

Código Limpo

Robert C. Martin

Capítulo 8:
~~Limites~~ Fronteiras
(Boundaries)
por James Grenning



Daniel Pereira Volpato

02/08/2022

A low-angle, upward-looking photograph of two modern skyscrapers with glass facades. The buildings are angled towards the right, creating a sense of height and scale. The sky is a pale, overcast grey. The word 'Introdução' is superimposed in the center in a large, bold, black font.

Introdução

O uso de código de terceiros (extra ou intra-empresa) é inevitável.

Como integrar esse código de forma limpa?

A low-angle, upward-looking photograph of two modern skyscrapers with glass facades. The buildings are slightly tilted, creating a sense of height and architectural grandeur. The sky is a pale, overcast grey. The text is centered over the image.

Importância do código de terceiros

Muita coisa nos passa despercebida no ecossistema de desenvolvimento de software:

- Bootloaders
- Máquinas virtuais de linguagens interpretadas
- Bibliotecas de estruturas de dados
- Bibliotecas matemáticas
- O trabalho dos transpiladores, compiladores e link-editores
- Garbage collector
- etc.

Uma experiência pessoal

3778

E se você não pudesse usar **NENHUMA biblioteca externa*** (de terceiros ou mesmo lib padrão da linguagem)?

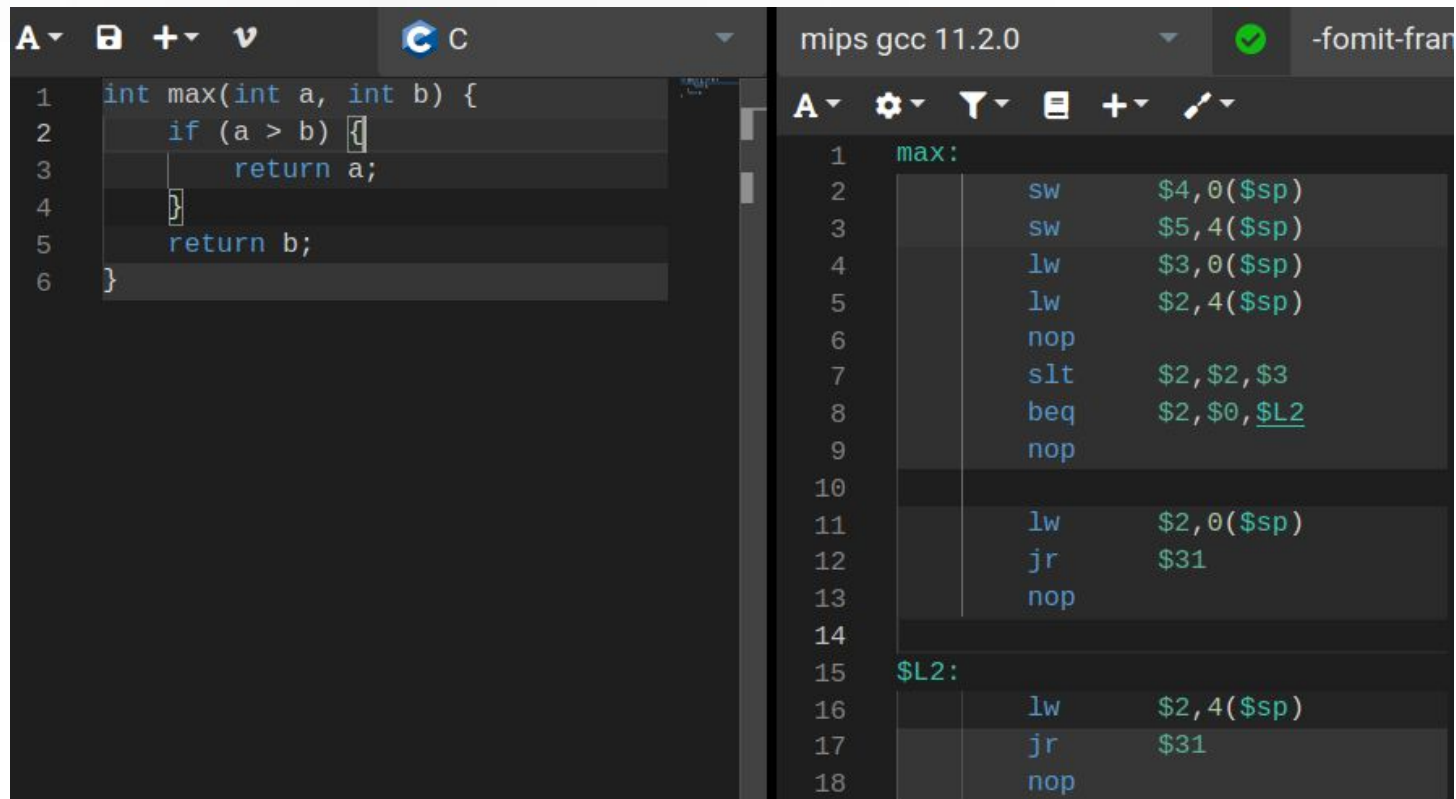


* a menos que você certifique esse código

Uma experiência pessoal

3778

E se você não pudesse nem mesmo confiar no trabalho do compilador?



The image shows a side-by-side comparison of C source code and its compiled MIPS assembly. The left pane shows the C code for a `max` function, and the right pane shows the corresponding MIPS assembly generated by `mips gcc 11.2.0`.

```
1 int max(int a, int b) {  
2     if (a > b) {  
3         return a;  
4     }  
5     return b;  
6 }
```

```
mips gcc 11.2.0 -fomit-fran  
1 max:  
2     sw     $4, 0($sp)  
3     sw     $5, 4($sp)  
4     lw     $3, 0($sp)  
5     lw     $2, 4($sp)  
6     nop  
7     slt    $2, $2, $3  
8     beq    $2, $0, $L2  
9     nop  
10  
11     lw     $2, 0($sp)  
12     jr     $31  
13     nop  
14  
15 $L2:  
16     lw     $2, 4($sp)  
17     jr     $31  
18     nop
```


Uma experiência pessoal

3778

