

Федеральное государственное образовательное бюджетное учреждение  
высшего образования  
**«Финансовый университет при Правительстве  
Российской Федерации»**  
(Финансовый университет)

**Колледж информатики и программирования**

ПМ.08 Разработка кода информационных систем

Группа: ЗИСИП-622

УТВЕРЖДАЮ

Председатель предметно-цикловой комиссии информационных систем и  
программирования

\_\_\_\_\_ Т.Г. Аксёнова

«\_\_\_\_\_» \_\_\_\_\_ 2025 г.

**ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №13**

**Знакомство с РНР**

Преподаватель

\_\_\_\_\_ Р. Р. Абзалимов

Исполнитель

\_\_\_\_\_ Л. Д. Слепцов

Оценка: \_\_\_\_\_

«\_\_\_\_\_» \_\_\_\_\_ 2025 г.

Москва

2025

## Цели работы

- Познакомиться с основами языка программирования PHP и его синтаксисом
- Освоить работу с различными типами данных в PHP
- Изучить основные управляющие конструкции (условия и циклы)
- Освоить работу с массивами и функциями в PHP
- Получить базовые навыки объектно-ориентированного программирования в PHP
- Научиться работать с файлами и строками в PHP
- Понять принципы обработки и валидации данных в PHP
- Изучить основы работы с датой и временем
- Развить навыки отладки и поиска ошибок в PHP-коде
- Сформировать практические навыки решения типовых задач с использованием PHP

## Ход работы

### Листинг php13.php

```
<!DOCTYPE html>
<html lang="ru">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Лабораторная работа №13</title>
  <!-- Подключаем Bootstrap CSS -->
  <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
rel="stylesheet">
```

```

<style>
  body {
    padding: 20px;
    background-color: #f8f9fa;
  }
  .task-block {
    background-color: #fff;
    padding: 20px;
    border-radius: 10px;
    margin-bottom: 20px;
    border: 1px solid #dee2e6;
  }
  .task-block h3 {
    margin-bottom: 20px;
    color: #0d6efd;
  }
  .output-block {
    background-color: #e9ecef;
    padding: 15px;
    border-radius: 5px;
    margin-top: 15px;
  }
</style>
</head>
<body>
  <div class="container">
    <h1 class="text-center mb-4">Лабораторная работа №13</h1>

    <!-- Задание 1: Базовый вывод информации -->
    <div class="task-block">
      <h3>1. Базовый вывод информации</h3>
      <form id="task1">
        <input type="hidden" name="task_id" value="task1">
        <label>Имя: <input type="text" name="name"></label><br>
        <label>Возраст: <input type="number" name="age"></label><br>
        <label>Хобби: <input type="text" name="hobby"></label><br>
        <button type="submit">Отправить</button>
      </form>
      <div class="output-block"></div>
    </div>

    <!-- Задание 2: Переменные и типы данных -->
    <div class="task-block">
      <h3>2. Переменные и типы данных</h3>
      <form id="task2">
        <input type="hidden" name="task_id" value="task2">
        <label>Целое число: <input type="number"
name="integer"></label><br>
        <label>Строка: <input type="text" name="string"></label><br>
        <label>Массив (в формате JSON): <input type="text"
name="array"></label><br>

