Министерство образования Республики Беларусь

Учреждение образования

Белорусский Государственный университет информатики и радиоэлектроники

Инженерно-экономический факультет

Кафедра экономической информатики

Отчет

по лабораторной работе № 2

«Работа с коллекциями и встроенными структурами данных»

|  |  |
| --- | --- |
| Выполнил: | Летко А. С. |
|  | студент группы 377901 |
| Проверил: | Сторожев Д.А. |
|  | канд.техн.наук, доцент |

Минск 2025

**Цель:** ознакомится с основными коллекциями и встроенными структурами данных в PHP, а также обучить их эффективному применению функций обработки массивов и возможностей SPL-классов для построения иерархичных, приоритетных и ленивых коллекций

**Задачи:**

− изучить синтаксис и семантику индексных, ассоциативных и многомерных массивов;

− освоить встроенные функции для работы с массивами: array\_map, array\_filter, array\_reduce, сортировочные функции и операции над множествами;

− разобрать и применить SPL-структуры данных: SplStack, SplQueue, SplHeap, SplFixedArray, SplObjectStorage;

− научиться создавать и использовать собственные итераторы и генераторы на основе интерфейсов Iterator и Generator;

− реализовать на практике алгоритмы обработки коллекций (поиск, фильтрация, агрегация, приоритетная очередь);

− сформировать навык проектирования кодовой архитектуры с явным разделением инструкций по работе с данными и алгоритмической логики;

− закрепить умение документировать контракт коллекций с помощью phpDoc и строгой типизации (array, iterable, Generator).

**Теоретическая часть.**

PHP предоставляет богатый набор коллекций “из коробки”: универсальные массивы, структуры данных SPL, итераторы и генераторы. В связке с встроенными функциями для массивов это покрывает большинство прикладных сценариев – от простых списков до приоритетных очередей и ленивых потоков данных.

$indexArray = [10, 20, 30, 40];

echo "<strong>Индексный массив:</strong> ";

print\_r($indexArray);

Массив в PHP объединяет динамический список и хеш-таблицу: он может быть индексным (целочисленные ключи) и ассоциативным (строковые ключи), а также многомерным.

// Ассоциативные массивы

$assocArray = [

"name" => "Иван",

"age" => 25,

"city" => "Минск"

];

echo "<strong>Ассоциативный массив:</strong> ";

print\_r($assocArray);

// Многомерные массивы

$multiArray = [

"user1" => ["name" => "Анна", "age" => 30],

"user2" => ["name" => "Петр", "age" => 25],

"matrix" => [

[1, 2, 3],

[4, 5, 6],

[7, 8, 9]

]

];

echo "<strong>Многомерный массив:</strong> ";

print\_r($multiArray);

**Индивидуальное задание.**

Вариант 8:

1. Задача (массивы): найти ключи массива, значения которых больше заданного порога.

Листинг кода.

index.php:

<?php

declare(strict\_types=1);

function findKeysAboveThreshold(array $array, $threshold): array {

$result = [];

foreach ($array as $key => $value) {

if ($value > $threshold) {

$result[] = $key;

}

}

return $result;

}

$originalArray = [

0 => 10,

1 => 25,

2 => 8,

3 => 45,

4 => 15,

5 => 32,

6 => 18,

7 => 50,

8 => 5,

9 => 28,

10 => 12,

11 => 38,

12 => 22,

13 => 41,

14 => 7

];

$threshold = 20;

$keysAboveThreshold = [];

$submitted = false;

if ($\_SERVER['REQUEST\_METHOD'] === 'POST' && isset($\_POST['threshold'])) {

$submitted = true;

$threshold = (int)$\_POST['threshold'];

$keysAboveThreshold = findKeysAboveThreshold($originalArray, $threshold);

}

$minValue = min($originalArray);

$maxValue = max($originalArray);

$averageValue = array\_sum($originalArray) / count($originalArray);

