Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет информационных технологий и управления

Кафедра ИТАС

Отчет

по лабораторной работе №1

«Основы *PHP*»

Вариант №17

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

Минск 2021

# Цель работы

Изучение основных возможностей объектно-ориентированного языка *PHP* в работе с переменными, типами данных, арифметическими операциями, логическими операциями, операциями сравнения и строками*.*

# Проектирование

Для каждого задания из лабораторной работы №1 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.

Константы представляют собой контейнеры для данных, как и переменные, но после присваивания константе значения его уже нельзя изменить. Константы создаются в *РНР* с помощью функции *define().*

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.

# Ход работы

## Вывод переменных

Программа, в которой с помощью оператора *echo* на экран будут выведены целочисленная переменная, переменная дробного типа, переменная булевского типа, строковая переменная и константа:

*<?php*

*$x = 5;*

*echo $x;*

*echo '</br>';*

*$x = 5.7;*

*echo $x;*

*echo '</br>';*

*$x = true;*

*echo $x;*

*echo '</br>';*

*$x = 'test';*

*echo $x;*

*echo '</br>';*

*define('PI', '3.14');*

*echo PI;*

*?>*

Результат выполнения программы представлен на рисунке 1.

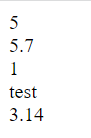


Рисунок 1 – Вывод значений переменных

Код программы повторного вывода с заключенными в двойные кавычки переменные:

*<?php*

*$x = 5;*

*echo "$x";*

*echo '</br>';*

*$x = 5.7;*

*echo "$x";*

*echo '</br>';*

*$x = true;*

*echo "$x";*

*echo '</br>';*

*$x = 'test';*

*echo "$x";*

*echo '</br>';*

*define('PI', '3.14');*

*echo "PI";*

*?>*

Результат повторного выполнения программы продемонстрирован на рисунке 2.



Рисунок 2 – Вывод значений переменных, заключенных в двоичные кавычки

Демонстрация кода повторного вывода с заключенными в одинарные кавычки переменные:

*<?php*

*$x = 5;*

*echo '$x';*

*echo '</br>';*

*$x = 5.7;*

*echo '$x';*

*echo '</br>';*

*$x = true;*

*echo '$x';*

*echo '</br>';*

*$x = 'test';*

*echo '$x';*

*echo '</br>';*

*define('PI', '3.14');*

*echo 'PI';*

*?>*

Результат выполнения программы проиллюстрирован на рисунке 3.



Рисунок 3 – Вывод значений переменных, заключенных в одинарные кавычки

## Работа со строками

Вывод на экран четверостишия, где каждая новая строка является отдельной строковой переменной:

*<?php*

*echo $s1 = "Печально я гляжу на наше поколенье!</br>";*

*echo $s2 = "Его грядущее — иль пусто, иль темно,</br>";*

*echo $s3 = "Меж тем, под бременем познанья и сомненья,</br>";*

*echo $s4 = "В бездействии состарится оно.</br></br>";*

*echo $a = "<u>М.Ю. Лермонтов</u>";*

*?>*

Результат вывода четверостишия показан на рисунке 4.

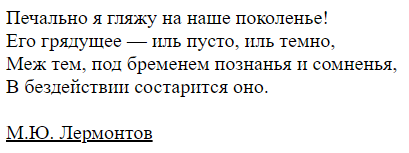


Рисунок 4 – Четверостишие с отдельными строковыми переменными

Программа, в которой все четверостишие заключено в одну переменную:

*<?php*

*echo $s = "Печально я гляжу на наше поколенье!</br>*

*Его грядущее — иль пусто, иль темно,</br>*

*Меж тем, под бременем познанья и сомненья,</br>*

*В бездействии состарится оно.</br></br>*

*<u>М.Ю. Лермонтов</u>";*

*?>*

После запуска программы получен результат как на рисунке 5.

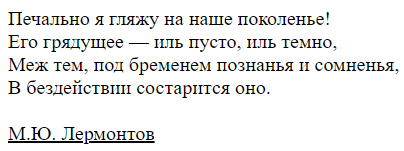


Рисунок 5 – Четверостишие с одной строковой переменной

## Сложение разных типов данных

Код страницы, в которой соединяются числовой и строковый тип данных:

*<?php*

*$x=50.7;*

*$y='UAH';*

*echo $x+$y;*

*?>*

При попытке соединения двух переменных разного типа, выводится ошибка как на рисунке 6.

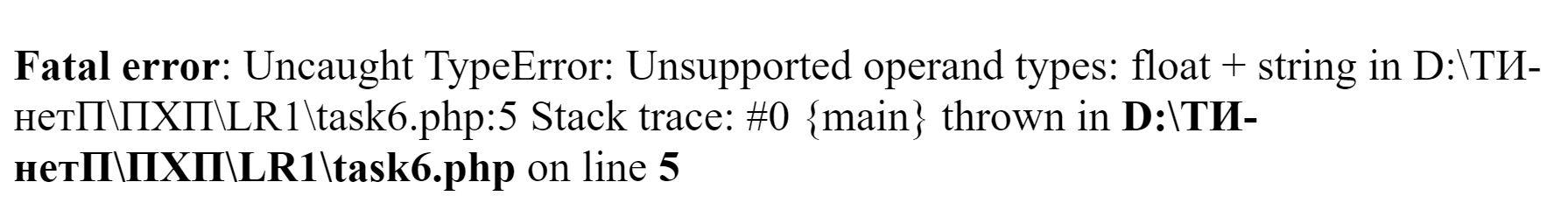


Рисунок 6 – Сложение разных типов данных

## Оператор *xor*

Cкрипт, который выводит значения операций со всеми возможными вариантами операндов:

*<?php*

*echo $a = (true xor true);*

*echo '</br>';*

*echo $b = (false xor true);*

*echo '</br>';*

*echo $c = (true xor false);*

*echo '</br>';*

*echo $d = (false xor false);*

*?>*

Результат работы оператора *xor* проиллюстрирован на рисунке 7.



Рисунок 7 – Значения операций со всеми возможными вариантами операндов

## Изменение значений переменных

Программа, в которой значения переменных меняются местами:

*<?php*

*echo $x = 10;*

*echo '</br>';*

*echo $y = 15;*

*echo '</br></br>';*

*$x = $x ^ $y;*

*$y = $x ^ $y;*

*$x = $x ^ $y;*

*echo '</br>';*

*echo $x;*

*echo '</br>';*

*echo $y;*

*?>*

После запуска программы получен результат как на рисунке 8.



Рисунок 8 – Изменение значений переменных

Заключение

В результате выполнения лабораторной работы №1 мною разработаны программы, иллюстрирующие выполнение заданий, с использованием такого языка программирования как *PHP*.