

Universidade Federal de Uberlândia
Faculdade de Computação
Disciplina: POO2
Prof. Fabiano Azevedo Dorça

Padrões Fábrica
Abstract Factory

Abstract Factory (Fábrica Abstrata)

- Fornece uma interface para criar famílias de objetos relacionados sem especificar as suas classes concretas.
- Existe uma classe conhecida pelo cliente que é a “Fábrica Abstrata”.
- Esta fábrica abstrata pode possuir diversas implementações concretas, cada uma capaz de fabricar instâncias de diferentes famílias de produtos concretos.

Abstract Factory (Fábrica Abstrata)

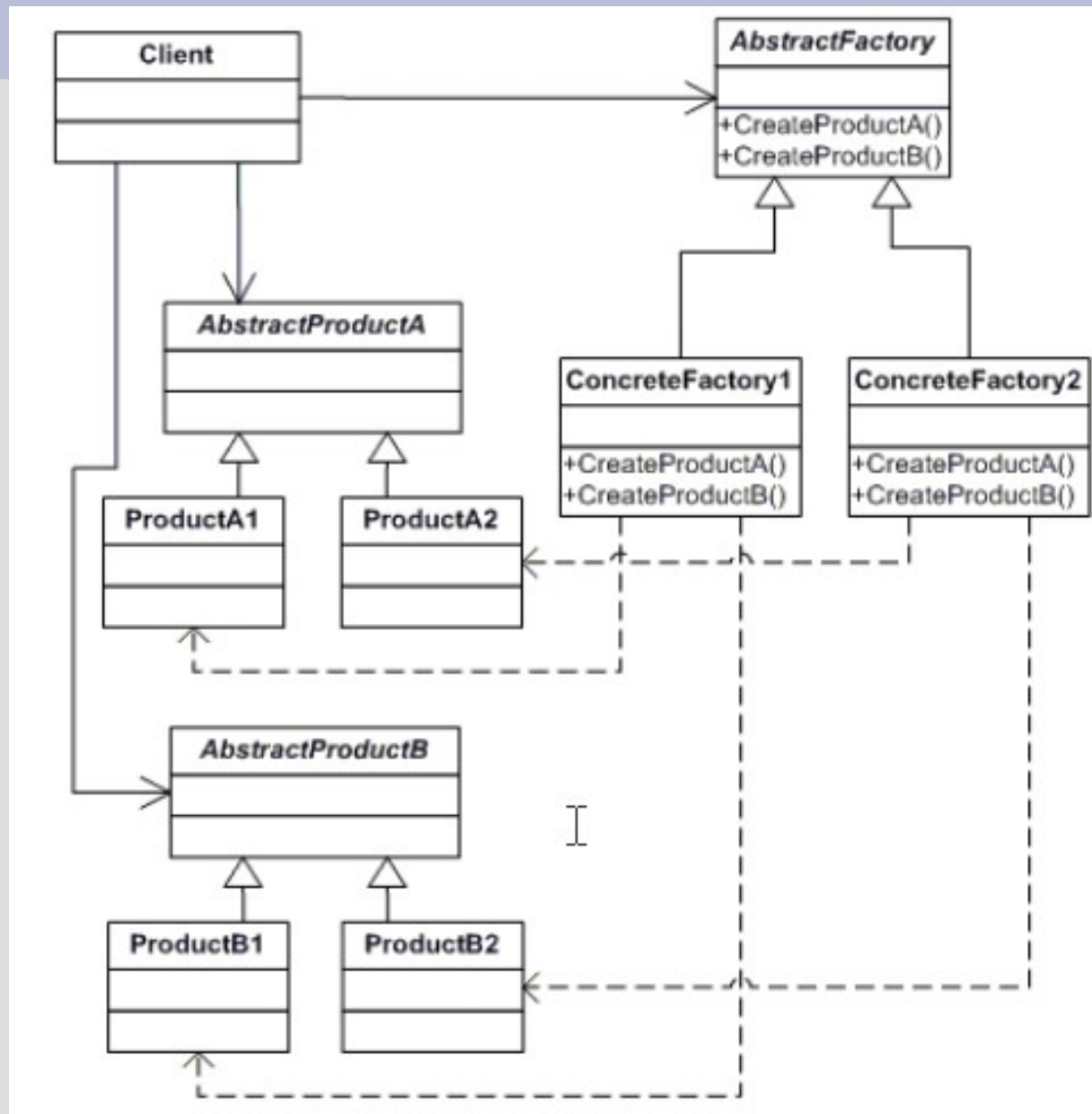
- Desta forma, o cliente estará apto a trabalhar com diversos tipos de fábricas concretas,
- consequentemente, com as diversas famílias de produtos concretos instanciados por elas.
- Podemos entender este padrão como sendo o resultado da junção dos padrões Strategy e Factory Method.