?>

<!DOCTYPE html>

<html lang="ru">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Задание 1 - Ключи выше порога</title>

<link rel="stylesheet" href="css/style.css">

</head>

<body>

<div class="container">

<div class="header">

<h1>Лабораторная работа №2</h1>

<p>Задание 1 - Поиск ключей массива выше порога</p>

</div>

<div class="task-description">

<h2>📋 Описание задания</h2>

<p>Найти ключи массива, значения которых больше заданного порога. Реализовано без использования встроенных функций поиска ключей.</p>

</div>

<div class="array-display">

<h3>🔢 Исходный массив для анализа</h3>

<div class="array-items">

<?php foreach ($originalArray as $key => $value): ?>

<div class="array-item">

<?= $value ?>

<div class="array-index">[<?= $key ?>]</div>

</div>

<?php endforeach; ?>

</div>

<div class="array-stats">

<div class="stat-item">

<div class="stat-value"><?= $minValue ?></div>

<div class="stat-label">Минимальное значение</div>

</div>

<div class="stat-item">

<div class="stat-value"><?= number\_format($maxValue, 1) ?></div>

<div class="stat-label">Максимальное значение</div>

</div>

<div class="stat-item">

<div class="stat-value"><?= number\_format($averageValue, 1) ?></div>

<div class="stat-label">Среднее значение</div>

</div>

<div class="stat-item">

<div class="stat-value"><?= count($originalArray) ?></div>

<div class="stat-label">Всего элементов</div>

</div>

</div>

</div>

<div class="form-section">

<h3 style="text-align: center; margin-bottom: 20px; color: #333;">🎯 Установите пороговое значение</h3>

<form method="POST">

<div class="form-group">

<label for="threshold">Пороговое значение:</label>

<input type="number" id="threshold" name="threshold"

value="<?= htmlspecialchars((string)$threshold) ?>"

step="1" required

placeholder="Введите любое число">

<small style="display: block; margin-top: 5px; color: #666;">

Диапазон значений в массиве: от <?= $minValue ?> до <?= $maxValue ?>

</small>

</div>

<button type="submit" class="btn">

🔍 Найти ключи выше порога

</button>

</form>

</div>

<?php if ($submitted): ?>

<div class="results-section">

<h2>📊 Результаты поиска</h2>

<div class="threshold-display">

Пороговое значение: <strong><?= $threshold ?></strong>

</div>

<?php if (!empty($keysAboveThreshold)): ?>

<div class="keys-result">

<h3>✅ Найдено ключей: <?= count($keysAboveThreshold) ?></h3>

<p>Ключи со значениями больше <?= $threshold ?>:</p>

<div class="keys-list">

<?php foreach ($keysAboveThreshold as $key): ?>

<div class="key-item">

Ключ [<?= $key ?>] = <?= $originalArray[$key] ?>

</div>

<?php endforeach; ?>

</div>

</div>

<?php else: ?>

<div class="no-keys">

<h3>❌ Ключи не найдены</h3>

<p>В массиве нет элементов со значениями больше <?= $threshold ?></p>

</div>

<?php endif; ?>

<div class="array-values">

<h4 style="text-align: center; margin-bottom: 15px;">📈 Все значения массива:</h4>

<?php foreach ($originalArray as $key => $value): ?>

<div class="value-item">

<span class="value-key">Ключ [<?= $key ?>]:</span>

<span class="value-number <?= $value > $threshold ? 'value-above' : 'value-below' ?>">

<?= $value ?>

<?= $value > $threshold ? '✅' : '❌' ?>

</span>

</div>

<?php endforeach; ?>

</div>

</div>

<?php endif; ?>

<div style="text-align: center; margin-top: 30px; color: #666; font-size: 0.9em;">

<p>Лабораторная работа по высокоуровневым языкам программирования</p>

<p>Вариант 8 - Задание 1</p>

</div>

</div>

</body>

</html>

Изображение выглядит как текст, программное обеспечение, Мультимедийное программное обеспечение, Значок на компьютере

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 1

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 2

2. Задача (SPL): смоделировать стек операций (LIFO) с помощью SplDoublyLinkedList.

