Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Ульяновский государственный технический университет»

Кафедра «Вычислительная техника»

Дисциплина «Математическая логика и дискретная математика»

Лабораторная работа № 5

«создание игры»

Выполнил студент

группы ИВТАПбд-12

Матюнин И.С.

Ульяновск, 2022

Оглавление

[**Цель Работы** 3](#_Toc102728764)

[**Описание метода работы** 3](#_Toc102728765)

[**Код:** 6](#_Toc102728766)

[**Список использованной Литературы** 8](#_Toc102728767)

# **Цель Работы:**

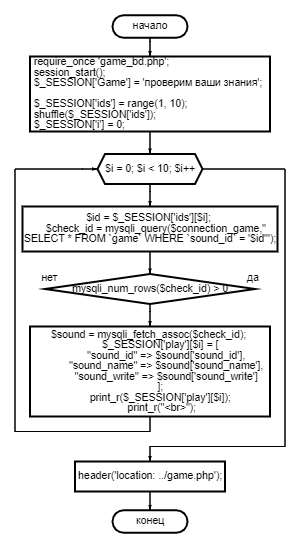
Создать игру на своём сайте с использованием всех полученных знаний о использовании баз данных, php, css и html.

# **Описание метода работы:**

Моя игра заключается в том, что пользователю случайным образом последовательно предоставляются произношения слов и поля для ввода этих слов.

Для её реализации я создал в папке проекта папку с необходимыми для игры файлами php формата и javascript файл. Так же я создал новую базу данных и в ней таблицу «game». В таблицу я занёс правильное написание слов и пути к записям этих слов. Записи я сделал сам в программе FL STUDIO.

Для старта игры, создаётся массив из идентификаторов, они перемешиваются и после, правильное написание слов и пути к их записям сохраняются в суперглобальный массив $\_SESSION[‘game’].



Блок-схема 1: запуск игры.

Когда игра начинается, сервер загружает из таблицы файлы с произношением слов и выводит кнопку проигрывания вместе с полями для ввода по одному. Переход на новое слово осуществляется по нажатию на кнопку «Следующее слово».

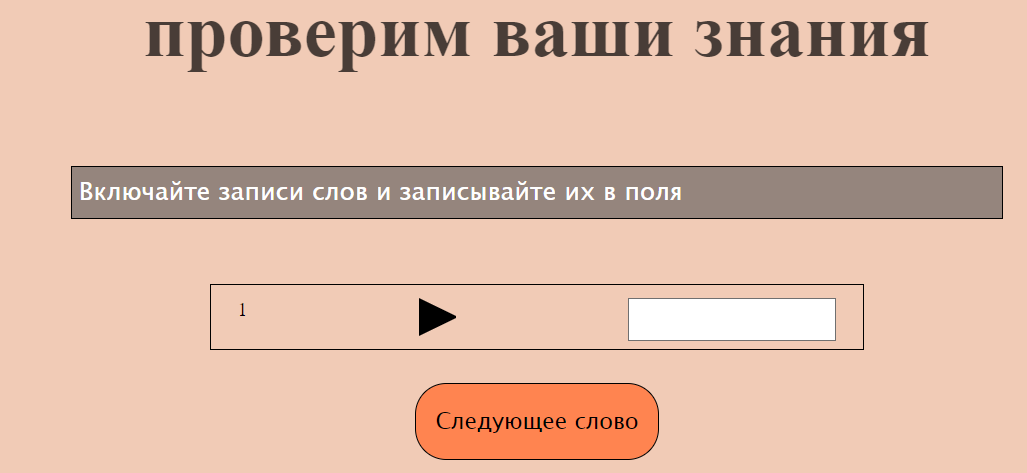
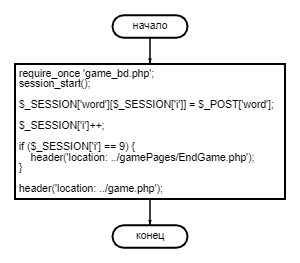


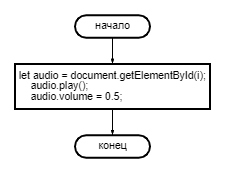
Рис. 1 – одно выведенное слово.



Блок-схема 2: сохранение введённого слова и вывод следующего.

После каждого нажатия на кнопку написанные пользователем слова сохраняются в суперглобальном массиве $\_SESSION['word']. Затем выводится новое слово и всё повторяется.

Записи слов воспроизводятся с помощью javascript.



