Министерство образования и науки Российской Федерации

ФГБОУ ВО «Ульяновский государственный технический университет»

Кафедра «Вычислительная техника»

Лабораторная работа №3

«Функции»

Выполнил студент:

группы ИВТАПбд-11

Галацков И.А.

Проверил:

Игонин А.Г.

Ульяновск, 2022

**Оглавление**

[**Постановка задачи 3**](#_Toc103941458)

[**Метод выполнения 4**](#_Toc103941459)

[**Блок – схема 5**](#_Toc103941460)

[**Интерфейс работы 6**](#_Toc103941461)

[**Листинг кода 7**](#_Toc103941462)

[**Выводы 10**](#_Toc103941463)

[**Литература 11**](#_Toc103941464)

# Постановка задачи

Определение является ли отношение функцией. Вводить необходимо отношение и элементы обоих множеств.

# Метод выполнения

Бинарное отношение из множества A в множество B является функцией в случае, если каждый элемент множества A связан с единственным элементом множества B. При этом элементы множества B могут быть связаны более чем с одним элементом множества А.

Для реализации данной работы был использован PHP в качестве языка для описания решения поставленной задачи.

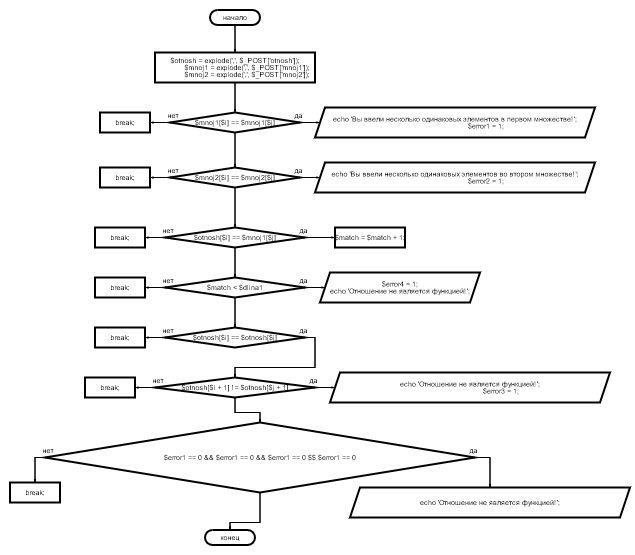
В php-коде интегрирован html-код, внутри которого описана форма, с которой считываются данные. Для того, чтобы введённое значение вписалось в переменную необходимо использовать свойство тега input – value, внутри которого содержится php-код, благодаря которому происходит запись данных в глобальную переменную POST. После чтения данных из каждого текстового поля происходит обработка полученных данных. При чтении используется explode в которой задан символ запятой – это предназначено для разделения и запись в переменную из текстового поля, в котором метод ввода задан как «a,b,c».

Сначала происходит проверка ввода на наличие повторяющихся элементов, далее происходит отыскание и подсчёт элементов первого множества, которые присутствуют в отношении и происходит запись элементов в переменную overlap. В случае несоответствия – вывод сообщения об ошибке.

Последующее действие – отыскание одинаковых элементов из первого множества во втором.

В каждом случае применяются переменные, содержащие ответ для последней функции, которая проверяет их на равность нулю. При равности выводится ответ – «Является функцией»