Листинг кода.

index.php:

<?php

declare(strict\_types=1);

class OperationStack {

private $stack;

private $operationLog;

public function \_\_construct() {

$this->stack = new SplDoublyLinkedList();

$this->operationLog = [];

$this->logOperation("Инициализация", "Стек операций создан");

}

public function push(string $operation): void {

$this->stack->push($operation);

$this->logOperation("PUSH", "Добавлена операция: '{$operation}'");

}

public function pop(): ?string {

if ($this->stack->isEmpty()) {

$this->logOperation("POP", "Попытка извлечения из пустого стека", "warning");

return null;

}

$operation = $this->stack->pop();

$this->logOperation("POP", "Извлечена операция: '{$operation}'", "success");

return $operation;

}

public function peek(): ?string {

if ($this->stack->isEmpty()) {

return null;

}

return $this->stack->top();

}

public function isEmpty(): bool {

return $this->stack->isEmpty();

}

public function getAllOperations(): array {

$operations = [];

$this->stack->rewind();

while ($this->stack->valid()) {

$operations[] = $this->stack->current();

$this->stack->next();

}

return $operations;

}

public function count(): int {

return $this->stack->count();

}

public function clear(): void {

$countBefore = $this->count();

$this->stack = new SplDoublyLinkedList();

$this->logOperation("CLEAR", "Стек очищен (удалено {$countBefore} операций)", "danger");

}

private function logOperation(string $type, string $message, string $status = "info"): void {

// Инициализируем лог, если он еще не инициализирован

if (!is\_array($this->operationLog)) {

$this->operationLog = [];

}

$this->operationLog[] = [

'timestamp' => date('H:i:s'),

'type' => $type,

'message' => $message,

'status' => $status

];

if (count($this->operationLog) > 50) {

array\_shift($this->operationLog);

}

}

public function getOperationLog(): array {

if (!is\_array($this->operationLog)) {

$this->operationLog = [];

}

return $this->operationLog;

}

public function getStats(): array {

return [

'total\_operations' => count($this->getOperationLog()),

'stack\_size' => $this->count(),

'is\_empty' => $this->isEmpty(),

'top\_operation' => $this->peek()

];

}

public function \_\_wakeup() {

if (!is\_array($this->operationLog)) {

$this->operationLog = [];

}

}

}

if (session\_status() === PHP\_SESSION\_NONE) {

session\_start();

}

if (!isset($\_SESSION['operation\_stack'])) {

$\_SESSION['operation\_stack'] = new OperationStack();

}

$stack = $\_SESSION['operation\_stack'];

if (!($stack instanceof OperationStack)) {

$\_SESSION['operation\_stack'] = new OperationStack();

$stack = $\_SESSION['operation\_stack'];

}

$message = '';

$messageType = '';

if ($\_SERVER['REQUEST\_METHOD'] === 'POST') {

if (isset($\_POST['push\_operation'])) {

$operation = trim($\_POST['operation']);

if (!empty($operation)) {

$stack->push($operation);

$message = "Операция '{$operation}' добавлена в стек";

$messageType = 'success';

} else {

$message = "Введите название операции";

$messageType = 'warning';

}

}

elseif (isset($\_POST['pop\_operation'])) {

$popped = $stack->pop();

if ($popped !== null) {

$message = "Операция '{$popped}' извлечена из стека";

$messageType = 'success';

} else {

$message = "Стек пуст - нечего извлекать";

$messageType = 'warning';

}

}

elseif (isset($\_POST['clear\_stack'])) {

$stack->clear();

$message = "Стек очищен";

$messageType = 'danger';

}

elseif (isset($\_POST['add\_sample\_operations'])) {

$sampleOperations = [

"Сложение чисел",

"Умножение матриц",

"Проверка условий",

"Чтение файла",

"Запись в базу данных",

"Отправка email",

"Валидация данных",

"Генерация отчета"

