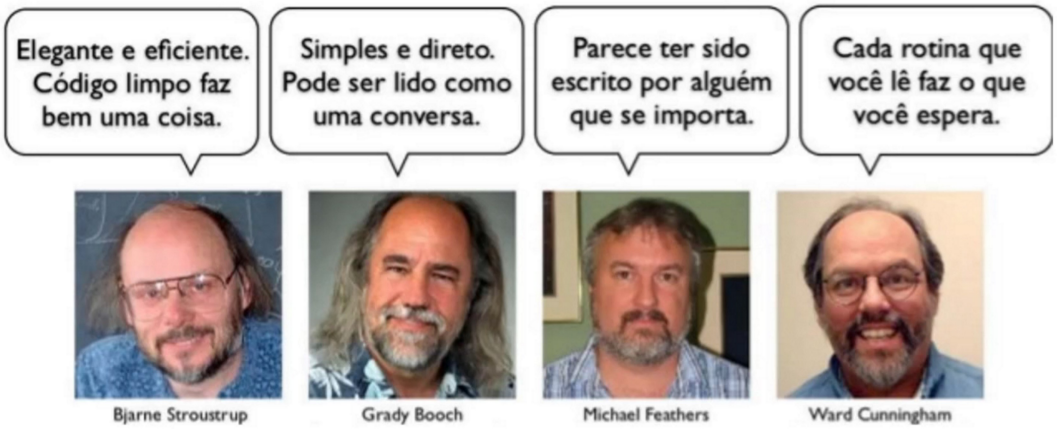
Clean Code

domingo, 23 de janeiro de 2022 19:04



- ✓ Simples
- ✓ Direto
- ✓ Eficiente
- ✓ Sem Duplicidade
- ✓ Elegante
- ✓ Feito com cuidado
- ✓ Fácil de ler

Qualquer tolo consegue escrever código que um computador entenda. Bons programadores escrevem código que humanos possam entender - Martin Fowler

Desculpas e Responsabilidades:

- ➢ Mas o cronograma está apertado
- ➢ Meu chefe me pressiona a entregar logo
- ➢ Quero mostrar produtividade
- ➢ Não ganho o suficiente para escrever o melhor código do mundo
- ➢ A empresa não valoriza bom código e sim a entrega

E De quem é a culpa?

- A sua carreira é sua responsabilidade
- Leia, Estude, Pratique
- Vá a conferências
- Faça Cursos

Não é da responsabilidade do seu empregador lhe oferecer cursos, oportunidades ou até mesmo um projeto com a tecnologia que pretende se especializar

Você recebe para trabalhar 40 horas por semana e resolver os problemas da sua empresa, não os seus. - Uncle Bob

Quanto custa o código ruim:

- ➢ Alta rotatividade
- ➢ Demora nas entregas de novas funcionalidades
- ➢ Dificuldade na manutenção
- ➢ Alta incidência de bugs
- ➢ Perda de confiança de cliente
- ➢ Desmotivação Profissional
- ➢ Mais tempo depurando código do que escrevendo

Como medir um bom código:

- ✓ Linhas de código
- ✓ Número de métodos
- ✓ Número de classes
- ✓ Linhas de código por método
- ✓ Complexidade ciclomática
- ✓ Número de estruturas de decisão
- ✓ Nomes Significativos:
  - Nomes que revelam a intenção
  - Por que existe

    - O que faz
    - Como é usado
    - Nomes fáceis de encontrar
    - Use nomes pronunciáveis
    - Não economize palavras
    - Revele a intenção do código
- ✓ Evite palavras que podem ser variáveis ou palavras reservadas em outras plataformas ex: user
- ✓ Evite dar nomes como "doubleValorPromocional", o tipo não precisa estar no nome
- ✓ Evite trocadilhos, não misture idiomas, não mescle nomes

Boas Práticas:

- ✓ Nomes Classes devem ser substantivos e não deve conter verbos
    - Ex: ClienteRepository
- ✓ Nomes do métodos devem contar verbos de preferência no infinitivo
    - Ex: AdicionarCliente
- ✓ Não seja genérico
- ✓ Menos é Mais:
    - A primeira regra dos métodos é que devem ser pequenos. A segunda regra é que eles devem ser menores ainda - Uncle Bob
    - Método <= 20 linhas
    - Linha <= 100 caracteres
    - Classe <= 500 linhas
- ✓ Métodos:
    - Extraia trechos em métodos privados
    - Métodos devem fazer apenas uma coisa, fazê-la certa e somente fazê-la
    - Evite muitos parâmetros
    - Não deixa o método mentir dizendo que faz uma coisa e faz outras "escondidas"
    - Se o método tiver mais de uma responsabilidade extraia em dois ou mais
    - Leia seu método de cima para baixo como uma narrativa, ele deve fazer sentido
    - Aplique uma boa indentação
- ✓ Comentários
    - Não ajudam um código ruim a ser melhor interpretado
    - Um código que requer comentário precisa ser reescrito
    - Não deixa trechos de código comentado
    - Quando comentar:
        - Alertar consequências que pode vir a causar
        - Licença direitos autorais, etc.
        - Necessidade de explicar uma regra de negocio interna
        - Decisões de design de código
        - Gerar documentação através de comentários
- ✓ Tratamento de Erros
    - Tratar e prever possíveis exceções é de responsabilidade do desenvolvedor
    - Retorne exceptions e não codigos de erro
    - Informe o máximo que puder na sua exception
    - Se necessário crie exceptions personalizadas ara um problema especifico
    - Não retorne null (Catch mudo)
    - Regra do escoteiros:
        - Deixe a area de acampamento mais limpa do que você encontrou

```
// processa folha de pagamento
Processa();
// calcula imposto de renda
Calcula();
```

OU?

```
ProcessarFolhaPagamento();

CalcularImpostoRenda();
```