Abstract Factory (Fábrica Abstrata)

Participantes:

- AbstractFactory: Declara uma interface para operações de criação de produtos abstratos.
- ConcreteFactory: Implementa as operações de criação de objetos de produtos concretos.
- AbstractProduct: Declara uma interface para um objeto do tipo produto.
- Product: Define um objeto do tipo produto a ser criado pela correspondente fábrica concreta. Implementa a interface AbstractProduct.
- Client: Utiliza as interfaces declaradas pelas classes AbstractFactory e AbstractProduct.

Abstract Factory (Fábrica Abstrata)



Abstract Factory (Fábrica Abstrata)

- O padrão Abstract Factory está a um nível de abstração maior do que o padrão Factory Method.
- Em outras palavras, o padrão Abstract Factory é uma fábrica de objetos que retorna uma das várias fábricas concretas.
- Para usar a fábrica, é necessário criar uma instância e passá-la para um código/método cliente para o tipo abstrato.

Abstract Factory (Fábrica Abstrata)

- Uma aplicação clássica do Abstract Factory é o caso em que o sistema operacional fornece suporte a múltiplos tipos de interfaces gráficas.
- Desta forma, você diz ao factory que quer que seu sistema operacional se pareça com o Windows e ele retorna a fábrica GUI que retorna os objetos relativos ao Windows.
- Então, quando você requisita um objeto específico como um botão, check-boxes e janelas, a fábrica de GUI retorna as instâncias desses componentes para o sistema operacional.

Abstract Factory (Fábrica Abstrata)

Exemplo:

- Implementação de um software que trabalhe com diversos toolkits gráficos,
 - tais como Motif,
 - GTK+ (GIMP toolkit) usado em um grande número de aplicações e no ambiente de desktop GNOME
 - Qt (cute) largamente no ambiente desktop KDE.
- O cliente (aplicação), interage com o toolkit através de uma fábrica abstrata sem ter conhecimento da implementação das classes concretas (botões, janelas, etc...).

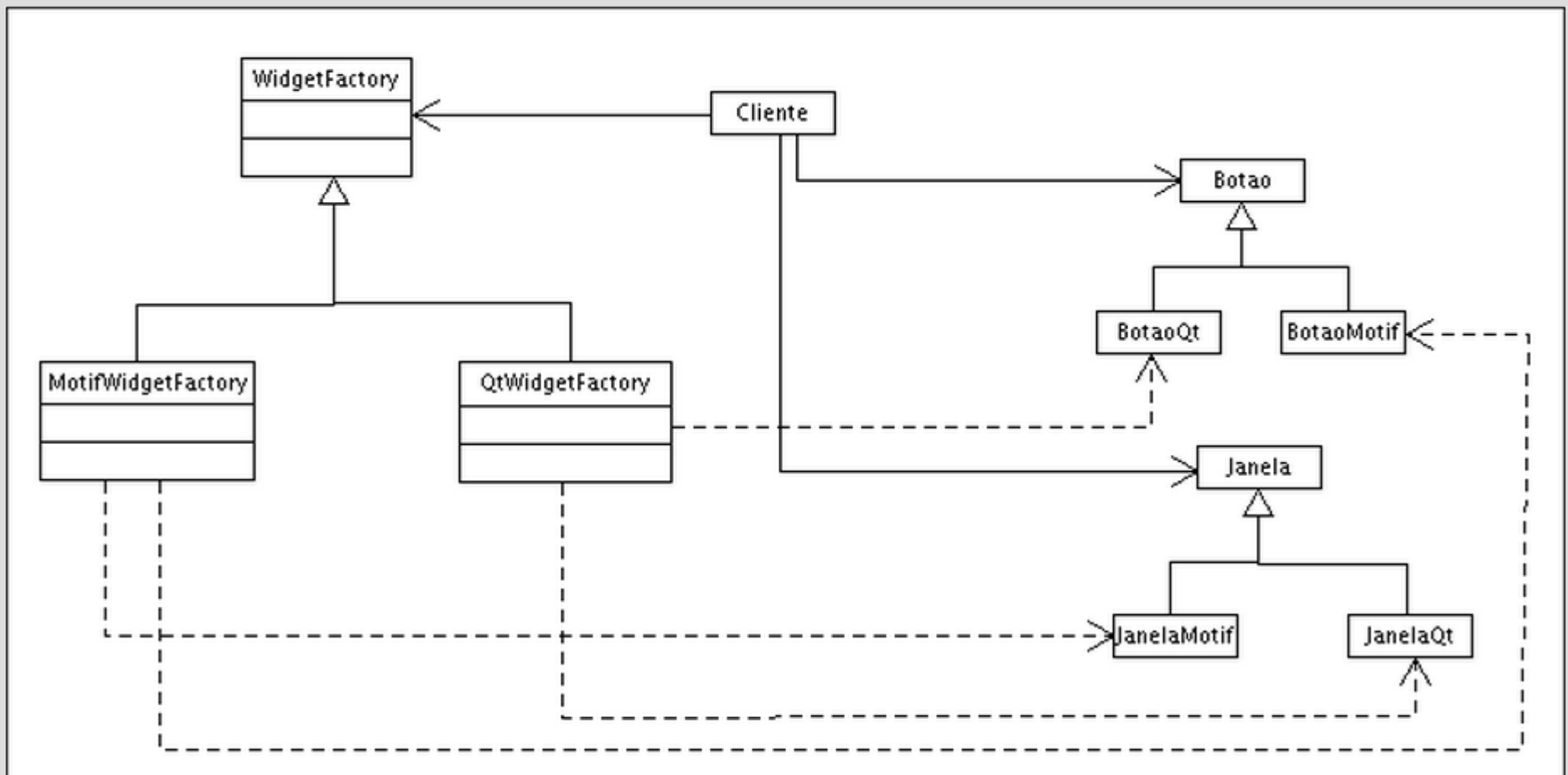
Abstract Factory (Fábrica Abstrata)