];

foreach ($sampleOperations as $operation) {

$stack->push($operation);

}

$message = "Добавлены примеры операций";

$messageType = 'success';

}

}

$operations = $stack->getAllOperations();

$operationLog = $stack->getOperationLog();

$stats = $stack->getStats();

?>

<!DOCTYPE html>

<html lang="ru">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Задание 2 - Стек операций</title>

<link rel="stylesheet" href="css/style.css">

</head>

<body>

<div class="container">

<div class="header">

<h1>Лабораторная работа №2</h1>

<p>Задание 2 - Моделирование стека операций (LIFO)</p>

</div>

<div class="task-description">

<h2>📋 Описание задания</h2>

<p>Смоделировать обслуживание операций в стеке (LIFO - Last In, First Out) с помощью SplDoublyLinkedList.

Стек - это структура данных, в которой последний добавленный элемент извлекается первым.</p>

</div>

<?php if ($message): ?>

<div class="log-item <?= $messageType ?>" style="margin-bottom: 20px;">

<span><?= htmlspecialchars($message) ?></span>

<span class="log-time"><?= date('H:i:s') ?></span>

</div>

<?php endif; ?>

<div class="controls-grid">

<!-- Форма добавления операции -->

<div class="control-card">

<h3>➕ Добавить операцию</h3>

<form method="POST">

<div class="form-group">

<label for="operation">Название операции:</label>

<input type="text" id="operation" name="operation"

placeholder="Введите название операции..." required>

</div>

<button type="submit" name="push\_operation" class="btn">

📥 Добавить в стек

</button>

</form>

</div>

<div class="control-card">

<h3>⚙️ Управление стеком</h3>

<form method="POST" style="display: grid; gap: 10px;">

<button type="submit" name="pop\_operation" class="btn btn-warning"

<?= $stack->isEmpty() ? 'disabled' : '' ?>>

📤 Извлечь операцию (POP)

</button>

<button type="submit" name="add\_sample\_operations" class="btn btn-success">

🎯 Добавить примеры операций

</button>

<button type="submit" name="clear\_stack" class="btn btn-danger"

