// A interface fábrica abstrata declara um conjunto de métodos

// que retorna diferentes produtos abstratos. Estes produtos são

// chamados uma família e estão relacionados por um tema ou

// conceito de alto nível. Produtos de uma família são

// geralmente capazes de colaborar entre si. Uma família de

// produtos pode ter várias variantes, mas os produtos de uma

// variante são incompatíveis com os produtos de outro variante.

interface GUIFactory is

method createButton():Button

method createCheckbox():Checkbox

// As fábricas concretas produzem uma família de produtos que

// pertencem a uma única variante. A fábrica garante que os

// produtos resultantes sejam compatíveis. Assinaturas dos

// métodos fabrica retornam um produto abstrato, enquanto que

// dentro do método um produto concreto é instanciado.

class WinFactory implements GUIFactory is

method createButton():Button is

return new WinButton()

method createCheckbox():Checkbox is

return new WinCheckbox()

// Cada fábrica concreta tem uma variante de produto

// correspondente.

class MacFactory implements GUIFactory is

method createButton():Button is

return new MacButton()

method createCheckbox():Checkbox is

return new MacCheckbox()

// Cada produto distinto de uma família de produtos deve ter uma

// interface base. Todas as variantes do produto devem

// implementar essa interface.

interface Button is

method paint()

// Produtos concretos são criados por fábricas concretas

// correspondentes.

class WinButton implements Button is

method paint() is

// Renderiza um botão no estilo Windows.

class MacButton implements Button is

method paint() is

// Renderiza um botão no estilo macOS.

// Aqui está a interface base de outro produto. Todos os

// produtos podem interagir entre si, mas a interação apropriada

// só é possível entre produtos da mesma variante concreta.

interface Checkbox is

method paint()

class WinCheckbox implements Checkbox is

method paint() is

// Renderiza uma caixa de seleção estilo Windows.

class MacCheckbox implements Checkbox is

method paint() is

// Renderiza uma caixa de seleção no estilo macOS.

// O código cliente trabalha com fábricas e produtos apenas

// através de tipos abstratos: GUIFactory, Button e Checkbox.

// Isso permite que você passe qualquer subclasse fábrica ou de

// produto para o código cliente sem quebrá-lo.

class Application is

private field factory: GUIFactory

private field button: Button

constructor Application(factory: GUIFactory) is

this.factory = factory

method createUI() is

this.button = factory.createButton()

method paint() is

button.paint()

// A aplicação seleciona o tipo de fábrica dependendo da atual

// configuração do ambiente e cria o widget no tempo de execução

// (geralmente no estágio de inicialização).

class ApplicationConfigurator is

method main() is

config = readApplicationConfigFile()

if (config.OS == "Windows") then

factory = new WinFactory()

else if (config.OS == "Mac") then

factory = new MacFactory()

else

throw new Exception("Error! Unknown operating system.")

Application app = new Application(factory)