- Através do padrão abstract factory podemos implementar fabricas de widgets.
- Um widget é um componente de uma interface gráfica do usuário (GUI), o que inclui janelas, botões, menus, ícones, barras de rolagem, etc..

Abstract Factory (Fábrica Abstrata)

- Para ser portátil entre vários estilos de aparência, uma aplicação não deve codificar rigidamente seus widgets em um determinado padrão.
- Instanciar classes específicas de um tipo de toolkit pela aplicação torna difícil mudar o estilo no futuro.
- Solução: definir uma fabrica abstrata WidgetFactory que declara uma interface para criação de cada tipo de widget.
- Desta forma os clientes ficam independentes do padrão de interação usado no momento.

Abstract Factory (Fábrica Abstrata)



Abstract Factory (Fábrica Abstrata)

```
//abstract factory
abstract class WidgetFactory
{
    public abstract Botao criarBotao();
    public abstract Janela criarJanela();
}
```

Abstract Factory (Fábrica Abstrata)

```
//concrete factory
class MotifWidgetFactory extends WidgetFactory
{
    public Botao criarBotao()    {
        return new BotaoMotif();
    }

    public Janela criarJanela()  {
        return new JanelaMotif();
    }
}
```

Abstract Factory (Fábrica Abstrata)

//concrete factory

```
class QtWidgetFactory extends WidgetFactory
{
    public Botao criarBotao()    {
        return new BotaoQt();
    }
    public Janela criarJanela()  {
        return new JanelaQt();
    }
}
```

Abstract Factory (Fábrica Abstrata)

```
//abstract product
abstract class Botao
{
    public abstract void desenhar();
}

//product
class BotaoMotif extends Botao
{
    public void desenhar() {
        System.out.println("Eu sou um botao Motif!");
    }
}

//product
class BotaoQt extends Botao
{
    public void desenhar() {
        System.out.println("Eu sou um botao Qt!");
    }
}
```

Abstract Factory (Fábrica Abstrata)

```
//abstract product
abstract class Janela
{
    public abstract void desenhar();
}

//product
class JanelaMotif extends Janela
{
    public void desenhar() {
        System.out.println("Eu sou uma janela Motif!");
    }
}

//product
class JanelaQt extends Janela
{
    public void desenhar() {
        System.out.println("Eu sou uma janela Qt!");
    }
}
```


Abstract Factory (Fábrica Abstrata)

```
//Client
public class Aplicacao
{
    public static void main(String[] args)
    {
        WidgetFactory factory = obterFactory(args[0]);
        Botao botao = factory.criarBotao();
        botao.desenhar();
        Janela janela = factory.criarJanela();
        janela.desenhar();
    }

    public static WidgetFactory obterFactory(String s)
    {
        if(s.equals("1") ) {return new MotifWidgetFactory();}
        else {return new QtWidgetFactory();}
    }
}
```

Abstract Factory (Fábrica Abstrata)

- Exercício:
 - Adicione uma fabrica de widgets Gtk+.

