

## Создание системы для решения случайных арифметических примеров на время в JavaScript

*Романюк Виктория Дмитриевна*

*Приамурский государственный университет им. Шолом-Алейхема*

*Студент*

*Фатеенков Данила Витальевич*

*Приамурский государственный университет им. Шолом-Алейхема*

*Студент*

### Аннотация

В работе рассматривается создание системы для решения арифметических примеров на время. Для реализации данной задачи был применен язык программирования JavaScript.

**Ключевые слова:** JavaScript, HTML, арифметика, программирование.

## Creating a system for solving random arithmetic examples on time in JavaScript

*Romanyuk Viktoriya Dmitrievna*

*Sholom-Aleichem Priamursky State University*

*Student*

*Fateenkov Danila Vitalievich*

*Sholom-Aleichem Priamursky State University*

*Student*

### Abstract

The paper considers the creation of a system for solving arithmetic examples for a while. To implement this task, the JavaScript programming language was used.

**Keywords:** JavaScript, HTML, arithmetic, programming.

### 1. Введение

#### 1.1 Актуальность

В современном мире информационных технологий все большую роль играют различные веб-приложения. Они позволяют пользователю быстро получать необходимую информацию и решать задачи в реальном времени. Одно из таких приложений – это система для решения случайных арифметических примеров на время, разработанная на языке программирования JavaScript.

В сфере образования система для решения случайных арифметических примеров на время может занять важную роль в обучении математике. Она

поспособствует развитию навыков решения задач, позволит тренировать устный счет и повысит скорость реакции на математические операции. Такая система позволит эффективно проводить занятия в классе, обеспечивая активную вовлеченность учащихся и повышая их мотивацию.

## **1.2 Обзор исследований**

Н.Ф. Калимулин, А.С. Павлова, А.Д. Борисова, Н.К. Петрова и Р.М. Хамитов опубликовали статью, в которой описали процесс написания алгоритма для приложения, которое сможет решать арифметические примеры [1]. Для разработки использовался язык программирования C++.

А.И. Дементьева, Е.Н. Лубягина и Д.В. Шабалин в своей исследовательской работе рассмотрели инструменты разработки веб-ресурсов с точки зрения создания сайта для преподавания математике и информатике [2]. В статье рассматривались также уже существующие системы, которые направлены на то, чтобы помочь пользователю в изучении тех или иных аспектов математики.

Т.А. Широкова рассмотрела организацию работы по формированию вычислительных навыков с помощью разноуровневых заданий [3].

## **1.3 Цель исследования**

Цель – разработать систему для решения арифметических примеров на время, сгенерированных рандомизированным алгоритмом.

## **2. Материалы и методы**

Для реализации поставленной цели используются следующие языки программирования: JavaScript и PHP. Для разметки элементов на странице используется HTML.

## **3. Результаты и обсуждения**

Перед тем, как начать разработку системы для решения случайных арифметических примеров на время, нужно рассмотреть все доступные инструменты для создания подобного веб-приложения.

Для реализации разметки элементов был выбран HTML. Его преимущество при написании заключается в том, что язык HTML является основой всех веб-страниц и доступен в любом браузере; он легко воспроизводится, различные элементы разметки легко копируются и вставляются; HTML достаточно быстро обрабатывается браузером.

Для реализации функционала приложения был выбран JavaScript. Данный язык программирования используется при написании многих веб-приложений и поддерживается многими библиотеками и фреймворками. JavaScript обеспечивает интерактивность и динамизм веб-страницы, а также работает со всеми браузерами.

Рассмотрим уже существующие системы для решения алгебраических примеров.

Одной из таких систем является сайт Symbolab (Рис.1). В основном данный сайт включает в себя множество различных калькуляторов, которые поддерживают расчеты, начиная от математических и заканчивая геометрическими и т.д. С помощью раздела Practice пользователь может самостоятельно прорешивать примеры разной сложности и по разным темам, которые представлены на сайте: Порядок операций (целые числа), Порядок операций (целые числа, экспоненты), Длинная арифметика, Множители и простые числа, Десятичные дроби. Все данные темы представлены в разделе Предварительная алгебра. На сайте же существует множество подобных разделов, а именно: Предварительная алгебра, Алгебра, Матрицы и векторы, Функции, Тригонометрия, Математическое исчисление и Словесные задачи.