```

```

        <label>Булево значение: <input type="text"
name="boolean"></label><br>
        <label>Null: <input type="text" name="null"></label><br>
        <button type="submit">Отправить</button>
    </form>
    <div class="output-block"></div>
</div>

<!-- Задание 3: Арифметические операции -->
<div class="task-block">
    <h3>3. Арифметические операции</h3>
    <form id="task3">
        <input type="hidden" name="task_id" value="task3">
        <label>Число A: <input type="number" name="a"></label><br>
        <label>Число B: <input type="number" name="b"></label><br>
        <button type="submit">Отправить</button>
    </form>
    <div class="output-block"></div>
</div>

<!-- Задание 4: Логические операторы -->
<div class="task-block">
    <h3>4. Логические операторы</h3>
    <form id="task4">
        <input type="hidden" name="task_id" value="task4">
        <label>Год: <input type="number" name="year"></label><br>
        <button type="submit">Проверить</button>
    </form>
    <div class="output-block"></div>
</div>

<!-- Задание 5: Условные конструкции -->
<div class="task-block">
    <h3>5. Условные конструкции</h3>
    <form id="task5">
        <input type="hidden" name="task_id" value="task5">
        <label>Возраст: <input type="number" name="age"></label><br>
        <button type="submit">Определить</button>
    </form>
    <div class="output-block"></div>
</div>

<!-- Задание 6: Тернарный оператор -->
<div class="task-block">
    <h3>6. Тернарный оператор</h3>
    <form id="task6">
        <input type="hidden" name="task_id" value="task6">
        <label>Число: <input type="number" name="number"></label><br>
        <button type="submit">Проверить</button>
    </form>
    <div class="output-block"></div>

```

```
</div>

<!-- Задание 7: Оператор switch -->
<div class="task-block">
  <h3>7. Оператор switch</h3>
  <form id="task7">
    <input type="hidden" name="task_id" value="task7">
    <label>Номер месяца: <input type="number" name="month" min="1"
max="12"></label><br>
    <button type="submit">Показать</button>
  </form>
  <div class="output-block"></div>
</div>

<!-- Задание 8: Цикл while -->
<div class="task-block">
  <h3>8. Цикл while</h3>
  <form id="task8">
    <input type="hidden" name="task_id" value="task8">
    <label>Введите число: <input type="number"
name="limit"></label><br>
    <button type="submit">Показать</button>
  </form>
  <div class="output-block"></div>
</div>

<!-- Задание 9: Цикл do-while -->
<div class="task-block">
  <h3>9. Цикл do-while</h3>
  <form id="task9">
    <input type="hidden" name="task_id" value="task9">
    <label>Введите число: <input type="number"
name="number"></label><br>
    <button type="submit">Показать</button>
  </form>
  <div class="output-block"></div>
</div>

<!-- Задание 10: Цикл for -->
<div class="task-block">
  <h3>10. Цикл for</h3>
  <form id="task10">
    <input type="hidden" name="task_id" value="task10">
    <label>Введите число: <input type="number"
name="number"></label><br>
    <button type="submit">Показать</button>
  </form>
  <div class="output-block"></div>
</div>

<!-- Задание 11: Работа с циклами и условиями -->
```

```

<div class="task-block">
  <h3>11. Работа с циклами и условиями</h3>
  <form id="task11">
    <input type="hidden" name="task_id" value="task11">
    <label>Введите число: <input type="number"
name="limit"></label><br>
    <button type="submit">Показать простые числа</button>
  </form>
  <div class="output-block"></div>
</div>

<!-- Задание 12: Цикл foreach -->
<div class="task-block">
  <h3>12. Цикл foreach</h3>
  <form id="task12">
    <input type="hidden" name="task_id" value="task12">
    <label>Товар 1: <input type="text" name="products[]"></label>
    <label>Цена: <input type="number" name="prices[]"></label><br>
    <label>Товар 2: <input type="text" name="products[]"></label>
    <label>Цена: <input type="number" name="prices[]"></label><br>
    <label>Товар 3: <input type="text" name="products[]"></label>
    <label>Цена: <input type="number" name="prices[]"></label><br>
    <button type="submit">Показать скидку</button>
  </form>
  <div class="output-block"></div>
</div>

<!-- Задание 13: Ассоциативные массивы -->
<div class="task-block">
  <h3>13. Ассоциативные массивы</h3>
  <form id="task13">
    <input type="hidden" name="task_id" value="task13">
    <label>Имя: <input type="text" name="name"></label><br>
    <label>Возраст: <input type="number" name="age"></label><br>
    <label>Факультет: <input type="text" name="faculty"></label><br>
    <label>Средний балл: <input type="number" step="0.1"
name="average_grade"></label><br>
    <button type="submit">Проверить стипендию</button>
  </form>
  <div class="output-block"></div>
</div>

<!-- Задание 14: Многомерные массивы -->
<div class="task-block">
  <h3>14. Многомерные массивы</h3>
  <form id="task14">
    <input type="hidden" name="task_id" value="task14">
    <label>День: <input type="text" name="day"></label>
    <label>Предметы (через запятую): <input type="text"
name="subjects"></label><br>
    <button type="submit">Добавить в расписание</button>
  </form>
  <div class="output-block"></div>
</div>

