|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |

Институт Информационных технологий

Кафедра Инструментального и прикладного программного обеспечения

**ПРАКТИЧЕСКАЯ РАБОТА №2**

по дисциплине «Разработка серверных частей интернет-ресурсов»

**Тема практической работы: Основы PHP**

**Студент группы** ИКБО-30-20 Патина В.С.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись студента)

**Руководитель практической работы** преподаватель Благирев М.М.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись руководителя)

Работа представлена «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_ 2022 г.

Допущен к работе «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2022 г.

Москва 2022

**СОДЕРЖАНИЕ**

[Постановка задачи 1](#_Toc113622084)

[Выполнение работы 1](#_Toc113622085)

[Ссылка на удаленный репозиторий 8](#_Toc113622086)

[Ответы на вопросы к практической: 9](#_Toc113622087)

[ВЫВОДЫ 15](#_Toc113622088)

[СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ 16](#_Toc113622089)

Постановка задачи

Используя серверную конфигурацию, разработанную в прошлой практической работе выполнить следующие упражнения. Предполагается создать 3 независимых сервиса, устойчивых к минимальному набору самых простых ошибок. Предполагается создание 1 общего проекта с разделенными сервисами, разделением проекта на файлы для разделения функционала и переиспользования файлов. Каждый сервис должен состоять как минимум из 2 файлов.

Выполнение работы

Создадим рабочую директорию со структурой папок (Рисунок 1).

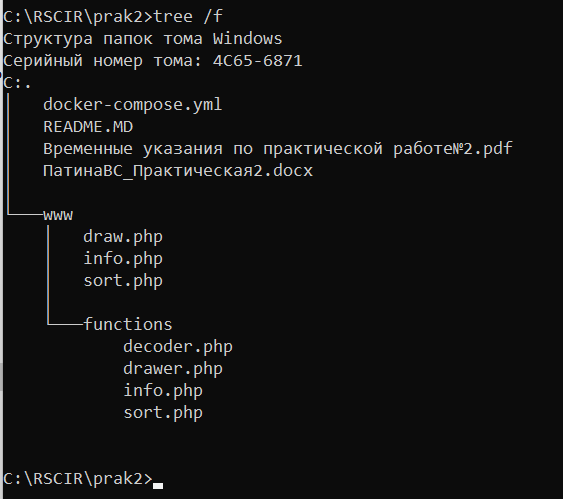


Рисунок 1 - Рабочая директория со структурой папок и файлов

Содержимое файла docker-compose.yml представлено на рисунке 2.

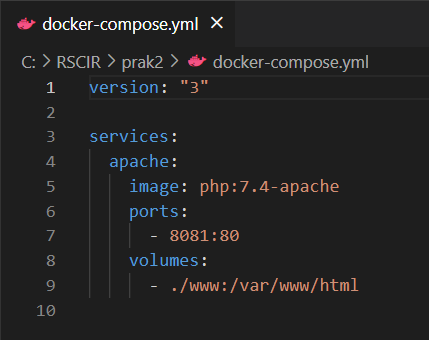


Рисунок 2 - Содержимое файла docker-compose.yml

Считываем код фигуры из запроса, расшифровываем его и выводим полученную фигуру на страницу (Рисунок 3).

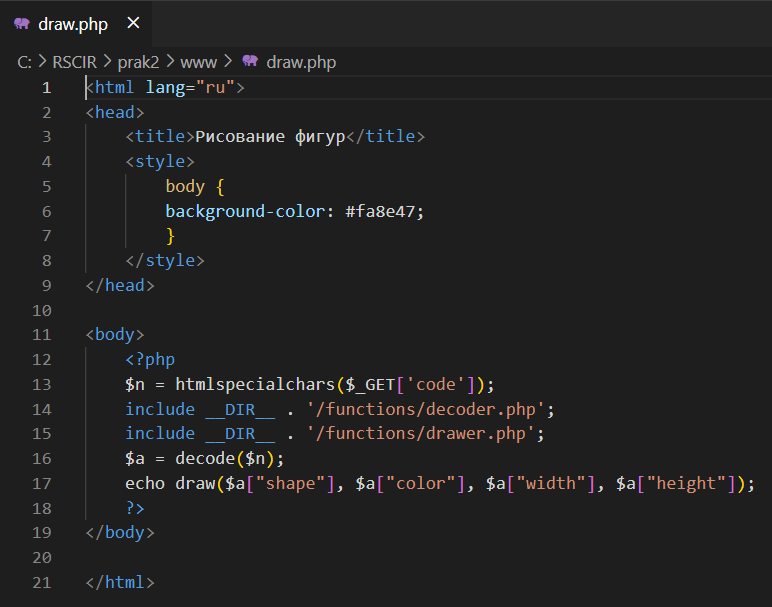


Рисунок 3 – Содержимое файла draw.php

Логика кодировки информации о фигуре описана на рисунке 4.