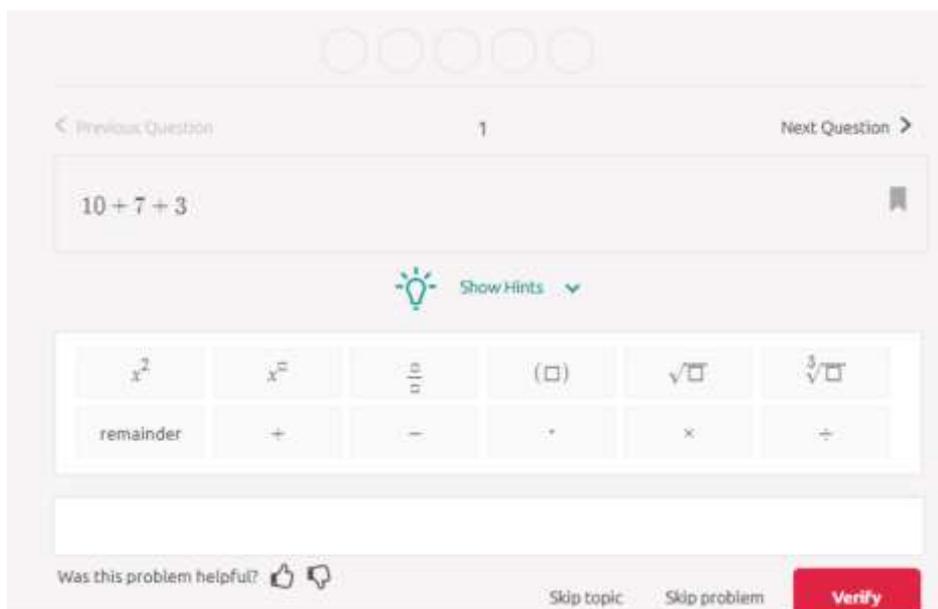


Рисунок 1 – Решение алгебраического примера на сайте Symbolab

Создание системы для решения алгебраических примеров можно подразделить на несколько этапов:

1. Определение требований. Необходимо определить, что должна делать система, как она должна работать, какие требования нужно удовлетворить.

2. Проектирование системы. На данном этапе определяется архитектура системы и ее компоненты.

3. Разработка. На этапе разработки происходит написание кода и тестирование его работы.

4. Тестирование и отладка. На этом этапе проверяется, соответствует ли система требованиям, а также выявляются и устраняются возможные ошибки.

Работу данной системы можно описать следующим образом:

1. Пользователь заходит на главную страницу, где ему предлагается начать решение математических примеров.

2. Открывается страница с алгебраическими примерами, которые необходимо решить.

3. Пользователь решает примеры до тех пор, пока не закончится отведенное время.

4. После окончания времени пользователь переходит на страницу с результатами, где ему предлагается начать при желании заново.

В начале работы над системой для решения случайных арифметических примеров необходимо определить, из каких файлов будет состоять проект и то, как они будут взаимодействовать между собой.

Всего в проекте находится 4 файла, которые нужны для работоспособности калькулятора:

- index.html – титульная страница, на которой объясняются правила и предлагается перейти к решению примеров;

- math.html – страница с решением арифметических примеров на время, на которой реализована система;

- result.html – страница, на которой выводится результат решения арифметических примеров;

- prog.js – JavaScript файл, в котором описана логика работы системы для решения случайных арифметических примеров.

На главной, титульной, странице, выводится приветствие, краткие правила и кнопка «Начать» для старта решения арифметических примеров (Рис. 2).

