

Padrões de Projeto

Prof. Adilson Vahldick

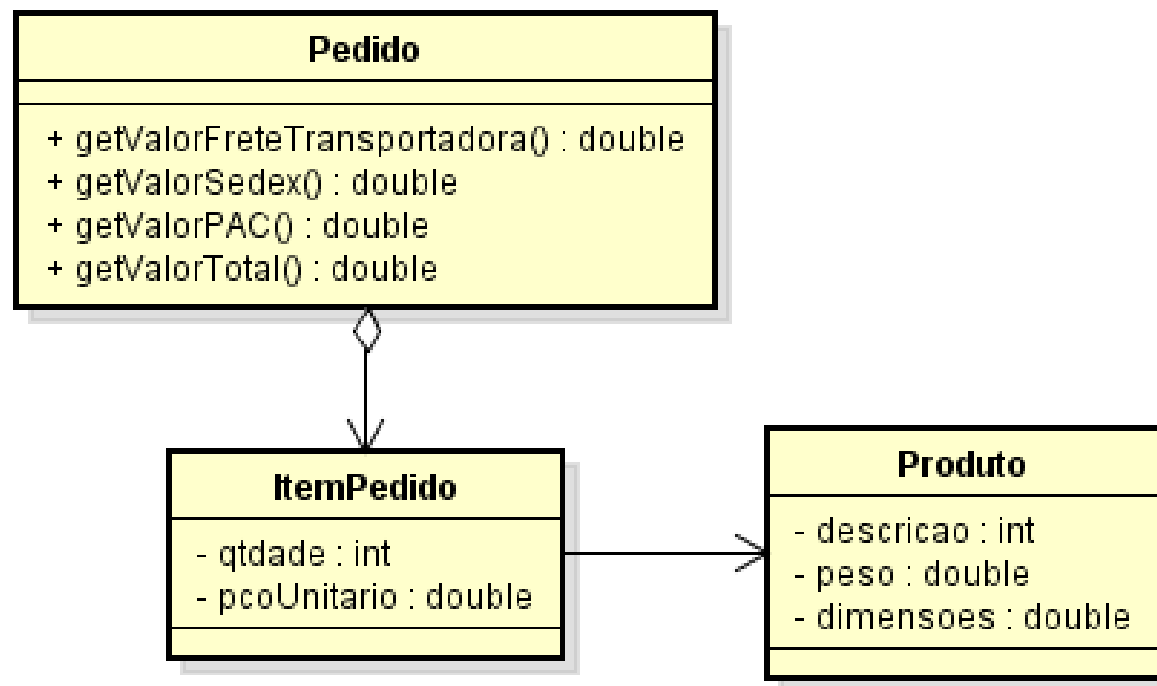
Departamento de Engenharia de Software

Udesc Ibirama

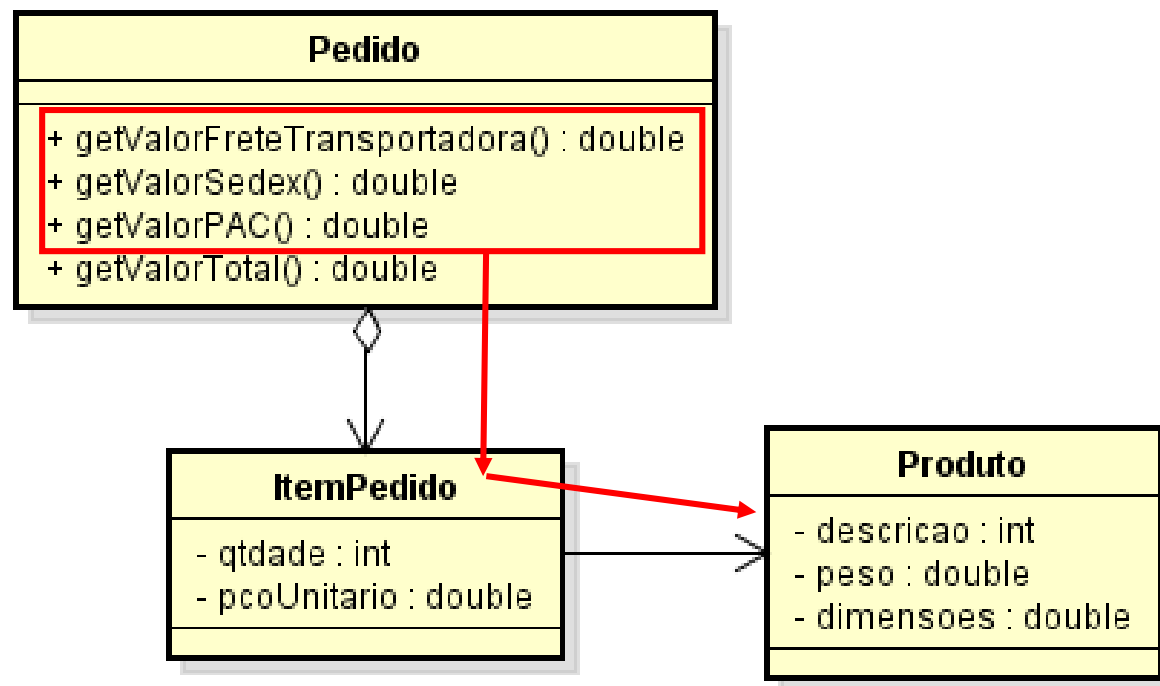
Objetivos da aula

- Conhecer e aplicar o padrão
 - Visitor

Problema (1)