Рисунок 4 - Содержимое файла decoder.php

Отрисовка фигуры производится с помощью функции draw() (Рисунок 5).

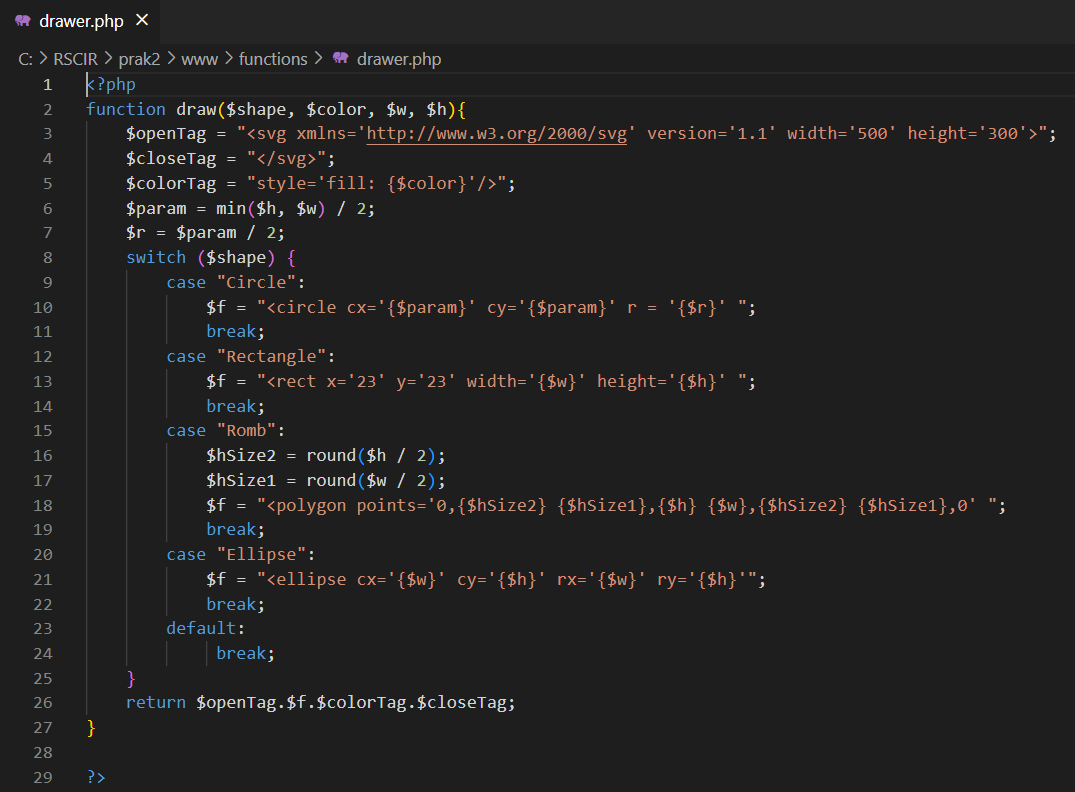


Рисунок 5 - Содежимое файла drawer.php

Работа веб-сервиса по рисованию svg-объектов представлена на рисунках 6, 7, 8.