```

```

        </form>
        <div class="output-block"></div>
    </div>

    <!-- Задание 15: Функции для работы с массивами -->
    <div class="task-block">
        <h3>15. Функции для работы с массивами</h3>
        <form id="task15">
            <input type="hidden" name="task_id" value="task15">
            <label>Числа (через запятую): <input type="text"
name="numbers"></label><br>
            <button type="submit">Показать результат</button>
        </form>
        <div class="output-block"></div>
    </div>

    <!-- Задание 16: Функции с параметрами -->
    <div class="task-block">
        <h3>16. Функции с параметрами</h3>
        <form id="task16">
            <input type="hidden" name="task_id" value="task16">
            <label>Числа (через запятую): <input type="text"
name="numbers"></label><br>
            <button type="submit">Показать сумму</button>
        </form>
        <div class="output-block"></div>
    </div>

    <!-- Задание 17: Функции с возвращаемым значением -->
    <div class="task-block">
        <h3>17. Функции с возвращаемым значением</h3>
        <form id="task17">
            <input type="hidden" name="task_id" value="task17">
            <label>Число: <input type="number" name="number"></label><br>
            <button type="submit">Вычислить факториал</button>
        </form>
        <div class="output-block"></div>
    </div>

    <!-- Задание 18: Область видимости переменных -->
    <div class="task-block">
        <h3>18. Область видимости переменных</h3>
        <form id="task18">
            <input type="hidden" name="task_id" value="task18">
            <label>Глобальная переменная: <input type="text"
name="global_var"></label><br>
            <button type="submit">Проверить</button>
        </form>
        <div class="output-block"></div>
    </div>