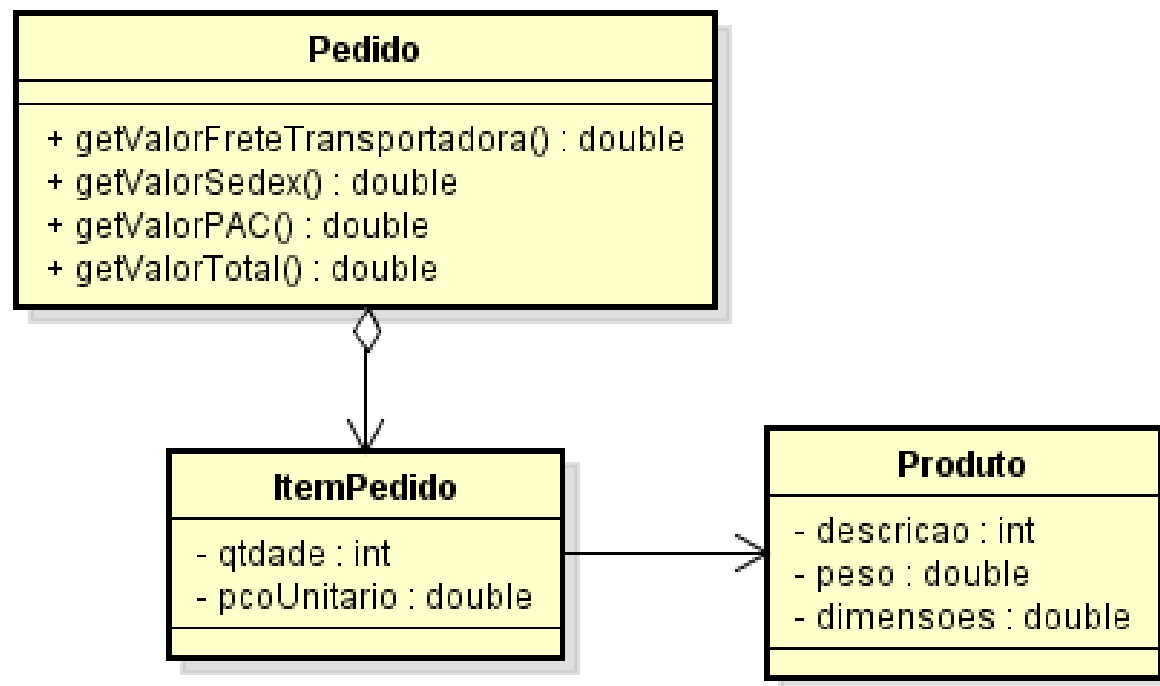
Problema (2)



A responsabilidade do cálculo do transporte de um produto, deveria ser no próprio Produto, pois ele é o **especialista da informação**.

Falha em termos de coesão, pois o Produto não deveria ter a responsabilidade do cálculo do transporte.

