

Method Replace Inline Code with Function Call

🎓 College: [Faculdade Metodista Granbery](#).

👤 Teacher: [Marco Antônio - Github](#) | [Linkedin](#)

📖 Book: [Refatoração - Aperfeiçoando o design de códigos existentes - Martin Fowler](#)

FOWLER, Martin. “Replace Inline Code With Function Call” no código. *In*: REFATORAÇÃO: Aperfeiçoando o design de códigos existentes. 2. ed. [S. l.: s. n.], 2019. cap. 8.

Método

O método substituir código internalizado por chamada de função (**Replace Inline Code with Function Call**), demonstra as funções como melhorias de códigos. Com o decorrer do desenvolvimento do software, acabamos criando códigos repetidos que trazem o mesmo resultado. Talvez nem prejudique o software, mas com as manutenções no decorrer do tempo do software pode ocorrer pode trazer dificuldades para alterar tantos códigos repetidos do mesmo resultado. Esse método melhora o código tendo que criar uma função, essa função vai se transformar em apenas um código para que não necessite ficar repetindo várias vezes. Como Martin Fowler fala no livro, iremos empacotar o código para dentro dessa função. Nos locais que possui o Código repetido vamos apenas chamar a função.

Código

O código que estou usando para exemplo, é um código do meu repositório. Que tem objetivo de verificar o valor de entrada e retorna se é impar ou par. [CODE](#)

Procedimento

- Substitua o código internalizado por uma chamada à função existente.
- Teste

Ferramenta para teste

PyTest

```
pip install ipytest
```

Objetivo da atividade

1. Elabore ou escolha um código que a refatoração possa ser aplicada.
2. Identifique o mau cheiro associado.
3. Escreva os casos de teste e execute
4. Aplique a refatoração
5. Execute os casos de teste novamente
6. Documente cada etapa para a apresentação

Mau Cheiro

Código Duplicado

O mau cheiro que encontrei dentro do código abaixo foi o **código duplicado**, nós como desenvolvedores acabamos criando repetições de códigos sem a necessidade. Podendo **prejudicar o código para manutenções**, exigindo tempo para procurar a expressão que está repetida.

Código antigo

Você pode visualizar como existe a repetição.

```
while True:
    try:
        valor = int(input('Digite um valor:'))
        if valor % 2 == 0:
            print('Número par')
        else:
            print('Número ímpar')

        valor2 = int(input('Digite um valor:'))
        if valor % 2 == 0:
            print('Número par')
        else:
            print('Número ímpar')

        y = input('Deseja continuar? (S|N): ').upper()
        if y == 'N':
            break

    except:
        print('Digite apenas números')
```

Código novo aplicando a refatoração

Agora observe a criação da função sendo utilizada para chamar a lógica apenas uma vez.

```
def verificaParouImpar(x):  
    if x % 2 == 0:  
        return 'Número par'  
    else:  
        return 'Número ímpar'  
  
if __name__ == "__main__":  
    while True:  
        try:  
            valor = int(input('Digite um valor:'))  
            print(verificaParouImpar(valor))  
            y = input('Deseja continuar? (S|N): ').upper()  
            if y == 'N':  
                break  
        except:  
            print('Digite apenas números')
```

FUNÇÃO

Realização dos Testes

```
Run: Python tests in test_parOuImpar.py  
✓ Tests passed: 2 of 2 tests - 0 ms  
Testing started at 09:44 ...  
===== test session starts =====  
collecting ... collected 2 items  
test_parOuImpar.py::test_answer_impár PASSED [ 50%]  
test_parOuImpar.py::test_answer_par PASSED [100%]  
===== 2 passed in 0.02s =====  
Process finished with exit code 0
```