```

```
<!-- Задание 19: Анонимные функции -->
<div class="task-block">
  <h3>19. Анонимные функции</h3>
  <form id="task19">
    <input type="hidden" name="task_id" value="task19">
    <label>Числа (через запятую): <input type="text"
name="numbers"></label><br>
    <button type="submit">Показать четные числа</button>
  </form>
  <div class="output-block"></div>
</div>

<!-- Задание 20: Строковые функции -->
<div class="task-block">
  <h3>20. Строковые функции</h3>
  <form id="task20">
    <input type="hidden" name="task_id" value="task20">
    <label>Строка: <input type="text" name="string"></label><br>
    <label>Старое слово: <input type="text"
name="old_word"></label><br>
    <label>Новое слово: <input type="text"
name="new_word"></label><br>
    <button type="submit">Очистить строку</button>
  </form>
  <div class="output-block"></div>
</div>

<!-- Задание 21: Работа с датой и временем -->
<div class="task-block">
  <h3>21. Работа с датой и временем</h3>
  <form id="task21">
    <input type="hidden" name="task_id" value="task21">
    <label>Дата 1: <input type="date" name="date1"></label><br>
    <label>Дата 2: <input type="date" name="date2"></label><br>
    <button type="submit">Вычислить разницу</button>
  </form>
  <div class="output-block"></div>
</div>

<!-- Задание 22: Работа с файлами - запись -->
<div class="task-block">
  <h3>22. Работа с файлами - запись</h3>
  <form id="task22">
    <input type="hidden" name="task_id" value="task22">
    <label>Имя файла: <input type="text" name="filename"></label><br>
    <label>Содержимое: <textarea
name="content"></textarea></label><br>
    <button type="submit">Записать в файл</button>
  </form>
  <div class="output-block"></div>
</div>
```



```
<!-- Задание 23: Работа с файлами - чтение -->
<div class="task-block">
  <h3>23. Работа с файлами - чтение</h3>
  <form id="task23">
    <input type="hidden" name="task_id" value="task23">
    <label>Имя файла: <input type="text" name="filename"></label><br>
    <button type="submit">Прочитать файл</button>
  </form>
  <div class="output-block"></div>
</div>

<!-- Задание 24: ООП - создание класса -->
<div class="task-block">
  <h3>24. ООП - создание класса</h3>
  <form id="task24">
    <input type="hidden" name="task_id" value="task24">
    <label>Ширина: <input type="number" name="width"></label><br>
    <label>Высота: <input type="number" name="height"></label><br>
    <button type="submit">Вычислить</button>
  </form>
  <div class="output-block"></div>
</div>

<!-- Задание 25: ООП - наследование -->
<div class="task-block">
  <h3>25. ООП - наследование</h3>
  <form id="task25">
    <input type="hidden" name="task_id" value="task25">
    <label>Радиус круга: <input type="number"
name="radius"></label><br>
    <label>Сторона квадрата: <input type="number"
name="side"></label><br>
    <button type="submit">Вычислить</button>
  </form>
  <div class="output-block"></div>
</div>

<!-- Задание 26: ООП - полиморфизм -->
<div class="task-block">
  <h3>26. ООП - полиморфизм</h3>
  <form id="task26">
    <input type="hidden" name="task_id" value="task26">
    <label>Сторона A: <input type="number" name="a"></label><br>
    <label>Сторона B: <input type="number" name="b"></label><br>
    <label>Сторона C: <input type="number" name="c"></label><br>
    <button type="submit">Вычислить</button>
  </form>
  <div class="output-block"></div>
</div>
```

```
<!-- Задание 27: Файловый менеджер -->
<div class="task-block">
  <h3>27. Файловый менеджер</h3>
  <form id="task27">
    <input type="hidden" name="task_id" value="task27">
    <label>Имя файла: <input type="text" name="filename"></label><br>
    <button type="submit" name="action"
value="view">Просмотреть</button>
    <button type="submit" name="action"
value="delete">Удалить</button>
  </form>
  <div class="output-block"></div>
</div>

<!-- Задание 28: Валидация формы -->
<div class="task-block">
  <h3>28. Валидация формы</h3>
  <form id="task28">
    <input type="hidden" name="task_id" value="task28">
    <label>Email: <input type="email" name="email"></label><br>
    <label>Пароль: <input type="password"
name="password"></label><br>
    <label>Телефон: <input type="text" name="phone"></label><br>
    <button type="submit">Проверить</button>
  </form>
  <div class="output-block"></div>
</div>

<!-- Задание 29: Работа с JSON -->
<div class="task-block">
  <h3>29. Работа с JSON</h3>
  <form id="task29">
    <input type="hidden" name="task_id" value="task29">
    <label>Имя: <input type="text" name="name"></label><br>
    <label>Возраст: <input type="number" name="age"></label><br>
    <label>Город: <input type="text" name="city"></label><br>
    <button type="submit">Сохранить в JSON</button>
  </form>
  <div class="output-block"></div>
</div>

<!-- Задание 30: Чтение JSON -->
<div class="task-block">
  <h3>30. Чтение JSON</h3>
  <form id="task30">
    <input type="hidden" name="task_id" value="task30">
    <label>Имя файла: <input type="text" name="filename"></label><br>
    <button type="submit">Прочитать JSON</button>
  </form>
  <div class="output-block"></div>
</div>
```

```

</div>

<!-- Подключаем Bootstrap JS -->
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js
"></script>
<!-- Подключаем jQuery -->
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
<script>
    $(document).ready(function() {
        // Обработчик для всех форм
        $('form').on('submit', function(event) {
            event.preventDefault(); // Отменяем стандартную отправку формы

            const form = $(this);
            const taskId = form.attr('id'); // ID формы
            const outputBlock = form.next('.output-block'); // Блок для
вывода результата

            // Отправляем данные на сервер
            $.ajax({
                url: 'process.php', // Файл для обработки данных
                type: 'POST',
                data: form.serialize(), // Сериализуем данные формы
                success: function(response) {
                    outputBlock.html(response); // Выводим результат
                },
                error: function() {
                    outputBlock.html('Ошибка при отправке данных.');
```