Problema (3)



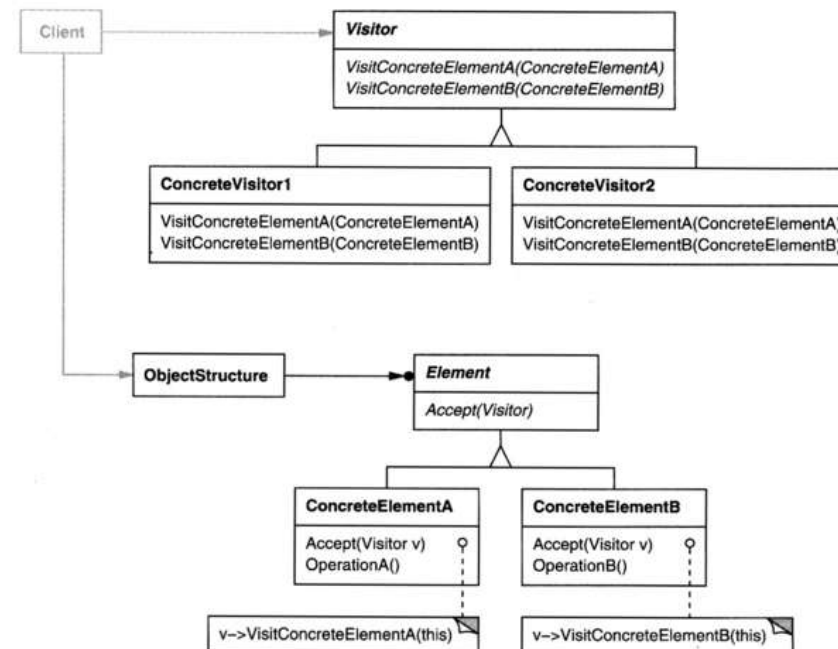
- Podem ser adicionados novos tipos de envio do produto
- Podem existir novas regras para definir o preço do produto, como os impostos e o parcelamentos

Consequência:

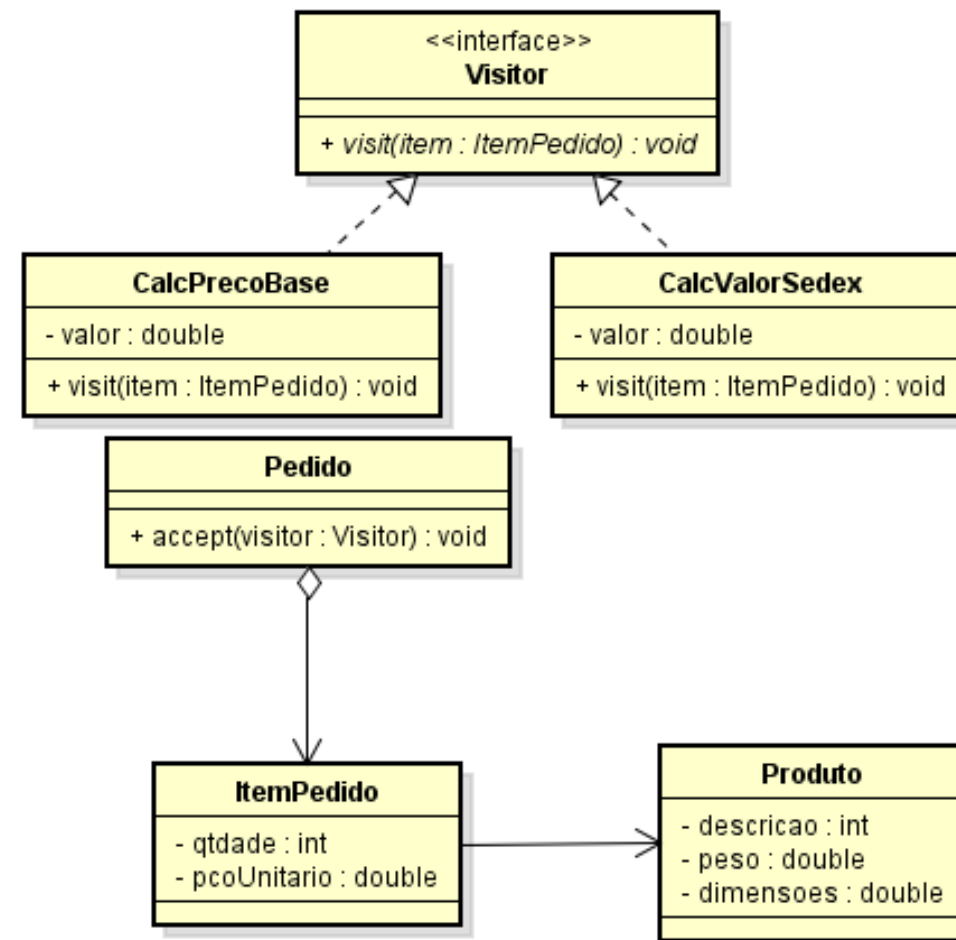
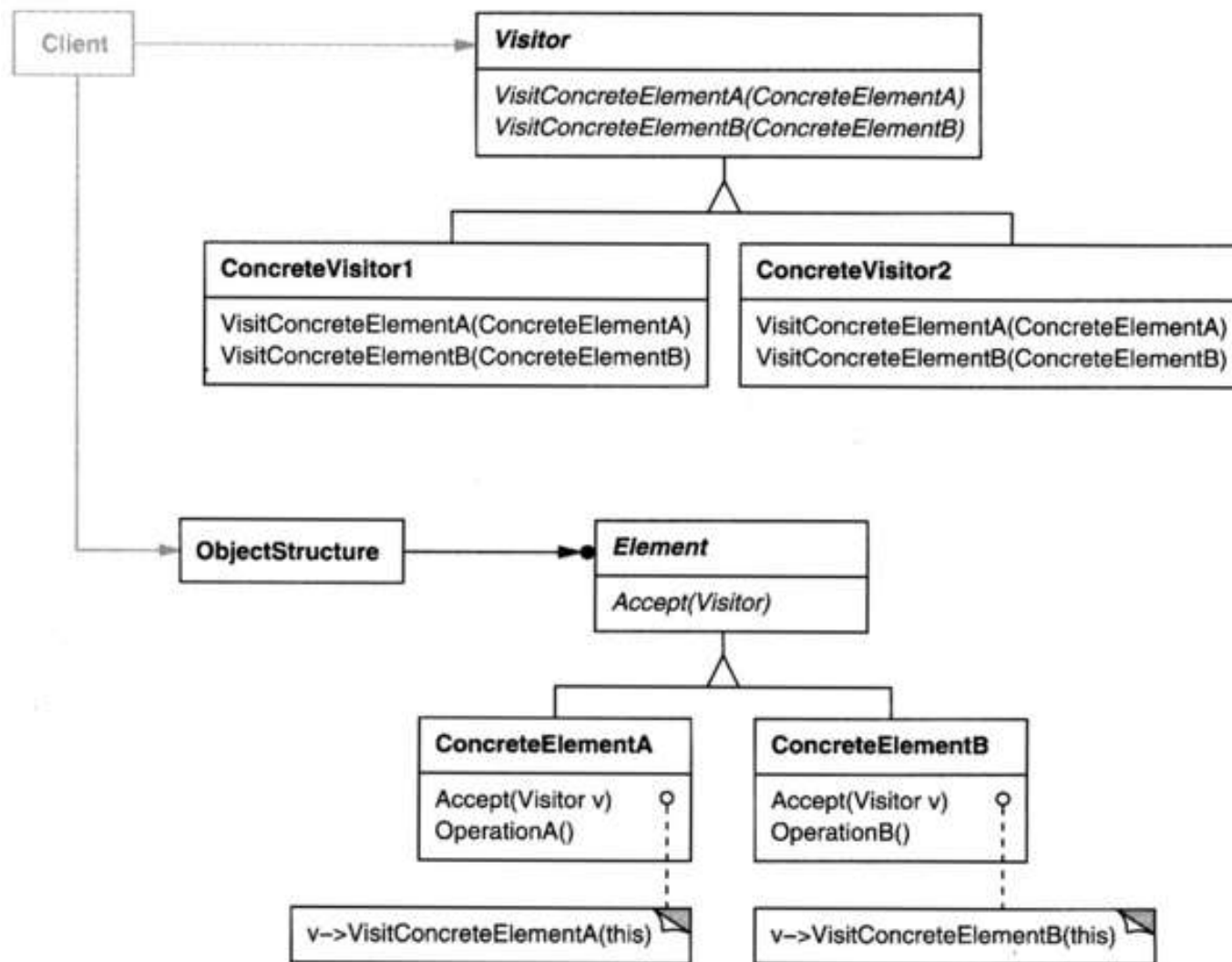
- Inchamento das classes com novas responsabilidades
- Propagação dessas responsabilidades pelas classes de itens