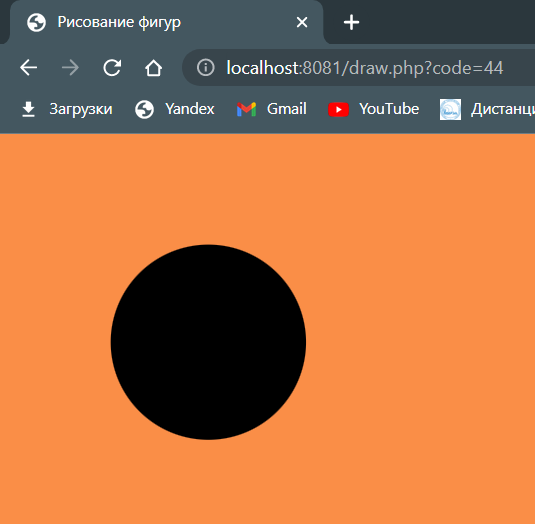


Рисунок 6 – Круг

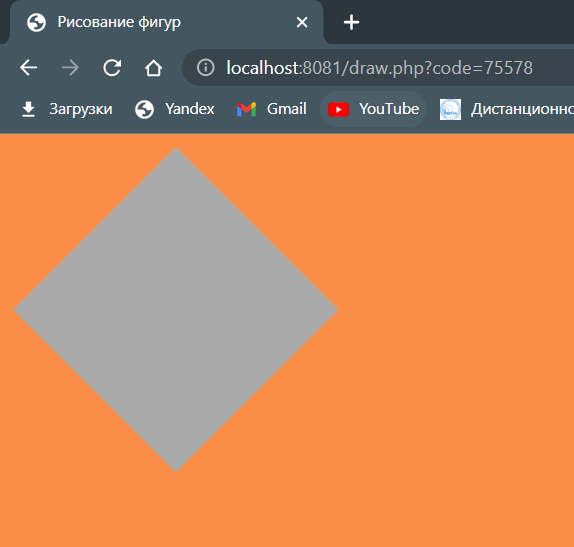


Рисунок 7 - Ромб

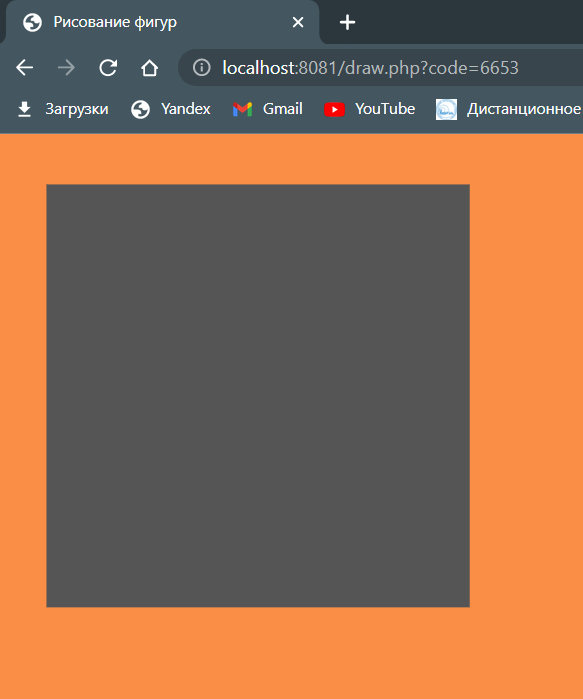


Рисунок 8 - Квадрат

Считываем массив (как строку) из запроса, преобразовываем эту строку в массив и выводим отсортированный массив (Рисунок 9).

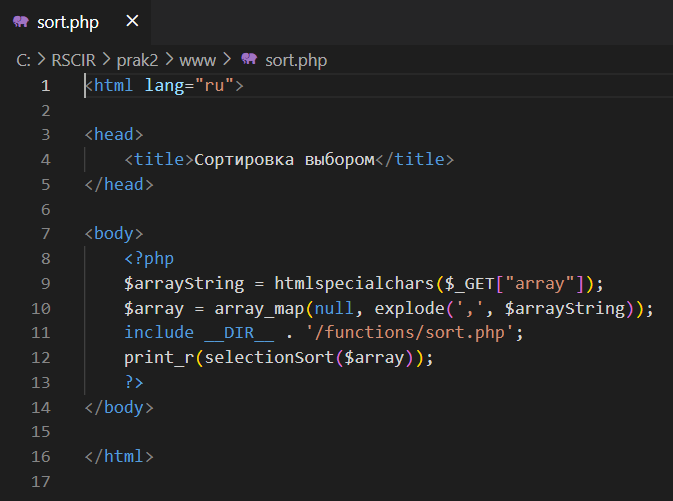


Рисунок 9 - Содержимое файла sort.php

Реализация сортировки выбором содержится в файле sort.php (Рисунок 10).