Abstract Factory (Fábrica Abstrata)

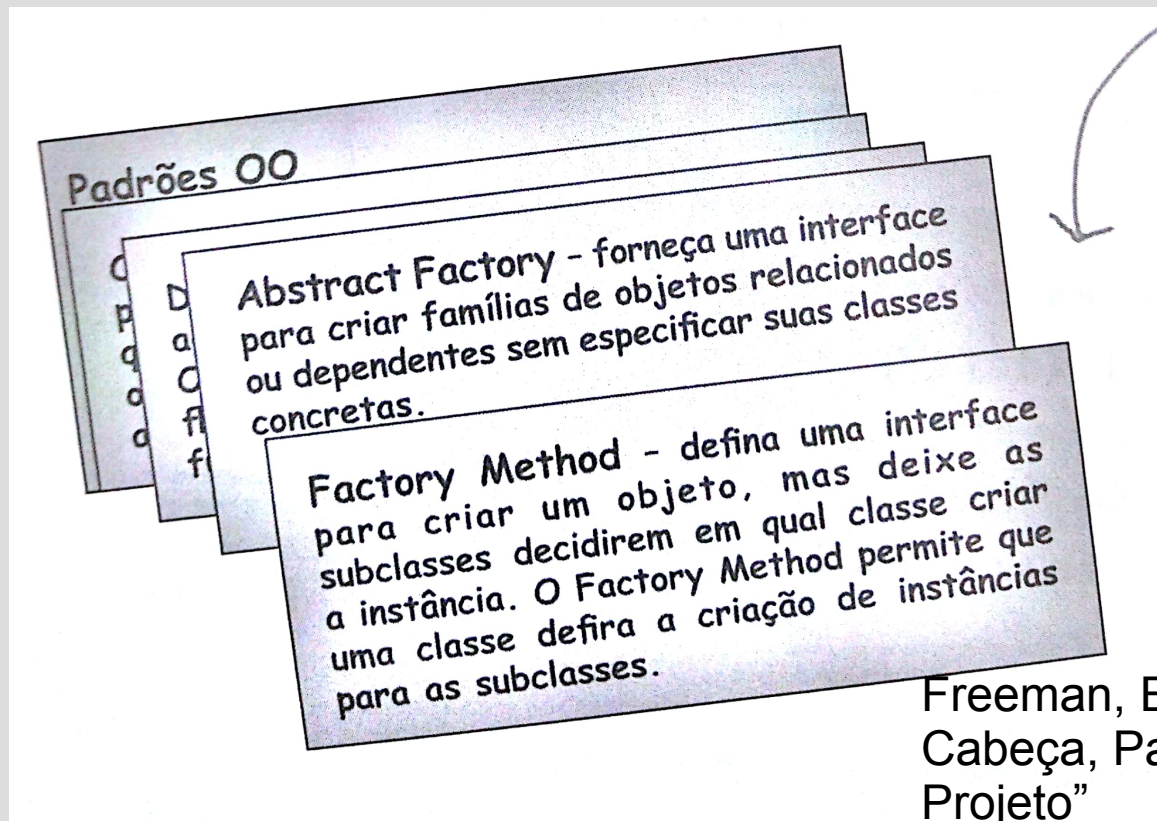
- Use o padrão Abstract factory quando:
 - Um sistema deve ser independente de como seus produtos são criados;
 - Um sistema deve ser configurado com uma família de produtos;
 - Uma família de produtos for projetada para ser usada em conjunto e se necessita garantir essa restrição.

Abstract Factory (Fábrica Abstrata)

- Conclusão:
 - As fábricas abstratas permitem alto nível de desacoplamento do cliente em relação a famílias de objetos, possibilitando flexibilidade na aplicação.
 - Torna fácil a troca de famílias de produtos.

Abstract Factory (Fábrica Abstrata)

- Relação entre Factory Method e Abstract Factory
 - Os métodos de uma abstract factory são implementados como factory methods.



Abstract Factory (Fábrica Abstrata)

- Fim.