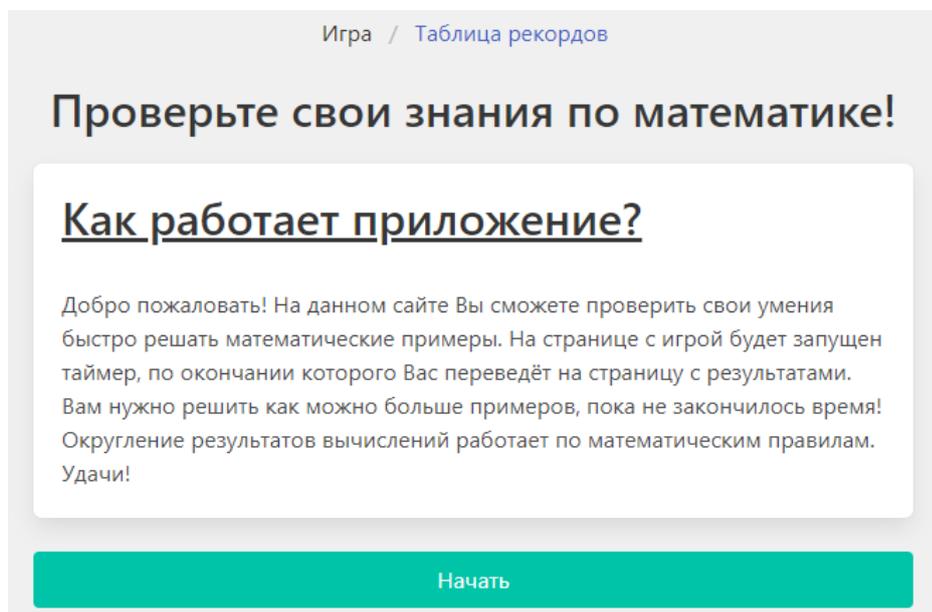


Рисунок 2 – Главная страница приложения

Если пользователь нажмёт на кнопку «Начать», то автоматически перейдет на страницу с генерацией случайных примеров, которые необходимо решить за 3 минуты (Рис. 3).

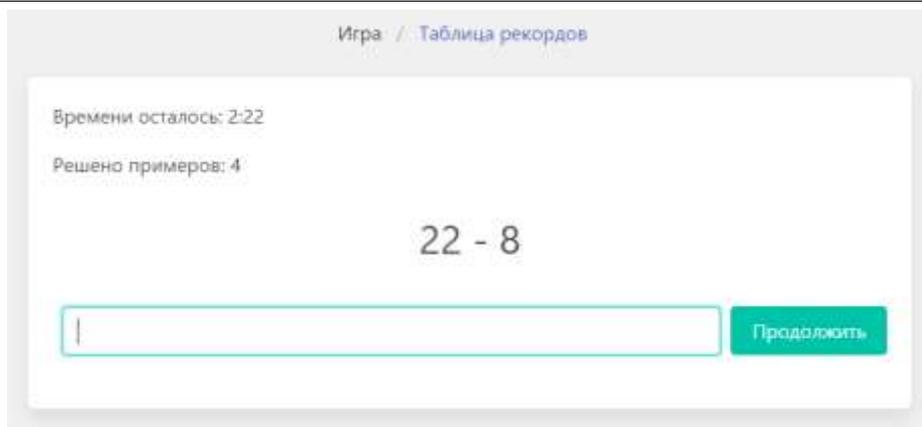


Рисунок 3 – Страница с решением арифметических примеров

На данной странице задействован скрипт prog.js. В верхней части страницы, в первой строке, представлено время в секундах. Во второй строке представлено количество решенных примеров. На третьей строке находится случайный сгенерированный пример, который пользователю предлагается решить, а ниже окно для введения итогового ответа. Кнопка «Продолжить» переносит пользователя на страницу с результатами (result.html).

Рассмотрим подробнее, как выглядит HTML код титульной страницы (Рис.4.). Для Frontend составляющей системы используется фреймворк Vulma, но на рисунках, демонстрирующих код, сохранены только основные элементы страницы, чтобы код выглядел менее нагруженным.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Главная страница</title>
  </head>
  <body>
    <h1>Проверьте свои знания по математике!</h1>
    <p>Добро пожаловать! Чтобы начать, нажмите кнопку ниже!</p>
    <p>На следующей странице случайным образом будут создаваться примеры и Вы должны их решать.<br>
    Решение нужно вводить в строку, которая будет расположена под примером.<br>
    У Вас будет ограниченное время и за это время Вы должны решить как можно больше математических примеров.<br>
    Также в примерах может встречаться деление чисел. Округляйте остаток по математическим правилам.</p>
    <button onclick="start()">Начать</button>
    <script src="prog.js"></script>
  </body>
</html>
```

Рисунок 4 – HTML код титульной страницы

Чтобы облегчить JavaScript код, переход на страницу с алгебраическими примерами осуществляется напрямую с HTML страницы. Для того, чтобы реализовать такой переход, нужно задать атрибут “onclick” и указать в нём JavaScript код, который будет переводить на другие страницы.

HTML код страницы с решением примеров (файл math.html) выглядит следующим образом (Рис 5.):

