

Padrões de Projeto

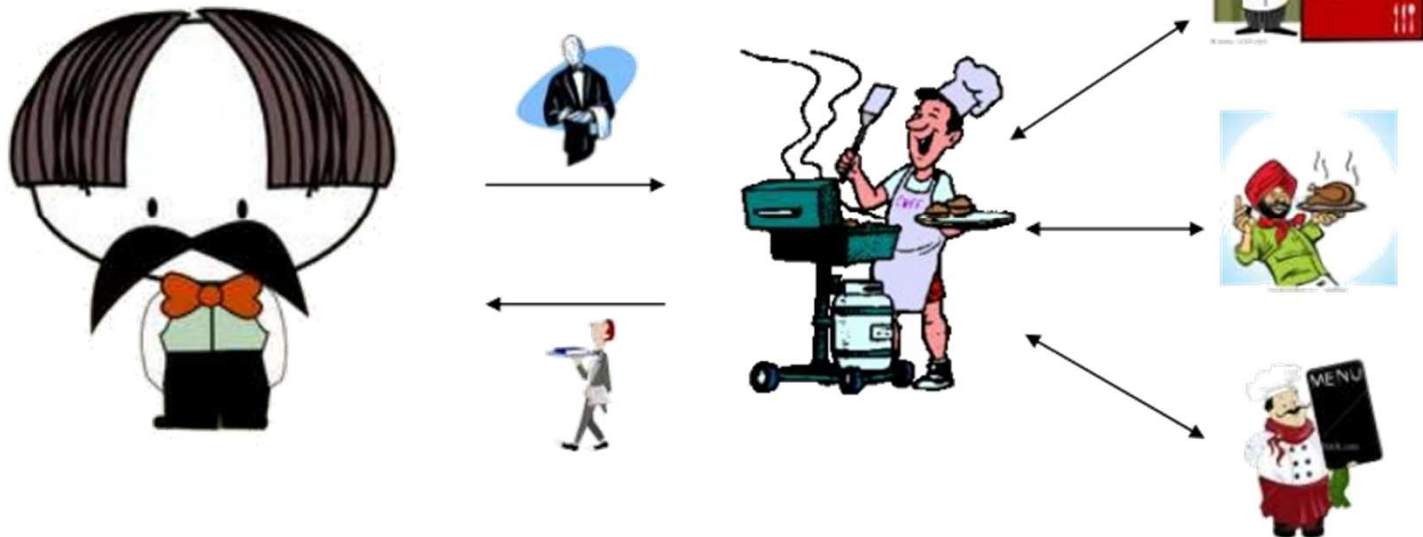
Prof. Adilson Vahldick

Departamento de Engenharia de Software

Udesc Ibirama

Objetivos da aula

- Conhecer e aplicar o padrão
 - Abstract Factory



Problema (1)

- Nosso sistema precisa de um recurso para fornecer classes responsáveis por acesso a dados. Mas temos clientes que usam XML e outros usam arquivos de acesso aleatório.

```
public class AcessoDados {  
  
    public ClienteDados getClienteDados(int tipo) {  
  
        switch (tipo) {  
            case 1: return new ClienteXML();  
            case 2: return new ClienteRandomAccess();  
        }  
        return null;  
    }  
}
```

Problema (2)

```
public class AcessoDados {  
  
    public ClienteDados getClienteDados(int tipo) {  
  
        switch (tipo) {  
            case 1: return new ClienteXML();  
            case 2: return new ClienteRandomAccess();  
        }  
        return null;  
    }  
}
```

Se eu precisar de mais um tipo de tabela a ser lida ?

Se eu precisar de outro formato de dados ?