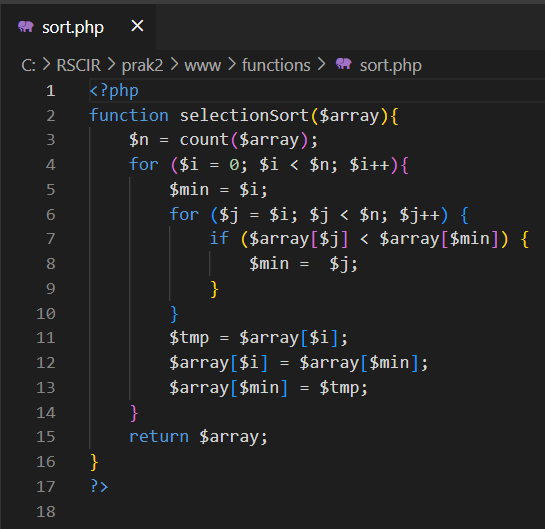


Рисунок 10 - Содержимое файла sort.php в папке functions

В итоге, получаем веб-страницу, содержащую отсортированный массив (Рисунок 11).

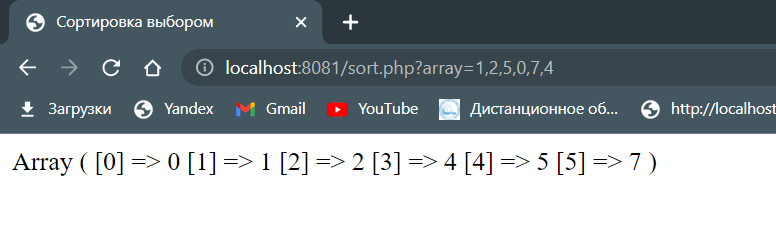


Рисунок 11 – Веб-страница, содержащая отсортированный массив

Считываем команды строкой, преобразовываем строку в массив из команд, выполняем их, печатаем результат выполнения команд на страницу (Рисунок 12).

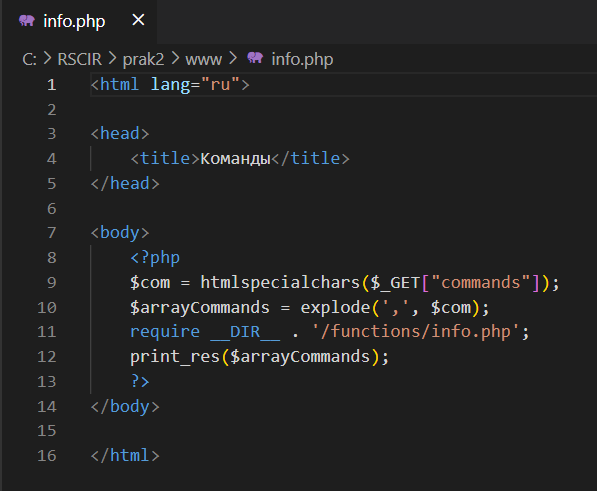


Рисунок 12 - Содержимое файла info.php

Выполнение и печать результата выполнения команд осуществляется с помощью функции print\_res (Рисунок 13).

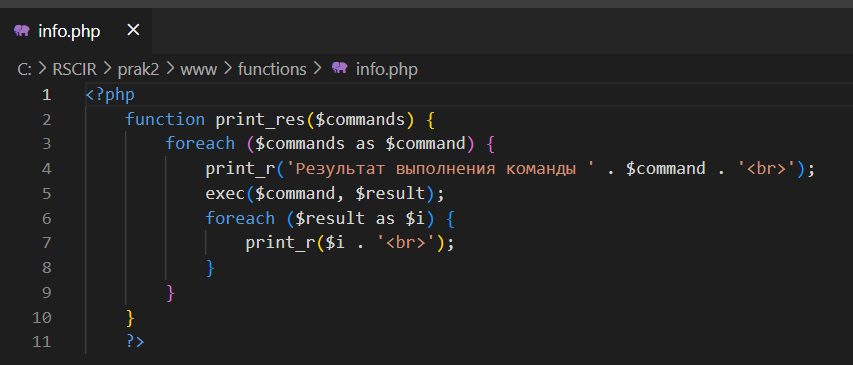


Рисунок 13 - Содержимое файла info.php в папке functions

Информационно-административная веб-страница о сервере (Рисунок 14).