## Листинг process.php

```

<?php
// process.php

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $taskId = $_POST['task_id']; // ID задания

    // Обработка данных в зависимости от задания
    switch ($taskId) {
        case 'task1':
            $name = $_POST['name'];
            $age = $_POST['age'];
```

```

$hobby = $_POST['hobby'];
echo "<ul>
    <li>Имя: $name</li>
    <li>Возраст: $age</li>
    <li>Хобби: $hobby</li>
</ul>";
break;

case 'task2':
    $integer = $_POST['integer'];
    $string = $_POST['string'];
    $array = json_decode($_POST['array']);
    $boolean = $_POST['boolean'];
    $null = $_POST['null'];
    echo "Тип \$integer: " . gettype($integer) . "<br>";
    echo "Тип \$string: " . gettype($string) . "<br>";
    echo "Тип \$array: " . gettype($array) . "<br>";
    echo "Тип \$boolean: " . gettype($boolean) . "<br>";
    echo "Тип \$null: " . gettype($null) . "<br>";
    break;

case 'task3':
    $a = $_POST['a'];
    $b = $_POST['b'];
    echo "$a + $b = " . ($a + $b) . "<br>";
    echo "$a - $b = " . ($a - $b) . "<br>";
    echo "$a * $b = " . ($a * $b) . "<br>";
    echo "$a / $b = " . ($a / $b) . "<br>";
    break;

case 'task4':
    $year = $_POST['year'];
    $isLeap = ($year % 4 == 0 && $year % 100 != 0) || ($year % 400 == 0);
    echo $isLeap ? "$year год високосный." : "$year год не високосный.";
    break;

case 'task5':
    $age = $_POST['age'];
    if ($age < 12) {
        echo "Ребенок";
    } elseif ($age >= 13 && $age <= 17) {
        echo "Подросток";
    } elseif ($age >= 18 && $age <= 64) {
        echo "Взрослый";
    } else {
        echo "Пожилый";
    }
    break;

case 'task6':
    $number = $_POST['number'];