Solução (1)

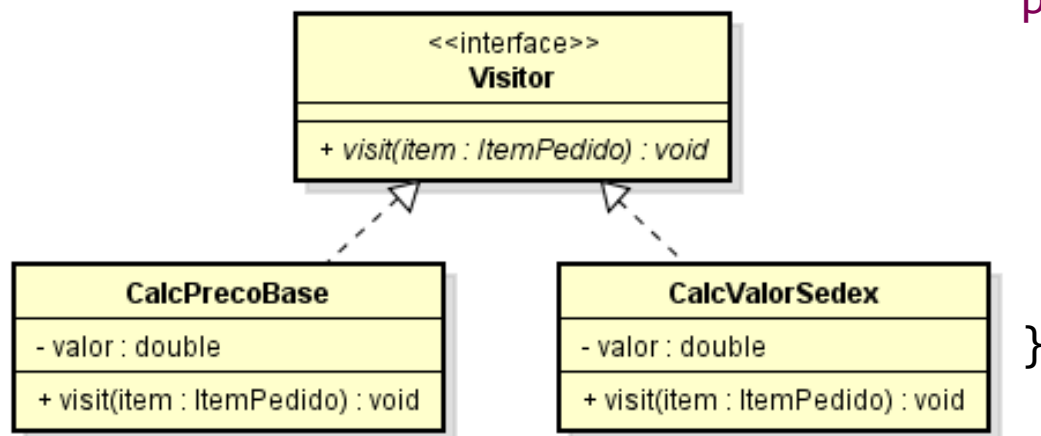
- **Visitor:** Representar uma operação a ser realizada nos elementos de uma estrutura de objetos. Visitor permite que você defina uma nova operação sem alterar as classes dos elementos nos quais a operação atua.



Solução (2)



Solução (3)



```

public class CalcPrecoBase implements Visitor {

    public void visit(ItemPedido itemPedido) {
        valor += itemPedido.getPcoUnitario()
                *itemPedido.getQtdade();
    }
}
  
```

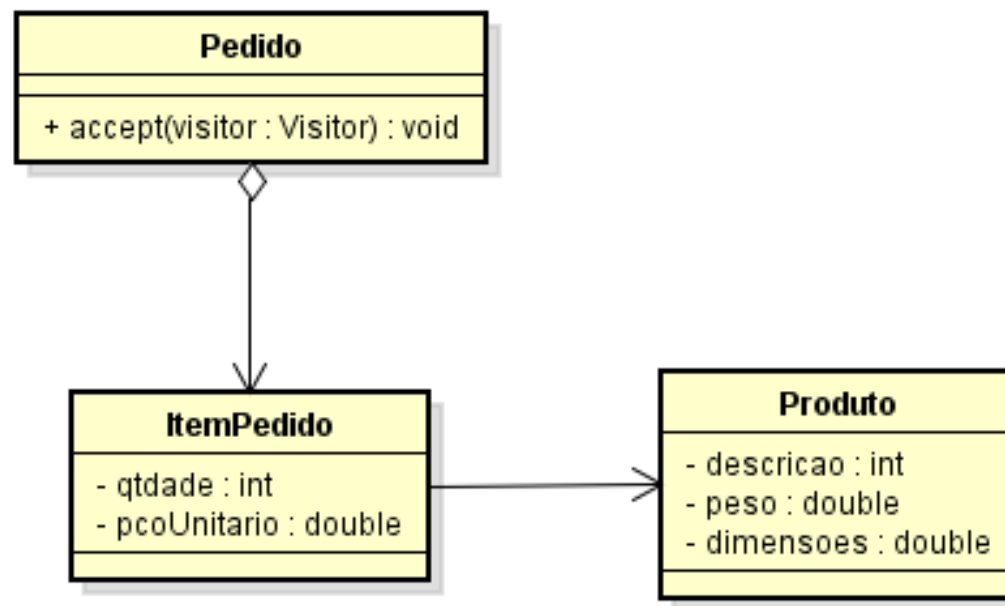
```

public class CalcValorSedex implements Visitor {

    public void visit(ItemPedido itemPedido) {
        valor+= itemPedido.getProduto().getPeso() * 0.5 * itemPedido.getQtdade();
    }

}
  