Bem-vindos ao mundo da **aviônica** e da DO-178C (Software Considerations in Airborne Systems and Equipment Certification)



Por que usar código de terceiros?

3778

Vantagens

- Economia de tempo de desenvolvimento
- Ter (potencialmente) a contribuição de desenvolvedores mais desenvolvedores

Desvantagens

- Maior aplicabilidade da lib = métodos mais genéricos
 - Usuário sempre quer algo voltado para suas necessidades
- Código pode ser fechado

A low-angle, upward-looking photograph of two modern skyscrapers with glass facades. The buildings are set against a pale, overcast sky. The perspective creates a sense of height and architectural scale. The text 'Na prática' is centered over the image.

Na prática

To wrap or not to wrap?

3778



```
Map<Sensor> sensors = new HashMap<Sensor>();  
// ...  
Sensor s = sensores.get(sensorId);
```

- ✓ Muito prática e direta
- ✗ Os diversos métodos de Map estão acessíveis
 - Usuário pode fazer operações indesejadas ou que quebrem a consistência do negócio
- ✗ Mais difícil de manter
 - Map pode mudar ou podemos ter que trocar Map por outra estrutura

To wrap or not to wrap?



```
public class Sensors {  
    private Map<Sensor> sensors = new HashMap<Sensor>();  
    // ...  
    public Sensor getById(String id) {  
        return sensors.get(id);  
    }  
}
```

- ✓ Limita a API para os métodos que fazem sentido para o negócio
- ✓ Melhor manutenção
- ✗ Mais código para se testar
- ✗ Use com moderação (é inviável criar wrapper para tudo!)

A low-angle, upward-looking photograph of two modern skyscrapers with glass facades. The buildings are slightly tilted, creating a sense of height and architectural scale. The sky is a pale, overcast grey. The text is overlaid in the center of the image.

Como integrar com código de terceiro?

Como integrar?

3778

- Ler documentação (getting started, exemplos, etc.
 - Nem sempre é fácil localizar exatamente o que se deseja saber

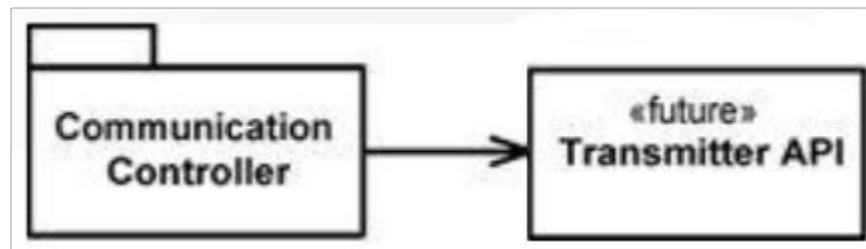


“Nesses testes [de aprendizagem] **chamamos a API de terceiro como faríamos ao usá-la em nossa aplicação**. Basicamente estamos realizando **experimentos controlados** que verificam nosso conhecimento daquela API. O foco do teste é no que desejamos obter daquela API.” (tradução livre)