```
<!DOCTYPE html>
<html>
  <head>
    <title>Решение примеров</title>
    <script src="prog.js"></script>
  </head>
  <body onload="main()">
    <p id="timer"></p>
    <p id="score">Решено примеров: 0</p>
    <p id="equation"></p>
    <input type="text" id="answer">
    <button onclick="checkAnswer()">Продолжить</button>
  </body>
</html>
```

Рисунок 5 – HTML код страницы с примерами

Как только у пользователя закончилось отведенное ему время на решение примеров, его автоматически переносит на страницу с результатами (Рис. 6).

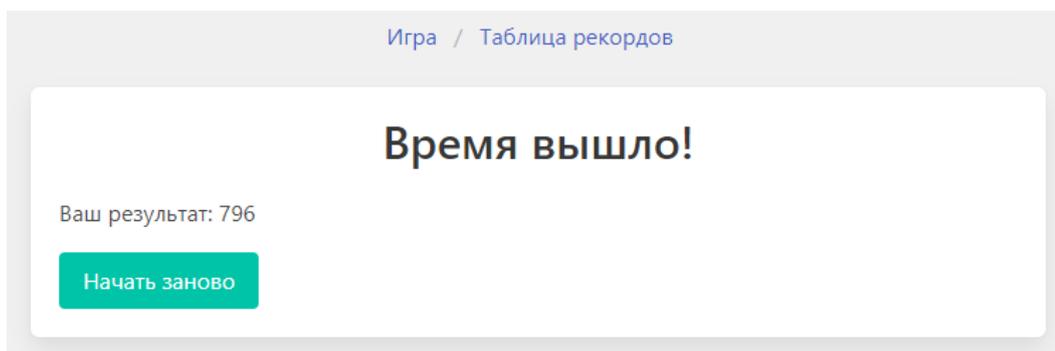


Рисунок 6 – Страница с результатом

HTML код страницы с результатами (файл result.html) представлен ниже (Рис. 7).

```
<!DOCTYPE html>
<html>
  <head>
    <title>Результат</title>
    <script src="prog.js"></script>
  </head>
  <body onload="printScore()">
    <h1>Время вышло!</h1>
    <p id="result"></p>
    <button onclick="restart()">Начать заново</button>
  </body>
</html>
```

Рисунок 7 – HTML код страницы с результатами

Все вышеуказанные файлы связаны между собой с помощью скрипта «prog.js». В основном в данном файле прописана работа генерации алгебраических примеров, работа таймера, правильность введенного ответа, функция перехода на страницу result.html и функция перезапуска программы.

Содержание файла «prog.js» выглядит следующим образом (Рис.8,9,10):

```
let timeLeft = 120;
let score = 0;
let rightAnswer = 0;
let equation = "";

function start() {
    localStorage.setItem('points', 0);
    location.href = "math.html";
}

function main() {
    displayTime();
    displayMath();
}

function displayTime() {
    const timer = document.getElementById("timer");
    const minutes = Math.floor(timeLeft / 60);
    let seconds = timeLeft % 60;
    if (seconds < 10) {
        seconds = "0" + seconds;
    }
    timer.innerHTML = "Времени осталось: " + minutes + ":" + seconds;
    if (timeLeft === 0) {
        end();
    } else {
        timeLeft--;
        setTimeout(displayTime, 1000);
    }
}
```

Рисунок 8 – Содержание файла «prog.js»

```
function displayMath() {
    const equationElement = document.getElementById("equation");
    const arithmeticSigns = ['+', '-', '*', '/'];
    const maxLength = 4;
    equation = "";
    const len = Math.floor(Math.random() * (maxLength - 2 + 1)) + 2; //2 4
    for (let i=0;i<Math.floor(len/2);i++) {
        let num1 = Math.floor(Math.random() * 100) + 1; //1 100
        let num2 = Math.floor(Math.random() * 100) + 1; //1 100
        let sign = arithmeticSigns[Math.floor(Math.random() * 4)];
        equation += num1 + " " + sign + " " + num2;
        if (i !== Math.floor(len/2)-1) {
            sign = arithmeticSigns[Math.floor(Math.random() * 4)];
            equation += " " + sign + " ";
        }
    }
    equationElement.innerHTML = equation;
}

function checkAnswer() {
    const answerUser = document.getElementById("answer").value;
    const scoreUser = document.getElementById("score");
    if (answerUser == Math.round(eval(equation))) {
        score += 1;
    }
    scoreUser.innerHTML = "Решено примеров: " + score;
    document.getElementById("answer").value = "";
    displayMath();
}
```

Рисунок 9 – Содержание файла «prog.js»

