

 09

Faça o que eu fiz na aula

Extrair função

Bem-vindo ao curso de Refatoração! para prosseguirmos com o curso, precisamos ter baixado um arquivo zip com o projeto, dentro desse projeto temos várias pastas que vamos utilizar ao longo do curso.

Primeiramente vamos iniciar o servidor embutido do PHP para acessarmos os arquivos do projeto, para fazermos isso, vamos abrir a linha de comando do sistema operacional na pasta do projeto, e depois, iniciar o servidor com o comando:

```
php -S 0.0.0.0:3000
```

Agora com o servidor iniciado, vamos acessar a pasta `extrair_funcao` no navegador, acessando o seguinte endereço:

```
https://localhost:3000/extrair_funcao
```

Abrindo o arquivo `index.php`, nós temos um array associativo com correntistas e seus respectivos saldos de contas bancárias, junto com uma lógica que decide qual mensagem será impressa na tela. Estamos vendo se o resultado da função `array_key_exists` é verdadeiro, se sim, indica que a chave do array foi encontrada, e vamos imprimir o texto na tela, se for falso, exibimos a mensagem de erro na tela.

Se executarmos o código no navegador, ele funciona, mas como vimos no vídeo, este código tem muitas responsabilidades, ajudaria muito a legibilidade e a manutenção se pudéssemos isolá-lo em partes separadas.

Então usamos a refatoração extrair função, e podemos adicionar o seguinte código:

```
function exibeSaldo(string $nome, array $saldos): void
{
    echo "<p>O saldo do $nome é: {$saldos[$nome]}</p>";
}

function exibeErro(): void
{
    echo '<p>Correntista não existente.</p>';
}
```

E vamos alterar a condicional para chamar estas funções:

```
if (array_key_exists($nome, $saldos)) {
    exibeSaldo($nome, $saldos);
} else {
    exibeErro();
}
```

E agora podemos extrair esta condicional para uma função também:

```
function exibeSaldoCorrentista(string $nome, array $saldos): void
{
    if (array_key_exists($nome, $saldos)) {
        exibeSaldo($nome, $saldos);
    } else {
        exibeErro();
    }
}
```

Agora que as funções estão prontas, podemos chamá-las:

```
exibeSaldoCorrentista('Giovanni', $saldos);
```

E recarregamos a página no navegador, temos o mesmo resultado, o que significa que a nossa refatoração deu certo.

Incorporar Função

Na última aula extraímos os códigos em funções, abra a pasta `incorporar_funcao` e no arquivo `index.php`, vemos que o código está bem separado em funções, porém, vamos olhar a função `getSaldo`

```
function getSaldo(string $nome, array $saldos): int
{
    return $saldos[$nome];
}
```

Esta função não faz mais nada do que pegar a posição de um array, extrair isso para um array é adicionar complexidade desnecessária para a aplicação. É importante sempre nos perguntarmos “esta refatoração deixou o código mais legível?” sempre que aplicarmos uma refatoração. Neste caso, como aplicarmos demais a refatoração Extrair Função, podemos aplicar uma refatoração que faz o contrário, chamada Incorporar Função para deixarmos o código mais simples e legível.

Vamos pegar a posição do array que está dentro da função e colocá-la diretamente dentro da condição do `if`:

```
if ($saldos[$nome] > 0) {
```

Depois que escrevemos esta condição, a função `getSaldo` não tem mais referências, então podemos remover a definição dela apagando as linhas:

```
function getSaldo(string $nome, array $saldos): int
{
    return $saldos[$nome];
}
```

Com isso o nosso código ficou mais simples e legível, é muito importante sempre avaliarmos o código enquanto estamos refatorando para ver se ele realmente ficou mais simples.

Extrair Variável

Na pasta `extrair_variavel` temos um arquivo `index.php` e dentro deste arquivo temos uma função que verifica se a variável recebida pelo método HTTP GET é um e-mail válido, se sim, imprime uma mensagem com o e-mail e se não for válido,

imprime uma mensagem de erro.

O código funciona, mas como vimos no vídeo, ele pode ser um pouco difícil de ler e interpretar, ele está todo resumido em uma linha, e isso torna ele mais difícil de ser lido e entendido.

Podemos aplicar aqui uma refatoração chamada Extrair Variável, vamos pegar a função `filter_var`, as mensagens de sucesso e erro, e atribuir em variáveis:

```
$email = filter_var($_GET['email'], FILTER_SANITIZE_EMAIL);
$mensagem = 'O seu e-mail é: ' . $_GET['email'];
$mensagemErro = 'O seu e-mail não é válido';
```

Usar a forma reduzida do `if` também dificulta a leitura do código, podemos trocar ele pela forma extendida, que é mais clara e objetiva.

```
if ($email !== false) {
    echo $mensagem;
} else {
    echo $mensagemErro;
}
```

Com isso, refatoramos o código para deixá-lo mais extenso, porém, mais descriptivo, mais fácil de ler e dar manutenção.