```

```
    echo ($number % 2 == 0) ? "Четное" : "Нечетное";
    break;

case 'task7':
    $month = $_POST['month'];
    switch ($month) {
        case 1: echo "Январь"; break;
        case 2: echo "Февраль"; break;
        case 3: echo "Март"; break;
        case 4: echo "Апрель"; break;
        case 5: echo "Май"; break;
        case 6: echo "Июнь"; break;
        case 7: echo "Июль"; break;
        case 8: echo "Август"; break;
        case 9: echo "Сентябрь"; break;
        case 10: echo "Октябрь"; break;
        case 11: echo "Ноябрь"; break;
        case 12: echo "Декабрь"; break;
        default: echo "Некорректный месяц";
    }
    break;

case 'task8':
    $limit = $_POST['limit'];
    $i = 1;
    while ($i <= $limit) {
        if ($i % 3 == 0) {
            echo "$i ";
        }
        $i++;
    }
    break;

case 'task9':
    $number = $_POST['number'];
    do {
        echo "$number ";
        $number--;
    } while ($number >= 1);
    break;

case 'task10':
    $number = $_POST['number'];
    for ($i = 1; $i <= 10; $i++) {
        echo "$number * $i = " . ($number * $i) . "<br>";
    }
    break;

case 'task11':
    $limit = $_POST['limit'];
    for ($i = 1; $i <= $limit; $i++) {
```

```

        $isPrime = true;
        for ($j = 2; $j < $i; $j++) {
            if ($i % $j == 0) {
                $isPrime = false;
                break;
            }
        }
        if ($isPrime && $i > 1) {
            echo "$i ";
        }
    }
    break;

case 'task12':
    $products = $_POST['products'];
    $prices = $_POST['prices'];
    echo "<table border='1'>";
    echo "<tr><th>Товар</th><th>Цена</th><th>Цена со скидкой</th></tr>";
    foreach ($products as $index => $product) {
        $price = $prices[$index];
        $discountedPrice = $price * 0.9;
        echo
"<tr><td>$product</td><td>$price</td><td>$discountedPrice</td></tr>";
    }
    echo "</table>";
    break;

case 'task13':
    $student = [
        "имя" => $_POST['name'],
        "возраст" => $_POST['age'],
        "факультет" => $_POST['faculty'],
        "средний_балл" => $_POST['average_grade']
    ];
    function hasScholarship($student) {
        return $student["средний_балл"] >= 4.5;
    }
    echo hasScholarship($student) ? "Стипендия положена" : "Стипендия не
положена";
    break;

case 'task14':
    $day = $_POST['day'];
    $subjects = explode(",", $_POST['subjects']);
    $schedule[$day] = $subjects;
    echo "<table border='1'>";
    echo "<tr><th>День</th><th>Предметы</th></tr>";
    foreach ($schedule as $day => $subjects) {
        echo "<tr><td>$day</td><td>" . implode(" ", $subjects) .
"</td></tr>";
    }

```

```

        echo "</table>";
        break;

    case 'task15':
        $numbers = array_map('intval', explode(",", $_POST['numbers']));
        echo "Минимальное значение: " . min($numbers) . "<br>";
        echo "Максимальное значение: " . max($numbers) . "<br>";
        echo "Среднее значение: " . array_sum($numbers) / count($numbers) .
"<br>";

        sort($numbers);
        echo "Отсортированный массив: " . implode(", ", $numbers);
        break;

    case 'task16':
        $numbers = array_map('intval', explode(",", $_POST['numbers']));
        function sumNumbers(...$numbers) {
            return array_sum($numbers);
        }
        echo "Сумма чисел: " . sumNumbers(...$numbers);
        break;

    case 'task17':
        $number = $_POST['number'];
        function factorial($n) {
            if ($n <= 1) return 1;
            return $n * factorial($n - 1);
        }
        echo "Факториал $number: " . factorial($number);
        break;

    case 'task18':
        $globalVar = $_POST['global_var'];
        function testScope() {
            global $globalVar;
            static $counter = 0;
            $localVar = "Локальная переменная";
            $counter++;
            echo "Глобальная: $globalVar, Локальная: $localVar, Счетчик:
$counter<br>";
        }
        testScope();
        testScope();
        break;

    case 'task19':
        $numbers = array_map('intval', explode(",", $_POST['numbers']));
        $evenNumbers = array_filter($numbers, function($n) {
            return $n % 2 == 0;
        });
        echo "Четные числа: " . implode(", ", $evenNumbers);
        break;

```

```

case 'task20':
    $str = $_POST['string'];
    $oldWord = $_POST['old_word'];
    $newWord = $_POST['new_word'];
    function cleanString($str, $oldWord, $newWord) {
        $str = trim(strtolower($str));
        $str = str_replace($oldWord, $newWord, $str);
        return $str;
    }
    echo cleanString($str, $oldWord, $newWord);
    break;

case 'task21':
    $date1 = $_POST['date1'];
    $date2 = $_POST['date2'];
    function daysBetweenDates($date1, $date2) {
        $diff = strtotime($date2) - strtotime($date1);
        return floor($diff / (60 * 60 * 24));
    }
    echo "Дней между $date1 и $date2: " . daysBetweenDates($date1,
$date2);
    break;

case 'task22':
    $filename = $_POST['filename'];
    $content = $_POST['content'];
    file_put_contents($filename, $content);
    echo "Файл $filename записан успешно.";
    break;

case 'task23':
    $filename = $_POST['filename'];
    if (file_exists($filename)) {
        $lines = file($filename);
        foreach ($lines as $lineNumber => $line) {
            echo "Строка #" . ($lineNumber + 1) . ": " .
htmlspecialchars($line) . "<br>";
        }
    } else {
        echo "Файл не существует.";
    }
    break;

case 'task24':
    $width = $_POST['width'];
    $height = $_POST['height'];
    class Rectangle {
        private $width;
        private $height;