```

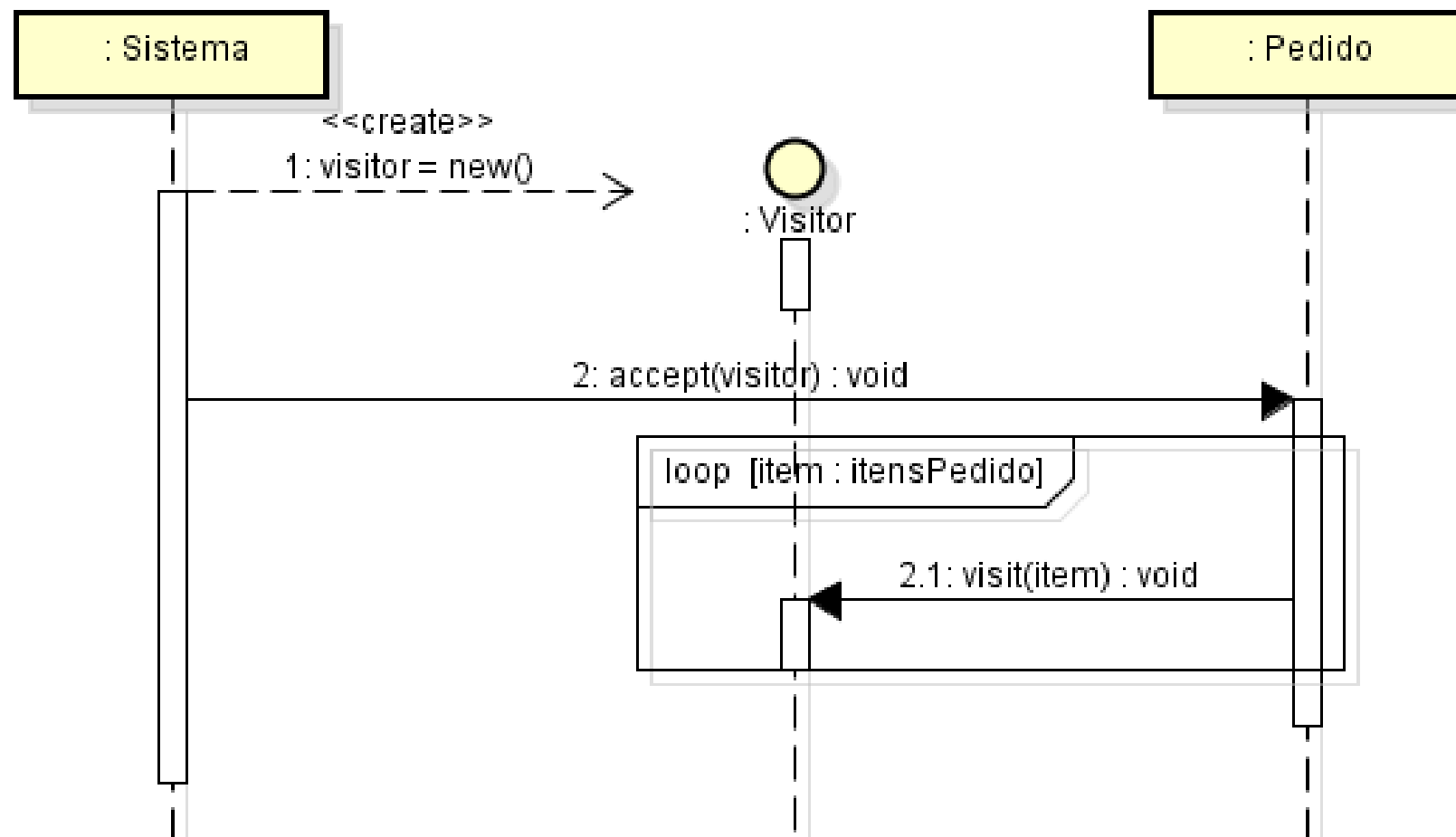

Solução (4)