Problema (3)

Se eu precisar de mais um tipo de tabela a ser lida ?

```
public class AcessoDados {  
  
    public ClienteDados getClienteDados(int tipo) { . . . }  
  
    public PedidoDados getPedidoDados(int tipo) {  
  
        switch (tipo) {  
            case 1: return new PedidoXML();  
            case 2: return new PedidoRandomAccess();  
        }  
        return null;  
    }  
}
```

Problema (4)

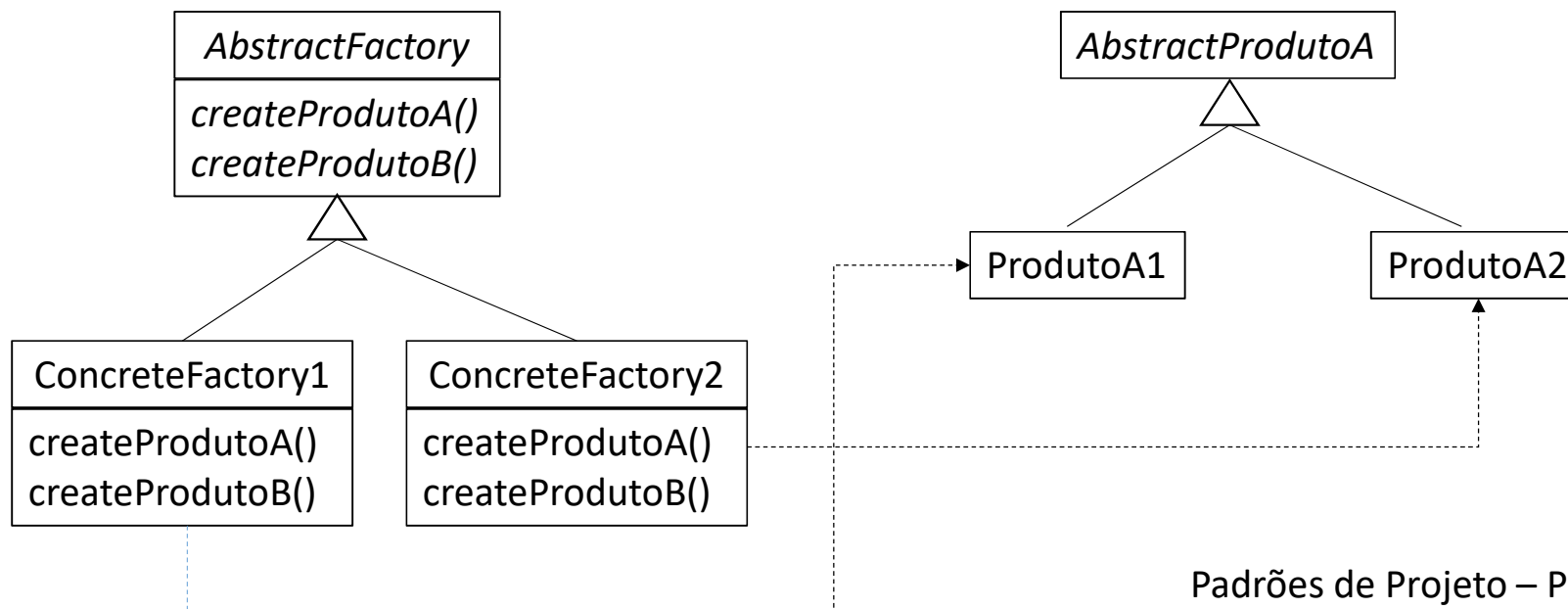
Se eu precisar de outro formato de dados ?

```
public class AcessoDados {  
  
    public ClienteDados getClienteDados(int tipo) {  
  
        switch (tipo) {  
            case 1: return new ClienteXML();  
            case 2: return new ClienteRandomAccess();  
            case 3: return new ClienteRemoto();  
        }  
        return null;  
    }  
}
```

Precisamos replicar em cada tipo de tabela !!

