

При анализе прошлых проектов затруднительно было найти какие то сложные антипаттерны. Простые есть: магические числа, зашифрованный код, программирование методом копирования и вставки, спагетти-код. Проекты были пока только учебные, не сложные и преподаватели сразу указывали на ошибки и говорили как делать нельзя не особо объясняя почему, так как смысл задания был в другом.

Антипаттерны в коде

Антипаттерн «Магическое число»

```
if ($_FILES["myfile"]["size"] > 2000000) {  
    header("Location: /gallery/?message=4");  
}
```

Нужно было вынести это число в отдельную константу с понятным названием.

Антипаттерн «Спагетти-код»

```
function spiralFilling($x, $y)  
{  
    if ($x === 0 || $y === 0) {  
        echo "Error" . "<br>";  
    } else {  
        $array = array_fill(0, $y, array_fill(0, $x, 0));  
        $count = 1;  
        $direction = 0;  
        $i = -1; //строка x  
        $j = 0; //столбец y  
        $x1 = 0; //коэф строки  
        $y1 = 0; //коэф столбца  
        $countMax = $x * $y;  
        while ($count <= $countMax) {  
            switch ($direction) {  
                case 0: //вниз  
                    if ($i < ($y - 1 - $y1)) {  
                        $i++;  
                    } else {  
                        $direction++;  
                        $j++;  
                    }  
                    break;  
                case 1: //вправо  
                    if ($j < ($x - 1 - $x1)) {  
                        $j++;  
                    } else {  
                        $direction++;  
                        $i--;  
                    }  
                    break;  
                case 2: //вверх  
                    if ($i > $y1) {  
                        $i--;  
                    } else {  
                        $direction++;  
                        $j--;  
                        $y1++;  
                    }  
                    break;  
                case 3: //влево  
                    if ($j > (1 + $x1)) {  
                        $j--;  
                    } else {  
                        $direction = 0;  
                    }  
            }  
            $count++;  
        }  
    }  
}
```

```

        $i++;
        $x1++;
    }
    break;
}
$array[$i][$j] = str_pad((string)$count, strlen($countMax), "0",
STR_PAD_LEFT);
$count++;
}
}
return $array;
}

```

Наверное реализация этой функции подходит под этот паттерн. Задача решалась «в лоб». Через некоторое время думаю смогу решить эту задачу более изящно.

Методологические антипаттерны

Антипаттерн «Затычка на ввод»

```

function getNewsContent($id)
{
    $sql = "SELECT * FROM news WHERE id = {$id}";
    $news = getAssocResult($sql);

    $result = [];
    if(isset($news[0]))
        $result = $news[0];
    return $result;
}

```

Переменные которые приходят от пользователя следует проверять на корректность типа данных и нельзя напрямую такую переменную вставлять в sql-запрос.

Добавить в самом начале функции строку с приведением переменной \$id к типу integer: \$id = (int) \$id;

Ошибка появилась по невнимательности. Эту ошибку можно отнести к паттерну программирование методом копирования и вставки. Так как это копи-паст другой аналогичной функции с небольшой доработкой. При копировании была потеряна строка.

Архитектурные антипаттерны

Антипаттерн «Волшебная кнопка»

```

<h2>Каталог</h2>

<? foreach ($catalog as $item): ?>
    <h2><a href="/product/card/?id=<?= $item['id'] ?>"><?= $item['name']
?></a></h2>
    <p><?= $item['description'] ?></p>
    <p>Цена: <?= $item['price'] ?></p>
    <button data-id="<?= $item['id'] ?>" class="buy">Купить</button>
    <hr>
<? endforeach; ?>

<a href="/product/catalog/?page=<?= $page ?>">Далее</a>

<script>
    let buttons = document.querySelectorAll('.buy');
    buttons.forEach((elem) => {
        elem.addEventListener('click', () => {
            let id = elem.getAttribute('data-id');

```

```

        (
            async () => {
                const response = await fetch('/basket/buy/', {
                    method: 'POST',
                    headers: new Headers({
                        'Content-Type': 'application/json'
                    }),
                    body: JSON.stringify({
                        id: id
                    })
                });
                const answer = await response.json();
                document.getElementById('count').innerText =
answer.count;
            }
        ) ();
    });
</script>

```

Смешивание разного рода кода.

Код на JavaScript нужно вынести в отдельный подключаемый файл.