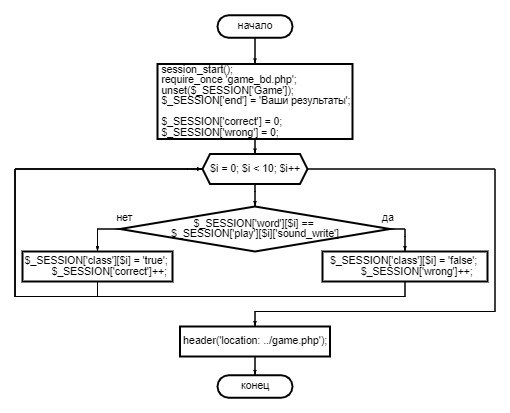
Блок-схема 3: воспроизведение записи слова.

Как только программа дошла до последнего слова пользователю выводится кнопка «завершить игру» вместо «Следующее слово». После этого загружается предпоследняя страница игры. На ней выводится кнопка «подвести итоги». Как только пользователь её нажимает, программа сверяет ответы пользователя с правильными и подсчитывает верные и неверные ответы.

Все верные ответы выделяются зелёным цветом, а все неверные – красным. Если правильных ответов больше, пользователь услышит аплодисменты. Иначе же воспроизведётся звук ворчания и недовольства. Звуки воспроизводятся с помощью скрипта, прописанного на самой странице.

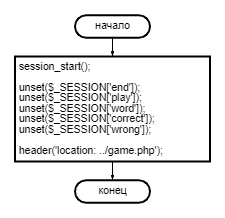


Рис. 2 – таблица итогов пользователя.



Блок-схема 5: подсчёт результатов.

Чтобы завершить игру пользователь должен нажать на кнопку «покинуть игру». Это удалит все ранее использованные данные.



Блок-схема 5: выход из игры.

# **Код:**

*GameStatus.php:*

<?php

require\_once 'game\_bd.php';

session\_start();

$\_SESSION['Game'] = 'проверим ваши знания';

$\_SESSION['ids'] = range(1, 10);

shuffle($\_SESSION['ids']);

$\_SESSION['i'] = 0;

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

    $id = $\_SESSION['ids'][$i];

    $check\_id = mysqli\_query($connection\_game,"SELECT \* FROM `game` WHERE `sound\_id` = '$id'");

    if (mysqli\_num\_rows($check\_id) > 0) {

        $sound = mysqli\_fetch\_assoc($check\_id);

        $\_SESSION['play'][$i] = [

            "sound\_id" => $sound['sound\_id'],

            "sound\_name" => $sound['sound\_name'],

            "sound\_write" => $sound['sound\_write']

        ];

        print\_r($\_SESSION['play'][$i]);

        print\_r("<br>");

    }

}

header('location: ../game.php');

?>

*nextGamePage.php:*

<?php

require\_once 'game\_bd.php';

session\_start();

$\_SESSION['word'][$\_SESSION['i']] = $\_POST['word'];

$\_SESSION['i']++;

if ($\_SESSION['i'] == 9) {

    header('location: ../gamePages/EndGame.php');

}

header('location: ../game.php');

?>

*EndGame.php:*

<?php

session\_start();

require\_once 'game\_bd.php';

unset($\_SESSION['Game']);

$\_SESSION['end'] = 'Ваши результаты';

$\_SESSION['correct'] = 0;

$\_SESSION['wrong'] = 0;

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

    if ($\_SESSION['word'][$i] == $\_SESSION['play'][$i]['sound\_write']) {

        $\_SESSION['class'][$i] = 'true';

        $\_SESSION['correct']++;

    } else {

        $\_SESSION['class'][$i] = 'false';

        $\_SESSION['wrong']++;

    }

}

header('location: ../game.php');

?>

*Restart.php:*

<?php

session\_start();

unset($\_SESSION['end']);

unset($\_SESSION['play']);

unset($\_SESSION['word']);

unset($\_SESSION['correct']);

unset($\_SESSION['wrong']);

header('location: ../game.php');

?>

*Game.js:*

function play(i) {

    let audio = document.getElementById(i);

    audio.play();

    audio.volume = 0.5;

}

# **Список использованной Литературы:**

* Воспроизведение звука на javascript [электронный ресурс] URL: <https://myrusakov.ru/javascript-sound.html>
* PHP: shuffle – manual [электронный ресурс] URL: <https://www.php.net/manual/ru/function.shuffle.php>
* Как написать игру на JavaScript [электронный ресурс] URL: <https://skillbox.ru/media/code/kak_napisat_igru_na_javascript/>