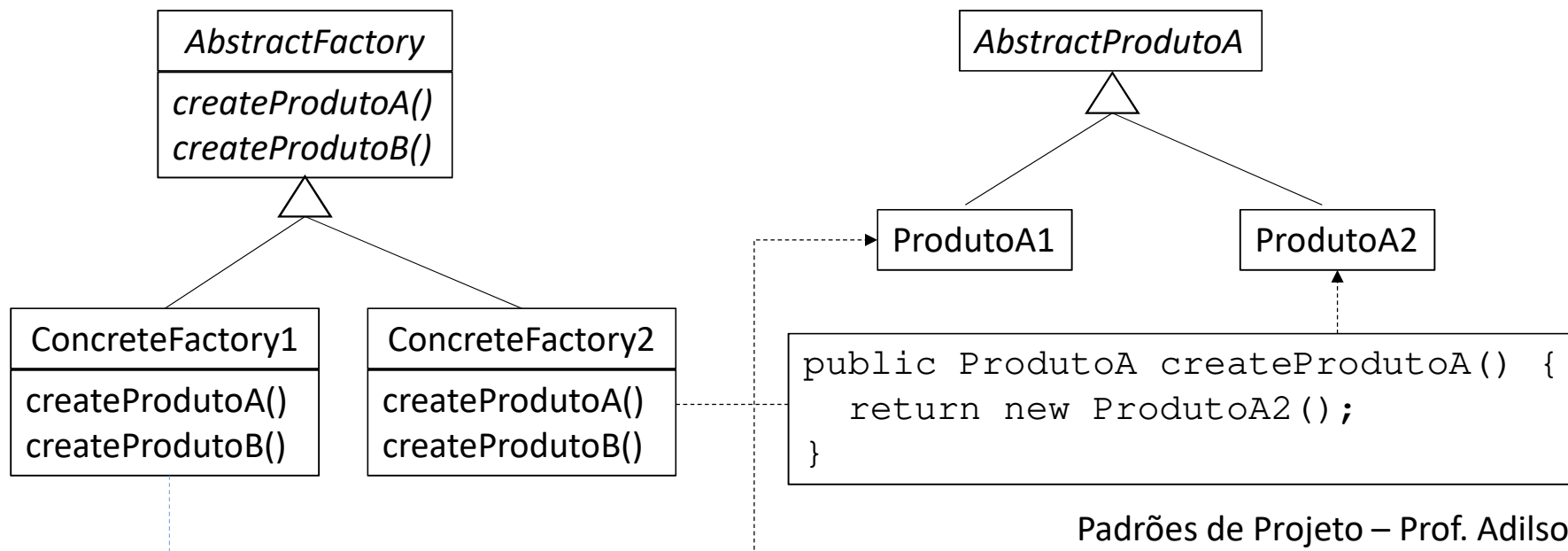
Problema (5)

- **Abstract Factory:** Fornecer uma interface para criação de famílias de objetos relacionados ou dependentes sem especificar suas classes concretas