# Блок – схема

****

# Интерфейс работы

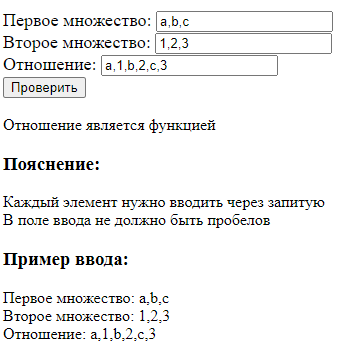
****

Рисунок 1 - страница лабораторной

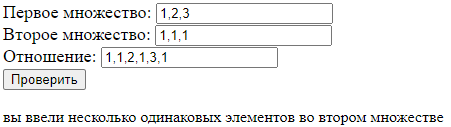


Рисунок 2 – пример ошибки

# Листинг кода

PHP

<!DOCTYPE html>  
<html>  
<head>  
 <meta charset="UTF-8">  
 <title>Лабораторная работа №3</title>  
</head>  
<body >  
 <form action="" method="post">  
 <p style="font-size: 18px;">Первое множество: <input type="text" name="mnoj1" value="<?php  
 if (isset($\_POST['mnoj1'])) {  
 echo $\_POST['mnoj1'];  
 }?>">  
 <br>Второе множество: <input type="text" name="mnoj2" value="<?php  
 if (isset($\_POST['mnoj2'])) {  
 echo $\_POST['mnoj2'];  
 }?>">  
 <br>Отношение: <input type="text" name="otnosh" value="<?php  
 if (isset($\_POST['otnosh'])) {  
 echo $\_POST['otnosh'];  
 }?>">  
 <br><input type="submit" name="sub" value="Проверить"><br>  
 </form>  
 <?php  
 //Cоздаем массивы и получаем данные из input'ов  
 ini\_set('display\_errors', 'Off');  
 if (isset($\_POST['sub'])) {  
 $indikator = 0;  
 $error1 = 0;  
 $error2 = 0;  
 //разбиваем данные на массивы по разделителю в виде запятой  
 $otnosh = explode(',', $\_POST['otnosh']);  
 $mnoj1 = explode(',', $\_POST['mnoj1']);  
 $mnoj2 = explode(',', $\_POST['mnoj2']);  
 $dlinaotn = count($otnosh);  
 $dlina1 = count($mnoj1);  
 $dlina2 = count($mnoj2);  
 //проверяем элементы первого множества на повтор с помощью двойного цикла.  
 for($i = 0; $i<$dlina1; $i++){  
 for($j = 0; $j<$dlina1; $j++){  
 if($i != $j){  
 if($mnoj1[$i] == $mnoj1[$j]){  
 echo 'вы ввели несколько одинаковых элементов в первом множестве';  
 echo "<br>";  
 $error1 = 1;  
 break;  
 }  
 }  
 }  
 if($error1 == 1){  
 break;  
 }  
 }  
 //проверяем элементы второго множества на повтор.  
 for($i = 0; $i<$dlina2; $i++){  
 for($j = 0; $j<$dlina2; $j++){  
 if($i != $j){  
 if($mnoj2[$i] == $mnoj2[$j]){  
 echo 'вы ввели несколько одинаковых элементов во втором множестве';  
 echo "<br>";  
 $error2 = 1;  
 break;  
 }  
 }  
 }  
 if($error2 == 1){  
 break;  
 }  
 }  
 /\*  
 Смотрим сколько элементов из первого множества присутствует в отношении и записываем в переменную $match.  
 \*/  
 $error4 = 0;  
 $match = 0;  
 for($i = 0; $i<$dlinaotn; $i++){  
 if($error4 == 0){  
 if($i % 2 == 0){  
 for($j = 0; $j<$dlina1; $j++){  
 if($otnosh[$i]==$mnoj1[$j]){  
 $match = $match + 1;  
 }  
 }  
 }  
 }  
 }  
 //Если какого то элемента не хватает, выводим сообщение об этом.  
 if($match < $dlina1){  
 $error4 = 1;  
 echo 'Отношение не является функцией.';  
 echo "<br>";  
 }  
 //Пробегаемся по отношению, и если встречаем несколько раз элемент первого множества, соответствующий разным элементам второго множества, узнаем что отношение не является функцией.  
 $error3 = 0;  
 for($i = 0; $i<$dlinaotn; $i++){  
 if($error3 == 0){  
 for($j = 0; $j<$dlinaotn; $j++){  
 if($error3 == 0){  
 if($i != $j){  
 if($i % 2 == 0){  
 if($otnosh[$i] == $otnosh[$j]){  
 if($otnosh[$i + 1] != $otnosh[$j + 1]){  
 // echo $otnosh[$i];  
 echo 'Отношение не является функцией.';  
 echo "<br>";  
 $error3 = 1;  
 break;  
 }  
 }  
 }  
 }  
 }  
 }  
 }  
 }  
 //Если все предыдущие условия прошли без ошибок, то отношение является функцией.  
 if($error1 == 0){  
 if($error2 == 0){  
 if($error3 == 0){  
 if($error4 == 0){  
 echo 'Отношение является функцией';  
 }  
 }  
 }  
 }  
 }  
 ?>  
<div class ="poyasnenie">  
 <p><h3>Пояснение:</h3></p>  
 <p>  
 Каждый элемент нужно вводить через запитую <br>  
 В поле ввода не должно быть пробелов <br>  
  
  
  
 </p>  
</div>  
<div class="primer">  
 <p><h3>Пример ввода:</h3></p>  
 <p>  
 Первое множество: a,b,c <br>  
 Второе множество: 1,2,3 <br>  
 Отношение: a,1,b,2,c,3  
  
 </p>  
</div>  
</body>  
</html>

# Выводы

С помощью данной лабораторной работы были изучены построение матрицы отношений из двух множеств, а также разработаны навыки проверки отношений на функцию.

# Литература

1. Р. Хаггарти. Дискретная математика для программистов.
2. https://habr.com/ru/post/515014/