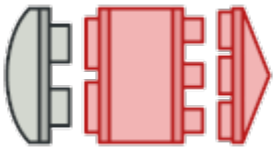
refactoring.guru

Adapter em Go / Padrões de Projeto

3–4 minutes



O **Adapter** é um padrão de projeto estrutural, que permite a colaboração de objetos incompatíveis.

O Adapter atua como um wrapper entre dois objetos. Ele captura chamadas para um objeto e as deixa reconhecíveis tanto em formato como interface para este segundo objeto.

Exemplo conceitual

Temos um código cliente que espera alguns recursos de um objeto (Lightning port), mas temos outro objeto chamado *adaptee* (laptop Windows) que oferece a mesma funcionalidade, mas por meio de uma interface diferente (porta USB)

É aqui que o padrão Adapter entra em cena. Criamos um tipo de struct conhecido como *adapter* que irá:

- Seguir a mesma interface que o cliente espera (Lightning port).
- Traduzir a solicitação do cliente para o adapter na forma que o adapter espera. O adapter aceita um conector Lightning e, em

seguida, converte seus sinais em um formato USB e os passa para a porta USB no laptop com Windows.

client.go: Código cliente

```
package main

import "fmt"

type Client struct {
}

func (c *Client) InsertLightningConnectorIntoComputer(com
Computer) {
    fmt.Println("Client inserts Lightning connector into computer.")
    com.InsertIntoLightningPort()
}
```

computer.go: Interface cliente

```
package main

type Computer interface {
    InsertIntoLightningPort()
}
```

mac.go: Serviço

```
package main
```

```
import "fmt"

type Mac struct {
}

func (m *Mac) InsertIntoLightningPort() {
    fmt.Println("Lightning connector is plugged into mac machine.")
}
```

windows.go: Serviço desconhecido

```
package main

import "fmt"

type Windows struct{}

func (w *Windows) insertIntoUSBPort() {
    fmt.Println("USB connector is plugged into windows machine.")
}
```

windowsAdapter.go: Adapter

```
package main

import "fmt"

type WindowsAdapter struct {
    windowMachine *Windows
}
```

```
func (w *WindowsAdapter) InsertIntoLightningPort() {
    fmt.Println("Adapter converts Lightning signal to USB.")
    w.windowMachine.insertIntoUSBPort()
}
```

main.go

```
package main

func main() {

    client := &Client{}
    mac := &Mac{}

    client.InsertLightningConnectorIntoComputer(mac)

    windowsMachine := &Windows{}
    windowsMachineAdapter := &WindowsAdapter{
        windowMachine: windowsMachine,
    }


client.InsertLightningConnectorIntoComputer(windowsMachineAdapter)
}
```

output.txt: Resultados da execução

```
Client inserts Lightning connector into computer.
Lightning connector is plugged into mac machine.
```

Client inserts Lightning connector into computer.

Adapter converts Lightning signal to USB.

USB connector is plugged into windows machine.

Adapter em outras linguagens