<?= $stack->isEmpty() ? 'disabled' : '' ?>>

🗑️ Очистить стек

</button>

</form>

</div>

</div>

<div class="operations-display">

<h3>🏗️ Визуализация стека операций (LIFO)</h3>

<div class="stack-visualization">

<?php if (!$stack->isEmpty()): ?>

<?php foreach (array\_reverse($operations) as $index => $operation): ?>

<div class="stack-item">

<?= htmlspecialchars($operation) ?>

<div style="font-size: 0.8em; opacity: 0.8; margin-top: 5px;">

Позиция: <?= count($operations) - $index ?>

</div>

</div>

<?php endforeach; ?>

<div class="stack-base"></div>

<?php else: ?>

<div class="empty-stack">

📭 Стек пуст<br>

<small>Добавьте операции, чтобы увидеть их здесь</small>

</div>

<?php endif; ?>

</div>

<div class="stats-grid">

<div class="stat-card">

<div class="stat-value"><?= $stats['stack\_size'] ?></div>

<div class="stat-label">Операций в стеке</div>

</div>

<div class="stat-card">

<div class="stat-value"><?= $stats['total\_operations'] ?></div>

<div class="stat-label">Всего операций</div>

</div>

<div class="stat-card">

<div class="stat-value">

<?= $stats['is\_empty'] ? 'Да' : 'Нет' ?>

</div>

<div class="stat-label">Стек пуст</div>

</div>

<div class="stat-card">

<div class="stat-value">

<?= $stats['top\_operation'] ? '✅' : '❌' ?>

</div>

<div class="stat-label">Верхний элемент</div>

</div>

</div>

<!-- Последние операции -->

<?php if (!$stack->isEmpty()): ?>

<div class="recent-operations">

<h4>📋 Операции в стеке (сверху вниз):</h4>

<div class="operations-list">

<?php foreach (array\_reverse($operations) as $operation): ?>

<div class="operation-tag">

<?= htmlspecialchars($operation) ?>

</div>

<?php endforeach; ?>

</div>

</div>

<?php endif; ?>

</div>

<!-- Лог операций -->

<div class="operations-display">

<h3>📝 Журнал операций</h3>

<div class="operations-log">

<?php if (!empty($operationLog)): ?>

<?php foreach (array\_reverse($operationLog) as $log): ?>

<div class="log-item <?= $log['status'] ?>">

<div>

<strong>[<?= $log['type'] ?>]</strong>

<?= htmlspecialchars($log['message']) ?>

</div>

<span class="log-time"><?= $log['timestamp'] ?></span>

</div>

<?php endforeach; ?>

<?php else: ?>

<div class="empty-stack" style="padding: 20px;">

Журнал операций пуст

</div>

<?php endif; ?>

</div>

</div>

<div style="text-align: center; margin-top: 30px; color: #666; font-size: 0.9em;">

<p>Лабораторная работа по высокоуровневым языкам программирования</p>

<p>Вариант 8 - Задание 2 | SplDoublyLinkedList | LIFO</p>

</div>

</div>

</body>

</html>

Изображение выглядит как текст, программное обеспечение, Значок на компьютере, Мультимедийное программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 3

Изображение выглядит как текст, программное обеспечение, Мультимедийное программное обеспечение, Значок на компьютере

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 4

3. Задача (HTML/CSS/PHP): отобразить массив в

сетке, дубликаты подчеркнуть стилем text-decoration: underline.

Листинг кода.

index.php:

<?php

declare(strict\_types=1);

function analyzeArray(array $array): array {

$result = [

'original' => $array,

'unique' => array\_unique($array),

'duplicates' => [],

'duplicate\_info' => [],

'stats' => []

];

$valueCounts = array\_count\_values($array);

foreach ($valueCounts as $value => $count) {

if ($count > 1) {

$result['duplicates'][] = $value;

$result['duplicate\_info'][$value] = $count;

}

}

$result['stats'] = [

'total\_elements' => count($array),

'unique\_elements' => count($result['unique']),

'duplicate\_count' => count($array) - count($result['unique']),

'duplicate\_types' => count($result['duplicates']),

'most\_common' => array\_keys($valueCounts, max($valueCounts))[0] ?? null,

'most\_common\_count' => max($valueCounts) ?: 0

];

return $result;

}

$defaultArray = [

'apple', 'banana', 'orange', 'apple', 'grape',

'banana', 'kiwi', 'mango', 'orange', 'pear',

'apple', 'kiwi', 'pineapple', 'banana', 'grape'

];

$customArray = $defaultArray;

$analysis = analyzeArray($defaultArray);

$submitted = false;

if ($\_SERVER['REQUEST\_METHOD'] === 'POST') {

$submitted = true;

if (isset($\_POST['use\_default'])) {

$customArray = $defaultArray;

$analysis = analyzeArray($defaultArray);

}

elseif (isset($\_POST['custom\_array'])) {

$input = trim($\_POST['custom\_array']);

if (!empty($input)) {

$customArray = preg\_split('/[\s,]+/', $input);

$customArray = array\_map('trim', $customArray);

$customArray = array\_filter($customArray, function($value) {

return $value !== '';

});

$customArray = array\_values($customArray);

if (!empty($customArray)) {

$analysis = analyzeArray($customArray);

} else {

$customArray = $defaultArray;

$analysis = analyzeArray($defaultArray);

}

}

}

elseif (isset($\_POST['add\_sample'])) {

$sampleData = [

'JavaScript', 'Python', 'Java', 'C++', 'PHP',

'Python', 'JavaScript', 'Ruby', 'Go', 'PHP',

'Rust', 'Java', 'TypeScript', 'Python', 'C#'