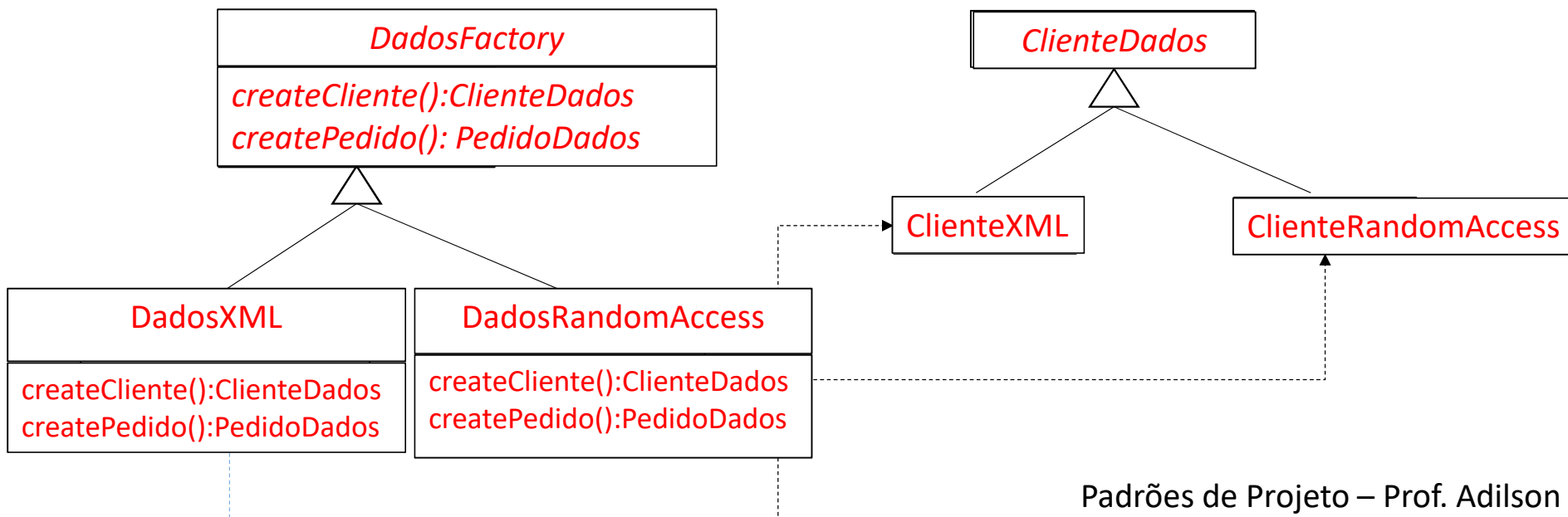
Problema (5)

- **Abstract Factory:** Fornecer uma interface para criação de famílias de objetos relacionados ou dependentes sem especificar suas classes concretas



Problema (6) – abstractfactory1

- **Abstract Factory:** Fornecer uma interface para criação de famílias de objetos relacionados ou dependentes sem especificar suas classes concretas



Exercício 1 – abstractfactory2

- Nessa aplicação existem famílias de modelos de carros
- Já está pronto da Ford. Inclusive, experimente a classe Cliente
- Para ajudar, faça um diagrama de classes do que já tem pronto

Tarefa:

- Crie uma nova família de carros da Fiat: Palio e PalioSedan. O bagageiro do PalioSedan suporta 300kg mas tem um rack no teto que suporta mais 100kg. `getKgBagageiro()` deve retornar somente o peso no bagageiro, desconsiderando o rack.
- Não esqueça de adicionar o teste dessa nova fábrica na classe Cliente

Exercício 2 – abstractfactory3

- A parte abstrata está feita (AbstractFactory e AbstractProduct).
- Tem uma tela no modelo MVC que está pronta para usar para testes. Abra a classe CalculadoraFretImpl e altere duas linhas de código que estão em comentário.
- Crie duas Fábricas Concretas e por consequência seus produtos.
- Veja nos slides a seguir as especificações das fábricas
- Todas as inconsistências devem disparar uma exceção

Exercício 2 – abstractfactory3

- Correio
 - Normal: se baseia somente na cubagem (por isso ela precisa ser maior que 0)
 - Verificar se foi especificado um destino

Destino	Preço do cm ³
Norte	0,03
Centro Oeste	0,04
Nordeste	0,02
Sudeste	0,01
Sul	0,01

Destino	Preço do cm ³
América Latina	0,07
América do Norte	0,10
Europa	0,15
África	0,25
Ásia	0,17
Oceania	0,30