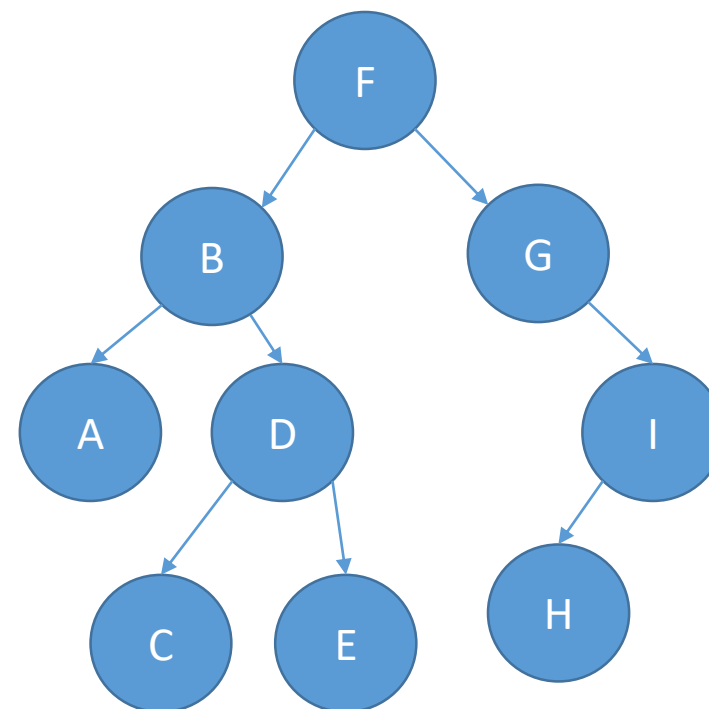
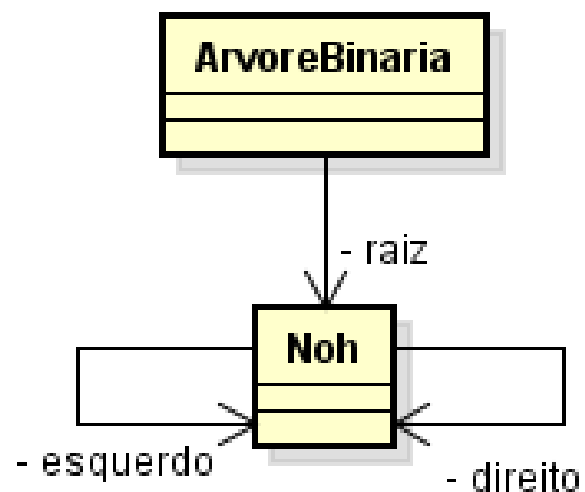
```
public class Pedido {
    ...
    public void accept(CalcVisitor visitor) {

        for (ItemPedido item:itensPedido)
            res += visitor.visit(item);
    }
}
```

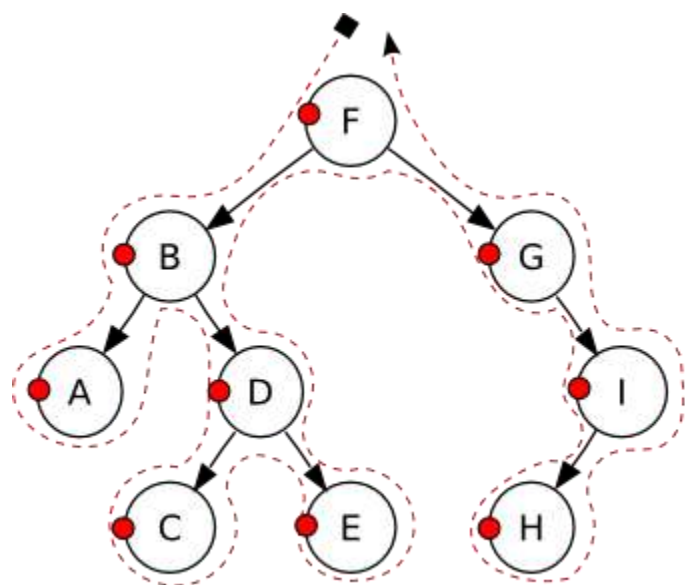
Solução (5)



