

O que aprendemos?

O que aprendemos?

- Refatoração NÃO É correção de bugs
 - Um código só deve ser refatorado DEPOIS de rodar corretamente os requisitos da aplicação
- Refatoração NÃO É criação de novas funcionalidades
- Código Limpo
 - evita duplicação de código
 - possibilita maior clareza
 - facilita comunicação entre a equipe de desenvolvimento
 - para identificar e eliminar as classes e métodos supérfluos
 - facilita manutenção do código
 - precisa garantir que todos os testes continuem passando
- Débito técnico
 - É o código ruim que vai se acumulando ao decorrer do desenvolvimento
 - É muito comum postergar melhorias. O famoso "deixar para depois".
 - Refatorar reduz/elimina o débito técnico
 - Refatorar garante código com maior qualidade
 - Código com qualidade reduz o tempo de correção de bugs

Técnica de Refatoração Nº 1

- Extrair método (Extract Method)
 - é usado quando você tem um trecho de código que pode ser agrupado
 - Quando Extrair método?
 - Quando houver código duplicado
 - Quando o método for muito grande
 - Quando houver comentários
 - Quando houver cadeias de mensagens
 - Quando NÃO extrair método?
 - Quando o corpo do método for ficar tão óbvio quanto seu nome

Técnica de Refatoração Nº 2

- Incorporar método (Inline Method)
 - Quando incorporar método?
 - Quando o corpo do método é tão óbvio quanto seu nome, e o método se torna desnecessário
 - Quando NÃO refatorar?
- Quando NÃO incorporar método?
 - Quando o corpo do método não é auto-explicativo:

```
int GetNextLevel() { return level + 1; }
```

é óbvio, mas

```
int GetValue() { return (valorBase + acrescimo - desconto) * promocao;}
```

tem uma expressão complexa demais para o nome do método, e não deveria ser incorporado