- ✓ Verifica se o componente se comporta como esperamos
- ✓ Facilita atualização de versão da biblioteca

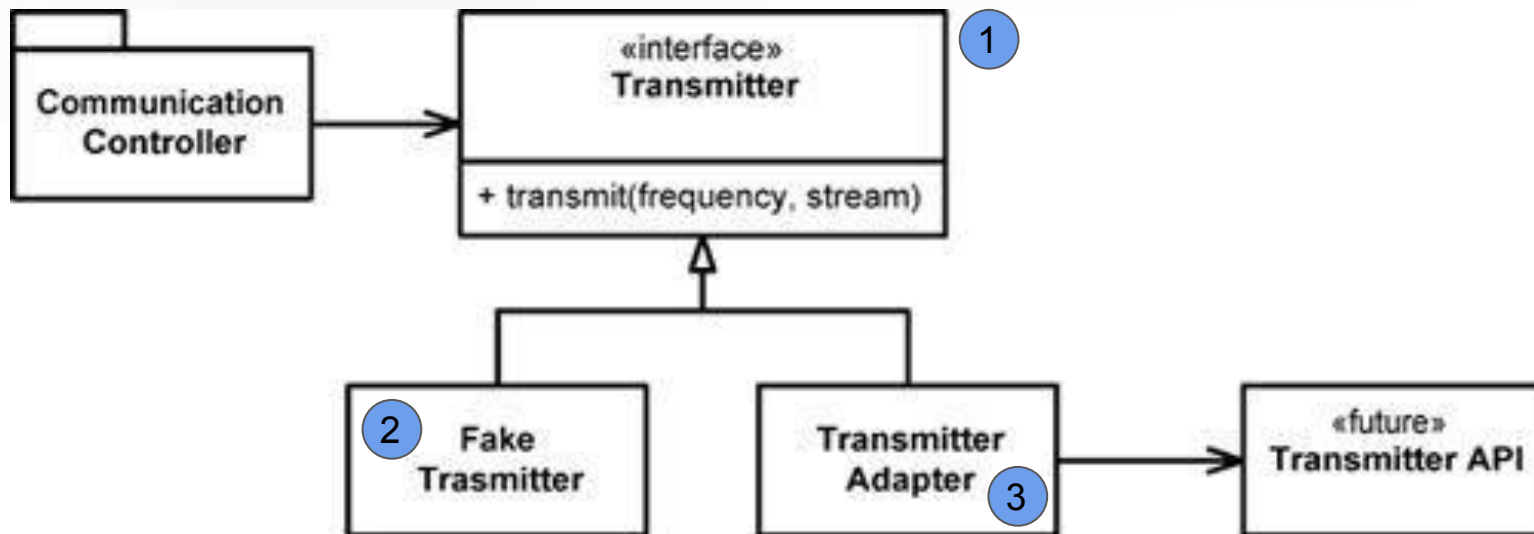


Como trabalhar com código que ainda não existe?



BACK TO THE FUTURE





1. Criamos uma interface com a qual desejamos trabalhar.
2. Criamos uma implementação *fake* enquanto a verdadeira está sendo produzida
 - a. Também nos permite testar usuários dessa interface.
3. Quando ela estiver pronta, criamos um Adapter (Wrapper).

A low-angle, upward-looking photograph of two modern skyscrapers with glass facades. The buildings are angled towards the right, creating a sense of height and scale. The sky is a uniform, overcast grey. The word "Resumo" is centered in the middle of the image in a large, bold, black sans-serif font.

Resumo

“Bons projetos de software acomodam modificações sem muito investimento ou trabalho”

- Evitar que a maior parte do nosso código enxergue detalhes do código de terceiros.
- Lidar com códigos de terceiros em poucos lugares
 - Empacotá-los (wrapper)
 - Usar um adaptador para converter nossa interface naquela que foi fornecida

Dúvidas e discussões

grupo ● 3778