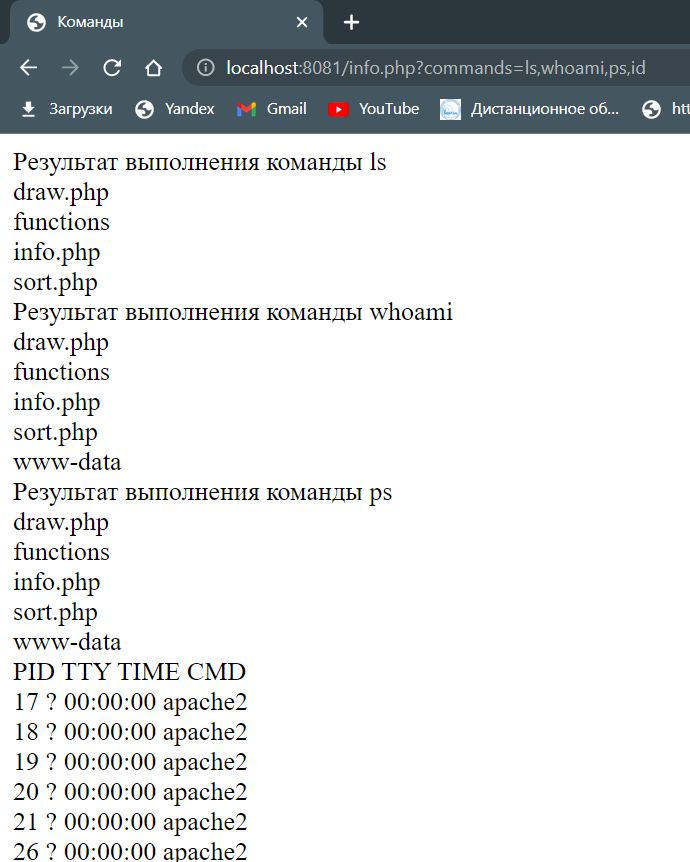


Рисунок 14 – Веб-страница, содержащая результат выполнения команд ls. whoami, id, ps

Ссылка на удаленный репозиторий

<https://github.com/devilishgrapefruit/RSCIR/tree/main/prak2> (Рисунок 15).

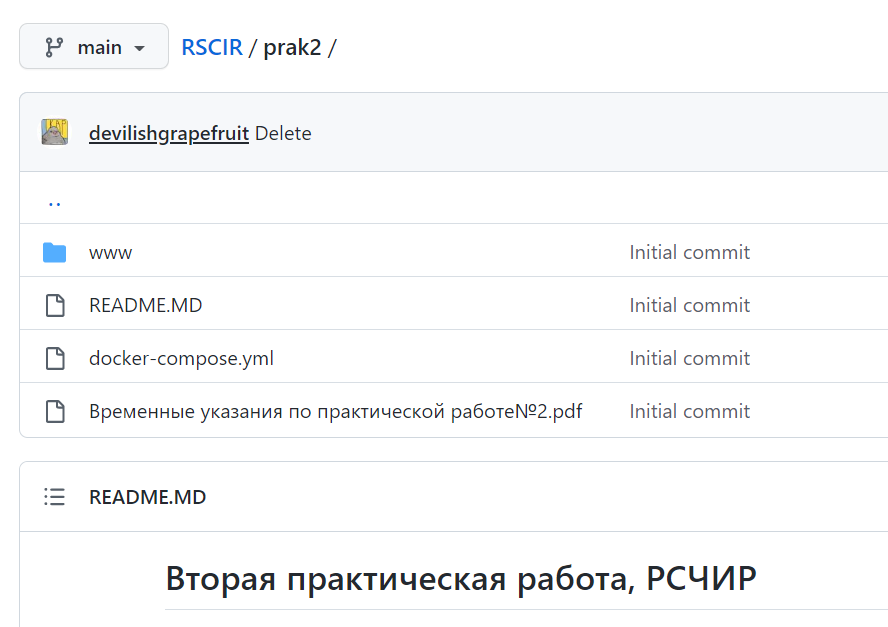


Рисунок 15 - Репозиторий

Ответы на вопросы к практической:

**1. Конфигурационный файл php.ini.**

В конфигурационном файле сосредоточены настройки интерпретатора PHP и его многочисленных расширений. По умолчанию файл конфигурации входит в дистрибутив PHP и называется php.ini-development. Чтобы интерпретатор PHP находил этот файл его нужно переименовать в php.ini.