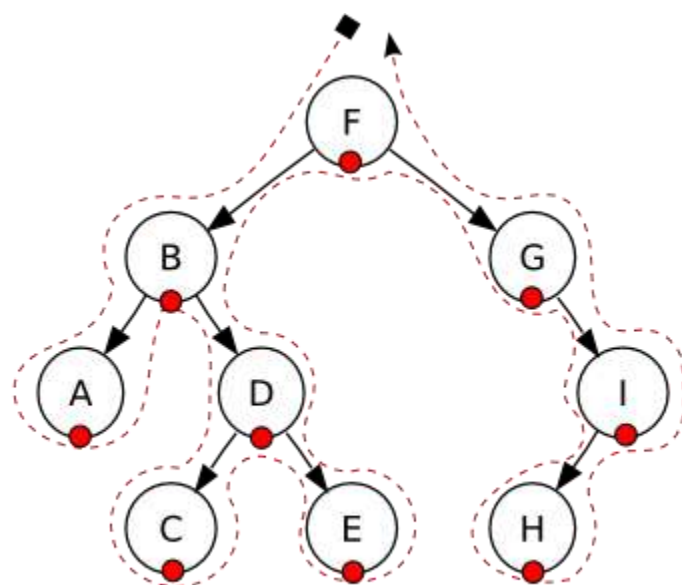
Exercício (1) – visitor2



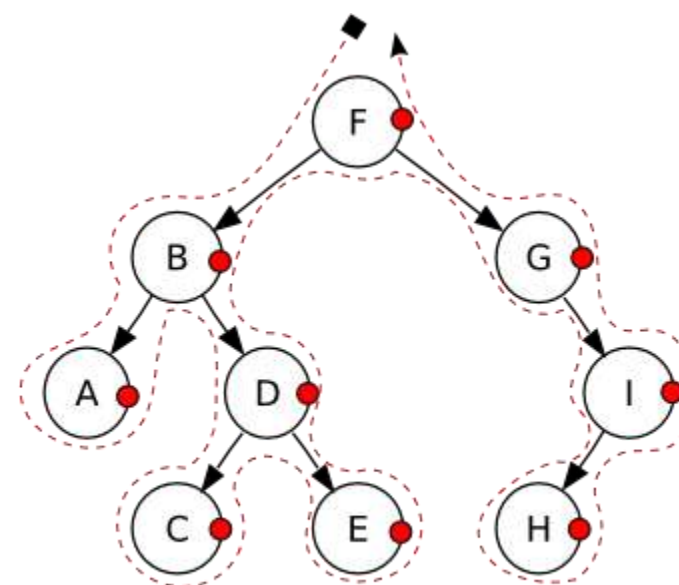
Exercício (1) – visitor2



Pré-ordem: F, B, A, D, C, E, G, I, H



Ordem simétrica: A, B, C, D, E, F, G, H, I



Pós-ordem: A, C, E, D, B, H, I, G, F



Exercício (2) – visitor3

- O projeto contém a estrutura de um setor com seus funcionários e gerentes
- Interface Visitor está pronta
- Faça visitantes concretos para:
 - Imprimir o total gasto com a folha de pagamento
 - Fazer gozarem as férias (descontar das férias a gozar): Gerente e (FuncNivelA \geq 5000), gozam 10 dias, e (FuncNivelB \geq 1500) gozam 5 dias. Imprimir o total de funcionários que gozaram férias

Exercício (3) – visitor4

- O projeto possui três tipos de personagens: Arqueiro, Lanceiro e Catapulta, cada um com diferentes pontos de vida e força de ataque.
- Desenvolva um visitor para representar o ataque de cada tipo de personagem. A tabela abaixo representa a relação de absorção do dano entre as unidades, cada linha representa o atacante, e as colunas os defensores:

	Arqueiro	Lanceiro	Catapulta
Arqueiro	0	+1 dano	0
Lanceiro	-1 dano	0	0
Catapulta	+ 1 dano	+2 danos	0

Exercício (3+) – visitor4

- Aplicando o padrão decorator, cria um item Armadura que diminui em 1 dano o ataque recebido (a catapulta não pode receber a armadura)
- Aplicando o padrão decorator, cria um item Fogo que aumenta em 1 dano o ataque feito