```
function printScore() {
    const points = localStorage.getItem('points');
    const resultText = document.getElementById("result");
    resultText.innerHTML = "Ваш результат: " + points;
}

function end() {
    localStorage.setItem('points', score);
    location.href = "result.html";
}

function restart() {
    location.href = "index.html";
}
```

Рисунок 10 – Содержание файла «prog.js»

Данную систему можно легко изменить и добавить новые элементы, которые будут отличительными особенностями этого приложения, например таблица рекордов, тренировки (решение примеров на определённые темы [стандартные арифметические операции, примеры с корнями и степенями, тригонометрия и др.]) и новый тип сложности – адаптивный.

Адаптивный тип сложности заключается в следующем: на старте у пользователя есть шкала уровня сложности, значение которой установлено на 50%. При каждом правильном решении показатель шкалы увеличивается и наоборот – при каждой ошибке сложность понижается.

Первым этапом добавления такой сложности стало изменение главной страницы (Рис. 11).



Рисунок 11. Изменённая главная страница

Также необходимо изменить подход к подсчёту конечного результата. Так как примеры больше не зафиксированы на одном шаблоне, то необходимо создать отдельную от стандартной сложности систему подсчёта конечного результата. Система построена на двух формулах – изменении сложности и вычислении добавляемых пользователю баллов. Оба показателя зависят от следующих параметров:

1. Количество правильно решённых примеров подряд ( $St$ ).
2. Сложность ( $Diff$ ).
3. Общее количество правильно решённых примеров ( $R$ ).
4. Общее количество неправильно решённых примеров ( $W$ ).

Формула подсчёта баллов была построена на основе выбранных параметров и выглядит следующим образом:

$$points = \frac{Diff + St}{5} + |R - W|$$

Данная формула позволяет достигнуть наибольших результатов при решении сложных результатов (от 40 до 50 баллов за правильно решённый пример) и наименьших при лёгкой сложности (около 10 за каждое правильное решение). Формула позволяет динамически изменять сложность, не слишком сильно повышая и понижая показатель, если пользователь допустил единичную ошибку. Если допустить слишком много ошибок подряд, то снижение сложности замедлится и при этом восстановить показатель сложности на прежний уровень будет также сложно.

Формула изменения сложности выглядит следующим образом:

$$Diff_{\pm} = \frac{Diff + St}{5} - |W + R + St + 1|$$

На странице решения примеров была добавлена строка прогресса, которая изменяется в зависимости от успехов пользователя (заполняется, когда она решает примеры и наоборот). Эта строка обозначает уровень сложности (Рис. 12).



## Рисунок 12. Страница решения примеров при адаптивной сложности

Также в системе была реализована таблица достижений для пользователей. Первым делом была изменена страница с выводом результатов решения примеров (Рис. 13).

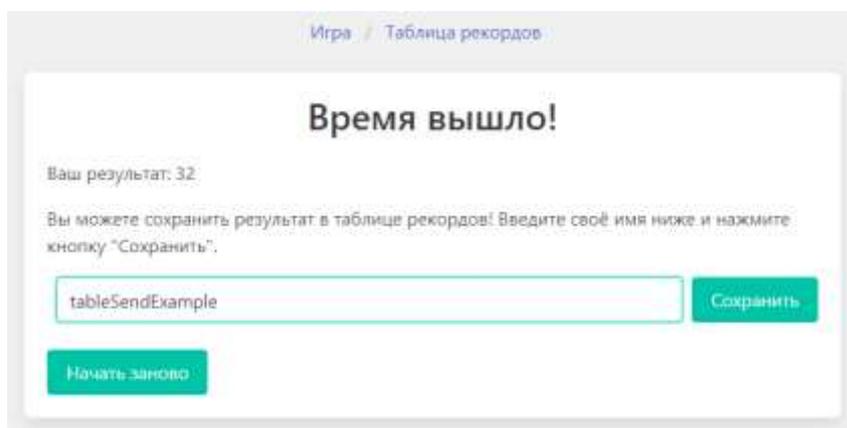


Рисунок 13. Изменённая страница вывода результатов

На странице была добавлена строка ввода имени пользователя. По нажатию на кнопку “Сохранить” данные пользователя сохраняются в JSON файле, структура которого состоит из следующих элементов: “userName” – имя пользователя и “score” – результат пользователя. Формирование JSON файла происходит за счёт использования языка программирования PHP (Рис. 14).

Также стоит отметить, что на странице формируется 2 таблицы, между которыми переключается пользователь в зависимости от интересующего типа сложности. Переключение между таблицами можно сделать с помощью JS и метода “element.style.display”, где element – любой элемент с определённым пользователем ID. Данный метод позволяет скрыть содержимое страницы, достаточно задать значение свойства “none”.