];

$customArray = $sampleData;

$analysis = analyzeArray($sampleData);

}

}

$stats = $analysis['stats'];

?>

<!DOCTYPE html>

<html lang="ru">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Задание 3 - Анализ массива с дубликатами</title>

<link rel="stylesheet" href="css/style.css">

</head>

<body>

<div class="container">

<div class="header">

<h1>Лабораторная работа №2</h1>

<p>Задание 3 - Отображение массива с подсветкой дубликатов</p>

</div>

<div class="task-description">

<h2>📋 Описание задания</h2>

<p>Отобразить массив в сетке, дубликаты подчеркнуть стилем. Показать визуальное представление исходного массива с выделением повторяющихся элементов.</p>

</div>

<div class="controls-section">

<h3>🎛️ Управление массивом</h3>

<div class="controls-grid">

<form method="POST">

<div class="form-group">

<label for="custom\_array">Введите свой массив (через запятую или пробел):</label>

<textarea id="custom\_array" name="custom\_array"

placeholder="например: apple, banana, orange, apple, grape, banana"

rows="4"><?= htmlspecialchars(implode(', ', $customArray)) ?></textarea>

</div>

<button type="submit" name="custom\_submit" class="btn">

🔍 Проанализировать массив

</button>

</form>

<div style="display: flex; flex-direction: column; gap: 10px;">

<form method="POST">

<button type="submit" name="use\_default" class="btn btn-success">

🍎 Использовать фруктовый массив

</button>

</form>

<form method="POST">

<button type="submit" name="add\_sample" class="btn">

💻 Использовать языки программирования

</button>

</form>

</div>

</div>

</div>

<div class="stats-grid">

<div class="stat-card">

<div class="stat-value"><?= $stats['total\_elements'] ?></div>

<div class="stat-label">Всего элементов</div>

</div>

<div class="stat-card">

<div class="stat-value"><?= $stats['unique\_elements'] ?></div>

<div class="stat-label">Уникальных элементов</div>

</div>

<div class="stat-card">

<div class="stat-value"><?= $stats['duplicate\_count'] ?></div>

<div class="stat-label">Дубликатов</div>

</div>

<div class="stat-card">

<div class="stat-value"><?= $stats['duplicate\_types'] ?></div>

<div class="stat-label">Типов дубликатов</div>

</div>

</div>

<div class="array-display-section">

<h3 class="section-title">📊 Визуализация массивов</h3>

<div class="array-grid">

<!-- Исходный массив -->

<div class="array-box">

<h4>🎯 Исходный массив</h4>

<div class="array-items">

<?php if (!empty($analysis['original'])): ?>

<?php foreach ($analysis['original'] as $index => $item): ?>

<div class="array-item <?= in\_array($item, $analysis['duplicates']) ? 'duplicate' : '' ?>">