**2. Как написать простой скрипт на php.**

Рассмотрим простейшую программу на PHP:

Весь код обернут в тег <?php?>, а сам код представляет собой оператор вывода echo и строку “Hello World”, которую данный оператор собственно выводит на экран. Важно отметить, что как самостоятельный код или как вставка в html-код скрипт PHP всегда оборачивается в тег <?php?>. В первом случае конец тега можно опустить.

**3. Основные правила, связанные с переменными в php.**

* Объявление переменной начинается со знака $. Данная особенность языка облегчает интерпретатору выделение переменных в тексте;
* Имена переменных должны состоять из латинских букв, цифр и нижнего подчеркивания. Хотя данное правило не является обязательным, рекомендуется его соблюдать;
* Имена переменных чувствительны к регистру.

**4. Основные типы данных в php.**

В PHP есть десять базовых типов данных: bool, int, float, string, array, object, callable, mixed.

**5. Какие существуют функции для работы с переменными в php вне**

**зависимости от типа данных.**

В PHP существует множество функций обработки переменных на уровне типов.

* Функция gettype. Эта функция возвращает тип, переданной ей переменной. Так как в PHP нет жесткого задания типа переменной, а также применяется динамическая типизация, обозначающая связывание типа переменной с ней при присвоении ей значения, она важна.
* Функция is\_type. Она возвращает ИСТИНУ или ЛОЖЬ в зависимости от того, принадлежит переменная типу type или нет.
* Функция isset. Она возвращает ИСТИНУ в случае, если переменная задана каким-либо значением. Чаще всего данная функция используется для проверки существования ключа в массиве.
* Функция settype, которая отвечает за преобразование типов. На вход ей подаются ссылка на переменную и тип данных к которой эту переменную нужно преобразовать.
* Функция unset. Данная функция удаляет переменную.
* Функция var\_dump. Результатом данной функции является удобно воспринимаемое человеком представление того, что ей передано с указанием типа и значения. Данная функция полезна в процессе отладки PHP-скриптов.

**6. Предопределенные переменные в php.**

PHP предоставляет всем скриптам большое количество предопределённых переменных. Эти переменные содержат всё, от внешних данных до переменных среды окружения, от текста сообщений об ошибках до последних полученных заголовков.

**7. Переменные переменных в php.**

Любопытным инструментом разработки PHP являются переменные переменных. То есть имя переменной, которое может быть определено и изменено динамически.

Листинг 9 - Пример переменной переменных

<?php

// определим обычную переменную

$a = 'hello';

// переменная переменной берет значение переменной и рассматривает его как имя переменной.

$$a = 'world'; // значение переменной $hello теперь такое

31

echo "$a ${$a}"; // hello world

echo "$a $hello"; //hello world

?>

В примере создается строковая переменная $a. Далее создается переменная переменной. То есть объявляется переменная с динамическим именем, хранимым в переменной $a. Переменная переменной берет значение переменной и рассматривает его как имя переменной.

**8. Выражения в php.**

Выражения − это самые важные строительные элементы PHP. Почти всё, что пишется на PHP, является выражением. Самое простое и точное определение выражения − "все, что угодно, имеющее значение". Самым простым примером выражения является переменная или константа в операторе присваивания.

**9. Арифметические операторы в php.**

* +$a – Идентичность;
* -$a – Отрицание;
* $a + $b – Сложение;
* $a - $b – Вычитание;
* $a \* $b – Умножение;
* $a / $b – Деление;
* $a % $b - Деление по модулю;
* $a \*\* $b – Возведение в степень.

**10. Битовые операции в php.**

* $a & $b - Побитовое 'и';
* $a | $b - Побитовое 'или’;
* $a ^ $b - Исключающее или;
* ~ $a – Отрицание;
* $a << $b - Сдвиг влево;
* $a >> $b - Сдвиг вправо.