```

<?php
function saveData() {
    $table = $_POST['gameType'];
    $tableFileName = "table".$table.".json";

    $dataJson = file_get_contents($tableFileName);
    $data = json_decode($dataJson,true);
    print_r($data['users']);
    $newData = array();
    foreach($data['users'] as $user) {
        $newData[] = array("userName"=>$user['userName'], "score"=>$user['score']);
    }
    $newData[] = array("userName"=>$_POST['userName'], "score"=>$_POST['score']);

    $updatedJsonData = json_encode(array("users" => $newData),JSON_FORCE_OBJECT);
    file_put_contents($tableFileName, $updatedJsonData);

    echo "<script>location.href='index.html'</script>";
}

if (isset($_POST['submit'])) {
    saveData();
}
?>

```

Рисунок 14. Алгоритм формирования JSON файла

Следующим этапом в создании таблицы рекордов стало написание алгоритма генерации таблицы. Генерация таблицы происходит с помощью языка программирования PHP. Важно также учесть, что данные в таблице должны быть отсортированы по убыванию и выводиться в таблицу должны только первые 10 участников (Рис. 15).

Сортировка пользователей в таблице достигается за счёт использования функции “usort”. Функция сортирует массив по значениям используя пользовательскую функцию для сравнения элементов:

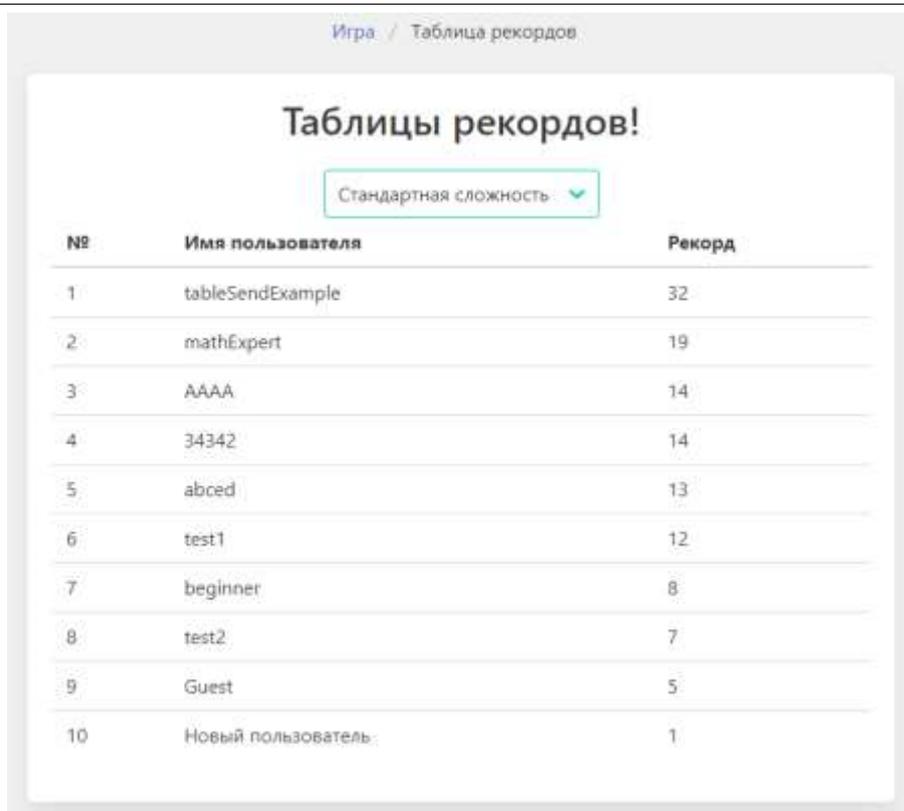
```
usort($data, function($a, $b) {  
    return $a['integerParam'] - $b['integerParam'];  
});
```

Функция позволяет сортировать не только целочисленные значения, но также и строки. Но для того, чтобы сортировка работала, необходимо использовать функцию сравнения строк “strcmp”.