```



```

        public function __construct($width, $height) {
            $this->width = $width;
            $this->height = $height;
        }

        public function getArea() {
            return $this->width * $this->height;
        }

        public function getPerimeter() {
            return 2 * ($this->width + $this->height);
        }
    }
    $rectangle = new Rectangle($width, $height);
    echo "Площадь: " . $rectangle->getArea() . "<br>";
    echo "Периметр: " . $rectangle->getPerimeter();
    break;

```

case 'task25':

```

    $radius = $_POST['radius'];
    $side = $_POST['side'];
    class Shape {
        public function getArea() {
            return 0;
        }

        public function getPerimeter() {
            return 0;
        }
    }

    class Circle extends Shape {
        private $radius;

        public function __construct($radius) {
            $this->radius = $radius;
        }

        public function getArea() {
            return pi() * $this->radius * $this->radius;
        }

        public function getPerimeter() {
            return 2 * pi() * $this->radius;
        }
    }

    class Square extends Shape {
        private $side;

        public function __construct($side) {

```

```

        $this->side = $side;
    }

    public function getArea() {
        return $this->side * $this->side;
    }

    public function getPerimeter() {
        return 4 * $this->side;
    }
}

$circle = new Circle($radius);
$square = new Square($side);
echo "Площадь круга: " . $circle->getArea() . "<br>";
echo "Периметр квадрата: " . $square->getPerimeter();
break;

case 'task26':
    $a = $_POST['a'];
    $b = $_POST['b'];
    $c = $_POST['c'];
    class Triangle extends Shape {
        private $a, $b, $c;

        public function __construct($a, $b, $c) {
            $this->a = $a;
            $this->b = $b;
            $this->c = $c;
        }

        public function getArea() {
            $s = ($this->a + $this->b + $this->c) / 2;
            return sqrt($s * ($s - $this->a) * ($s - $this->b) * ($s -
$this->c));
        }

        public function getPerimeter() {
            return $this->a + $this->b + $this->c;
        }
    }

    function printShapeInfo(Shape $shape) {
        echo "Площадь: " . $shape->getArea() . ", Периметр: " . $shape-
>getPerimeter() . "<br>";
    }

    $triangle = new Triangle($a, $b, $c);
    printShapeInfo($triangle);
    break;

```

```

case 'task27':
    $filename = $_POST['filename'];
    $action = $_POST['action'];
    if (file_exists($filename)) {
        if ($action == 'view') {
            echo "<pre>" . file_get_contents($filename) . "</pre>";
        } elseif ($action == 'delete') {
            unlink($filename);
            echo "Файл $filename удален.";
        }
    } else {
        echo "Файл не существует.";
    }
    break;

case 'task28':
    $email = $_POST['email'];
    $password = $_POST['password'];
    $phone = $_POST['phone'];

    function validateForm($email, $password, $phone) {
        $errors = [];
        if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
            $errors[] = "Некорректный email.";
        }
        if (strlen($password) < 8 || !preg_match("/[A-Za-z]/", $password)
|| !preg_match("/\d/", $password)) {
            $errors[] = "Пароль должен содержать минимум 8 символов,
буквы и цифры.";
        }
        if (!preg_match("/^\+\d{10,}$/", $phone)) {
            $errors[] = "Некорректный номер телефона.";
        }
        return $errors;
    }

    $errors = validateForm($email, $password, $phone);
    if (empty($errors)) {
        echo "Форма валидна.";
    } else {
        echo "Ошибки: " . implode(" ", $errors);
    }
    break;

case 'task29':
    $data = [
        "имя" => $_POST['name'],
        "возраст" => $_POST['age'],
        "город" => $_POST['city']
    ];
    $json = json_encode($data, JSON_UNESCAPED_UNICODE);