<?= htmlspecialchars($item) ?>

<div class="array-index">[<?= $index ?>]</div>

</div>

<?php endforeach; ?>

<?php else: ?>

<div class="empty-state">Массив пуст</div>

<?php endif; ?>

</div>

</div>

<div class="array-box">

<h4>✅ Уникальные элементы</h4>

<div class="array-items">

<?php if (!empty($analysis['unique'])): ?>

<?php foreach ($analysis['unique'] as $index => $item): ?>

<div class="array-item unique">

<?= htmlspecialchars($item) ?>

<div class="array-index">[<?= $index ?>]</div>

</div>

<?php endforeach; ?>

<?php else: ?>

<div class="empty-state">Нет уникальных элементов</div>

<?php endif; ?>

</div>

</div>

</div>

<div class="color-legend">

<div class="legend-item">

<div class="legend-color legend-normal"></div>

<span>Уникальные элементы (встречаются 1 раз)</span>

</div>

<div class="legend-item">

<div class="legend-color legend-duplicate"></div>

<span>Дубликаты (встречаются 2+ раза)</span>

</div>

<div class="legend-item">

<div class="legend-color legend-unique"></div>

<span>Только уникальные элементы</span>

</div>

</div>

</div>

<div class="duplicate-analysis">

<h4>🔍 Детальный анализ дубликатов</h4>

<?php if (!empty($analysis['duplicates'])): ?>

<div class="duplicate-list">

<?php foreach ($analysis['duplicate\_info'] as $value => $count): ?>

<div class="duplicate-item">

<?= htmlspecialchars($value) ?>

<span class="duplicate-count">×<?= $count ?></span>

</div>

<?php endforeach; ?>

</div>

<?php if ($stats['most\_common']): ?>

<div style="text-align: center; margin-top: 20px; padding: 15px; background: #fff3cd; border-radius: 8px;">

<strong>🏆 Самый частый элемент:</strong>

"<?= htmlspecialchars($stats['most\_common']) ?>"

(встречается <?= $stats['most\_common\_count'] ?> раз)

</div>

<?php endif; ?>

<?php else: ?>

<div class="no-duplicates">

✅ В массиве нет дубликатов - все элементы уникальны!

</div>

<?php endif; ?>

</div>

<div class="array-comparison">

<div class="comparison-box">

<h4>📈 Эффективность массива</h4>

<div style="text-align: center; margin: 20px 0;">

<div style="font-size: 3em; color: #667eea; margin-bottom: 10px;">

<?= round(($stats['unique\_elements'] / $stats['total\_elements']) \* 100, 1) ?>%

</div>

<div style="color: #666;">Уникальность данных</div>

</div>

<div style="background: #e9ecef; height: 20px; border-radius: 10px; overflow: hidden; margin: 20px 0;">

<div style="background: linear-gradient(135deg, #28a745 0%, #20c997 100%);

height: 100%; width: <?= ($stats['unique\_elements'] / $stats['total\_elements']) \* 100 ?>%;">

</div>

</div>

<div style="display: flex; justify-content: space-between; font-size: 0.9em; color: #666;">

<span>Уникальные: <?= $stats['unique\_elements'] ?></span>

<span>Дубликаты: <?= $stats['duplicate\_count'] ?></span>

</div>

</div>

<div class="comparison-box">

<h4>📋 Сводка данных</h4>

<div style="line-height: 2;">

<div><strong>Всего элементов:</strong> <?= $stats['total\_elements'] ?></div>

<div><strong>Уникальных:</strong> <?= $stats['unique\_elements'] ?></div>

<div><strong>Дубликатов:</strong> <?= $stats['duplicate\_count'] ?></div>

<div><strong>Типов дубликатов:</strong> <?= $stats['duplicate\_types'] ?></div>

<?php if ($stats['most\_common']): ?>

<div><strong>Самый частый:</strong> "<?= htmlspecialchars($stats['most\_common']) ?>"</div>

<div><strong>Количество:</strong> <?= $stats['most\_common\_count'] ?> раз</div>

<?php endif; ?>

</div>

</div>

</div>

<div style="text-align: center; margin-top: 40px; color: #666; font-size: 0.9em;">

<p>Лабораторная работа по высокоуровневым языкам программирования</p>

<p>Вариант 8 - Задание 3 | Подсветка дубликатов | CSS Grid</p>

</div>

</div>

</body>

</html>

Изображение выглядит как текст, программное обеспечение, Значок на компьютере, Мультимедийное программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 5 – Пример с массивом фруктов

Изображение выглядит как текст, программное обеспечение, Значок на компьютере, Мультимедийное программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 6 – Пример с массивом фруктов

Изображение выглядит как текст, программное обеспечение, Значок на компьютере, Мультимедийное программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 7 – Пример с массивом ЯП

Изображение выглядит как текст, программное обеспечение, Значок на компьютере, Мультимедийное программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 8 – Пример с массивом ЯП