Exercício 2 – abstractfactory3

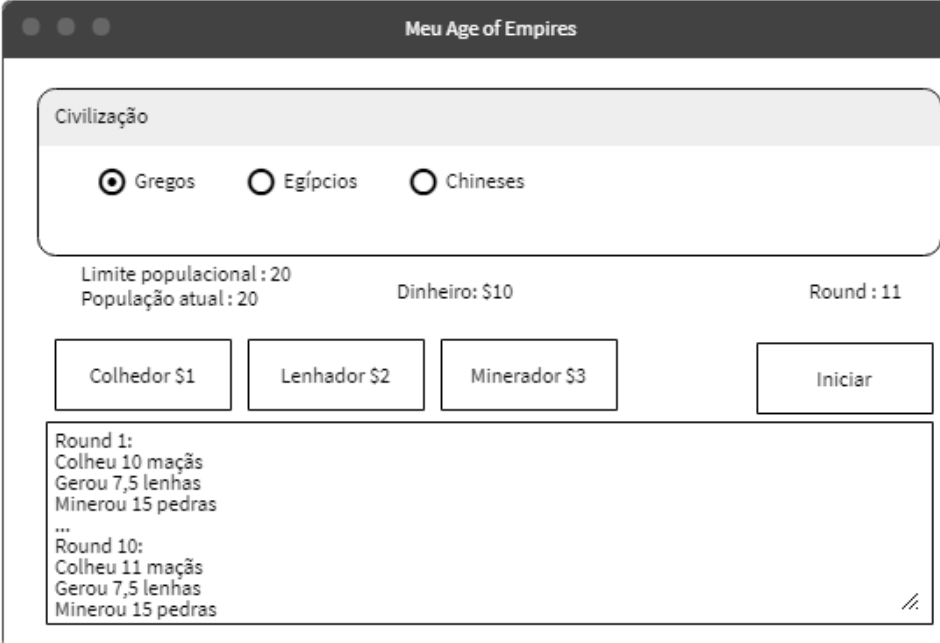
- Correio
 - Expresso: se baseia somente no peso (por isso ele precisa ser maior que 0)
 - Verificar se foi especificado um destino válido (somente as regiões do Brasil)
 - Permitir especificar somente destinos válidos
 - Norte ou Nordeste: peso * 0,05
 - Centro Oeste: peso * 0,075
 - Sul ou Sudeste: peso * 0,025

Exercício 2 – abstractfactory3

- Exportar para América do Norte
 - Normal:
 - Já cria o objeto com destino para América do Norte
 - Alterar o destino não tem efeito
 - Consistir se o peso e a cubagem são maiores que ZERO
 - Cálculo do preço é $\text{cubagem} * 0.009 + \text{peso} * \text{fator de modificação} * 0.005$
 - Fator de modificação:
 - 0.3: Para cubagem até (inclusive) 27000
 - 0.7: Cubagem entre 27001 e até (inclusive) 21600
 - 1.0 para os outros casos
 - Expresso: Modo não suportado de frete

Exercício 3 – abstractfactory4

- Abra o enunciado



The screenshot shows a Java Swing window titled "Meu Age of Empires". Inside the window, there is a section labeled "Civilização" with three radio buttons: "Gregos" (selected), "Egípcios", and "Chineses". Below this, there are three status labels: "Limite populacional : 20", "População atual : 20", "Dinheiro: \$10", and "Round : 11". There are four buttons: "Colhedor \$1", "Lenhador \$2", "Minerador \$3", and "Iniciar". At the bottom, there is a text area showing the game log for Round 1 and Round 10. Round 1 log: "Round 1: Colheu 10 maçãs, Gerou 7,5 lenhas, Minerou 15 pedras". Round 10 log: "Round 10: Colheu 11 maçãs, Gerou 7,5 lenhas, Minerou 15 pedras".

Meu Age of Empires

Civilização

☒ Gregos ☐ Egípcios ☐ Chineses

Limite populacional : 20
População atual : 20

Dinheiro: \$10

Round : 11

Colhedor \$1 Lenhador \$2 Minerador \$3 Iniciar

Round 1:
Colheu 10 maçãs
Gerou 7,5 lenhas
Minerou 15 pedras
...
Round 10:
Colheu 11 maçãs
Gerou 7,5 lenhas
Minerou 15 pedras