке 3.5.

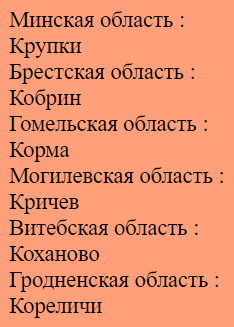


Рисунок 3.5 – Задание 3.5

В задании 3.6 необходимо объявить массив, индексами которого являются буквы русского языка, а значениями – соответствующие латинские буквосочетания. Реализовать функцию транслитерации строк. Результат выполнения задания показан на рисунке 3.6.



Рисунок 3.6 – Задание 3.6

В задании 3.7 необходимо написать функцию, которая заменяет в строке пробелы на подчеркивания и возвращает видоизмененную строчку. Результат выполнения задания показан на рисунке 3.7.



Рисунок 3.7 – Задание 3.7

В задании 3.8 необходимо объединить ранее написанные функции в одну, которая получает строку на русском языке, производит транслитерацию и замену пробелов на подчеркивания (аналогичная задача решается при конструировании *url*-адресов на основе названия статьи в блогах). Результат выполнения задания показан на рисунке 3.8.



Рисунок 3.8 – Задание 3.8

# 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.

Мною были изучены циклы и массивы в языке *PHP*. Также были изучены основные функции для работы с массивами.