```

```

        file_put_contents("data.json", $json);
        echo "Данные сохранены в файл data.json.";
        break;

    case 'task30':
        $filename = $_POST['filename'];
        if (file_exists($filename)) {
            $jsonFromFile = file_get_contents($filename);
            $decodedData = json_decode($jsonFromFile, true);
            echo "Декодированные данные: ";
            print_r($decodedData);
        } else {
            echo "Файл не существует.";
        }
        break;
    }
}

```

## Контрольные вопросы

### 1. Что такое PHP и каковы его основные области применения?

PHP (Hypertext Preprocessor) — это интерпретируемый язык программирования, предназначенный для веб-разработки. Основные области применения:

- Создание динамических веб-страниц
- Работа с базами данных (MySQL, PostgreSQL)
- Обработка форм и управление сессиями
- Разработка API
- Генерация отчетов и обработка файлов

### 2. Какие типы данных существуют в PHP?

PHP поддерживает несколько типов данных:

- integer (целые числа) — `$a = 10;`
- float (числа с плавающей точкой) — `$b = 3.14;`
- string (строка) — `$c = "Hello";`
- boolean (логический тип) — `$d = true;`

- array (массив) — `$e = [1, 2, 3];`
- object (объект) — `$obj = new ClassName();`
- null (отсутствие значения) — `$f = null;`

### 3. Разница между `==` и `===`

- `==` проверяет только значение (`5 == "5" → true`)
- `===` проверяет и тип данных (`5 === "5" → false`)

### 4. Циклы в PHP

- `for` — используется, когда известно количество итераций
- `while` — выполняется, пока условие истинно
- `do...while` — сначала выполняет код, затем проверяет условие
- `foreach` — перебирает массивы

### 5. Ассоциативный и многомерный массив

Ассоциативный массив:

```
$person = ["name" => "John", "age" => 30];
```

Многомерный массив:

```
$matrix = [  
    ["Apple", "Banana"],  
    ["Carrot", "Tomato"]  
];
```

### 6. Область видимости переменных

- Локальные переменные — доступны только внутри функции
- Глобальные переменные — доступны во всем коде (используется `global`)

- Статические переменные — сохраняют свое значение между вызовами функции

## 7. Анонимная функция

Функция без имени, полезна для колбэков:

```
$sayHello = function($name) {  
    return "Hello, $name!";  
};  
echo $sayHello("John");
```

## 8. ООП в PHP

- Класс — шаблон для создания объектов
- Объект — экземпляр класса
- Свойство — переменная внутри класса
- Метод — функция внутри класса

```
class Car {  
    public $color;  
    public function drive() {  
        echo "Driving";  
    }  
}  
$myCar = new Car();  
$myCar->color = "Red";
```

## 9. Наследование и полиморфизм

Наследование позволяет одному классу унаследовать свойства и методы другого:

```
class Animal {
```

```
public function makeSound() {  
    echo "Some sound";  
}  
  
class Dog extends Animal {  
    public function makeSound() {  
        echo "Bark";  
    }  
}  
  
$dog = new Dog();  
$dog->makeSound(); // Bark
```

Полиморфизм позволяет переопределять методы в дочерних классах.

## 10. Работа с файлами в PHP

Основные функции:

- `fopen()` — открыть файл
- `fwrite()` — записать в файл
- `fread()` — прочитать файл
- `fclose()` — закрыть файл
- `file_get_contents()` — получить содержимое файла
- `file_put_contents()` — записать данные в файл

Пример:

```
$file = fopen("test.txt", "w");  
fwrite($file, "Hello, PHP!");  
fclose($file);
```