**11.Оператор присваивания в php.**

Отдельным разделом рассматриваются возможные операции с оператором присваивания. Основным является оператор простого присваивания “=”. Оператор присваивания означает, что левый операнд получает значение правого выражения, то есть устанавливается значением. Результатом выполнения оператора присваивания является само присвоенное значение.

**12.Операторы сравнения в php.**

Операторы сравнения представляют собой уникальные операции, независимо от типа операндов, возвращающие одно из 2 значений: true или false.

* $a > $b – Больше;
* $a < $b – Меньше;
* $a == $b – Равно;
* $a === $b – Тождественно равно;
* $a != $b – Не равно;
* $a <> $b – Не равно;
* $a <= $b – Меньше или равно;
* $a >= $b – Больше или равно;
* $a <=>$b – Космический корабль.

**13.Логические операторы в php.**

* $a and $b – И;
* $a or $b – Или;
* $a xor $b - Исключающее или;
* ! $a – Отрицание;
* $a && $b – И;
* $a || $b – Или.

**14.Условная конструкция в php.**

Данная конструкция предоставляет возможность условного выполнения фрагментов кода. Структура if реализована в PHP по аналогии с языком C. Нужно понимать, что выражение else выполняется только, если выражение if вычисляется как false, и если нет других любых выражений elseif, или если они все равны false. Конструкция elseif, есть сочетание конструкций if и else. Аналогично else, она расширяет оператор if для выполнения различных выражений в случае, когда условие начального оператора if эквивалентно false. Однако, в отличие от else, выполнение альтернативного выражения произойдет только тогда, когда условие оператора elseif будет являться равным true.

**15.Циклы в php.**

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: Цикл с предусловием (while); Цикл с постусловием (do-while); Цикл со счетчиком (for); Специальный цикл перебора массивов (foreach).

**16. Конструкции switch и match в php.**

В ситуации сложных и многоуровневых условных конструкций на помощь приходит конструкция switch-case. Данная конструкция используется, когда требуется сравнивать одну и ту же переменную (или выражение) с множеством различных значений и выполнять различные участки кода в зависимости от того, какое значение принимает эта переменная (или выражение).

В версии PHP 8.0.0 появилась новая конструкция для ветвления потока исполнения на основании проверки совпадения значения с заданным условием. Конструкция аналогична switch, но имеет некоторые существенные различия, которые способствуют применению данной конструкции. Главным отличием от switch является возвращение результата, а также привязка сравнения также к типам.

**17. Include и require в php.**

Идея разделения программы на несколько файлов очень важна для переиспользования кода, создания шаблонов и т.д. Первое, о чем нужно сказать в данном разделе это задание путей поиска файлов для PHP. Это задается директивой include\_path в файле php.ini, указывается список директорий, в которых функции require, include, fopen(), file(), readfile() и file\_get\_contents() ищут файлы. Итак, выражение include включает и выполняет указанный файл.

Оператор include выдает предупреждение и продолжает исполнение кода, если запрашиваемый файл не был найден. В отличие от выражения include выражение require при невозможности добавления файла останавливает выполнение скрипта и выдает фатальную ошибку уровня E\_COMPILE\_ERROR.

**18. Функции в php.**

Функции являются неотъемлемой частью любого языка программирования, и PHP не исключение. Для понимания синтаксиса описания функций и их использования нужно понимать, что:

1) у функции может быть переменное количество параметров и параметры по умолчанию;

2) функция имеет собственную область видимости. Это нужно учитывать при работе с глобальными переменными программы и с локальными переменными функции, которые уничтожаются при окончании ее работы;

3) в PHP тип возвращаемого значения может быть любым и тип возвращаемого значения не регламентируется. То есть функция может возвращать и число, и строку одновременно;

4) допускается создание и использование при необходимости анонимных функций;

5) в пределах одного сценария не должно быть определений двух функций с одинаковыми именами.

ВЫВОДЫ

В работе была продолжена настройка сложной серверной конфигурации, состоящей из связки apache+nginx+php+База данных. В результате работы было создано веб-приложение с динамическими веб-страницами и веб-страницами со статическим контентом.

СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ

1. Разработка серверных частей интернет-ресурсов [Электронный ресурс]: Учебное пособие / Волков М.Ю., Литвинов В.В., Лобанов А.А. и др. — М.: МИРЭА – Российский технологический университет, 2021.