```
<table id="mainTable1" class="table">  
    <thead>  
        <tr>  
            <th>№</th>  
            <th>Имя пользователя</th>  
            <th>Рекорд</th>  
        </tr>  
    </thead>  
    <tbody id="tableBody">  
        <?php  
            $table = fopen("table1.json", "r");  
            $jsonData = file_get_contents("table1.json");  
            $data = json_decode($jsonData, true);  
            usort($data['users'], function($b, $a) {  
                return $a['score'] - $b['score'];  
            });  
            $tableLen = 10;  
            if (count($data['users']) < $tableLen) { $tableLen = count($data['users']);}  
            for ($i=0; $i<$tableLen; $i++) {  
                echo "<tr>";  
                echo "<td>".($i+1)."</td>";  
                echo "<td>".$data['users'][$i]['userName']."</td>";  
                echo "<td>".$data['users'][$i]['score']."</td>";  
                echo "</tr>";  
            }  
        ?>  
    </tbody>  
</table>
```

Рисунок 15. Алгоритм генерации таблицы рекордов

Таким образом была создана отдельная страница, где будут выводиться лучшие результаты пользователей и при этом получилось реализовать данный функционал без использования баз данных (Рис. 16).



№	Имя пользователя	Рекорд
1	tableSendExample	32
2	mathExpert	19
3	AAAA	14
4	34342	14
5	abced	13
6	test1	12
7	beginner	8
8	test2	7
9	Guest	5
10	Новый пользователь	1

Рисунок 16. Страница таблицы рекордов

Результаты пользователей также распределяются по двум таблицам (так как подход к подсчёту конечного результата на сложностях различается) и из выпадающего списка над таблицей можно выбрать интересующий тип сложности.

### Выводы

Таким образом, полученный результат является полезным для тех, кто желает научиться решать арифметические примеры быстро и точно. Система может использоваться как для обучения, так и для развлечения и представляет собой полезный инструмент для развития математических навыков.

Регулярное использование системы для решения случайных арифметических примеров на время способствует развитию когнитивных навыков, таких как концентрация внимания, работа с числами, аналитическое мышление и способность к быстрому принятию решений. Эти навыки являются важными в повседневной жизни и в профессиональной деятельности.

Система для решения случайных арифметических примеров на время может быть применена в различных сферах, включая игровую индустрию и программы тренировки памяти. Она предоставляет возможность улучшить навыки быстрого и точного решения математических примеров, а также помогает тренировать мозг на логическое мышление и принятие решений в ограниченное время.

**Библиографический список**

1. Калимулин Ф.Н. и др. Разработка приложения, обеспечивающего расчёт арифметического выражения, заданного в виде символьной строки // Modern Science. 2021. № 1-2. С. 458-463.
2. Дементьева А.И., Лубягина Е.Н., Шабалин Д.В. Разработка специализированных сайтов по математике и информатике // Математический вестник Вятского государственного университета. 2021. № 4 (23). С. 33-45.
3. Широкова Т.А. Формирование вычислительных навыков на уроках математики // Педагогический форум. 2020. № 2 (6). С. 28-29.
4. Качков М.С., Пахомов П.А., Горин И.А. Выбор инструментальных средств для разработки образовательного веб-приложения // Известия Тульского государственного университета. Технические науки. 2023. № 1. С. 58-62.
5. Крокфорд Д. Как устроен JavaScript. СПб.: Питер, 2019. 304 с.
6. Минник К. JavaScript для чайников. М.: Диалектика, 2019. 320 с.
7. Фримен Э. Изучаем программирование на JavaScript. СПб.: Питер, 2015. 635 с.
8. Symbolab – онлайн-калькулятор URL:  
<https://ru.symbolab.com/practice/order-of-operations-whole-practice> (дата обращения: 04.05.2023).