Exercicio 1 – Abstract Factory

Considere uma fábrica de salgados. Essa fábrica produz 2 tipos de salgados:

- Coxinha

- Empada

Cada salgado é produzido com ingredientes conforme as regiões de Goiás e Ceará, onde:

Coxinha de Goiás:

* Massa: Fina
* Molho: Pequi
* Recheio: Frango

Coxinha do Ceará:

* Massa: Grossa
* Molho: Pimenta
* Recheio: Carne Seca

Empada de Goiás:

* Massa: Podre
* Molho: Tomate
* Recheio: Calabresa

Empada do Ceará:

* Massa: Crepioca
* Molho: Camarao
* Recheio: Peixe

Considerando que cada região tem seu tipo de salgado. Construa um programa em Java, que permita as seguintes construções:

* Modelos:
  + **Salgado (Abstract)**: atributos protected String -> massa, molho e recheio
  + **Coxinha** (subclasse de Salgado).
  + **Empada** (subclasse de Salgado).
  + Salgado possui o atributo: **protected SalgadoIngredientesFactory ingredientes**
* Fábricas:
  + **public abstract class SalgadoIngredientesFactory**, com os métodos:
    - **public abstract String criarMassa**();
    - **public abstract String criarMolho**();
    - **public abstract String criarRecheio**();
  + Fábrica concreta **GoiasCoxinhaIngredientesFactory** (subclasse de SalgadoIngredientesFactory). As implementações concretas dessa classe retornam as strings correspondentes à massa, molho e recheio da coxinha de Goiás.
  + Fábrica concreta **CearaCoxinhaIngredientesFactory** (subclasse de SalgadoIngredientesFactory). As implementações concretas dessa classe retornam as strings correspondentes à massa, molho e recheio da coxinha do Ceara.
  + Fábrica concreta **GoiasEmpadaIngredientesFactory** (subclasse de SalgadoIngredientesFactory). As implementações concretas dessa classe retornam as strings correspondentes à massa, molho e recheio da empada de Goiás.
  + Fábrica concreta **CearaEmpadaIngredientesFactory** (subclasse de SalgadosFactory). As implementações concretas dessa classe retornam as strings correspondentes à massa, molho e recheio da empada do Ceara.
  + **SalgadosFactory**: fábrica principal de salgados. Essa fábrica usa a **SalgadoIngredientesFactory** (composição) para criar um atributo chamado **ingredientes**. Essa fábrica também possui um atributo chamado **salgadinho** do tipo Salgado.
    - Métodos:
    - **public void assarSalgado(String tipo, String regiao)**: utilize esse método para criar um salgadinho conforme tipo (coxinha ou empada) e região (Ceara ou Goias).
    - **public SalgadoIngredientesFactory getIngredientesCoxinha(String regiao)**: utilize esse método para retornar um objeto do tipo SalgadoIngredientesFactory de acordo com a região (Ceara ou Goias) específico para a coxinha.
    - **public SalgadoIngredientesFactory getIngredientesEmpada(String regiao)**: utilize esse método para retornar um objeto do tipo SalgadoIngredientesFactory de acordo com a região (Ceara ou Goias) específico para a empada.
    - **public Salgado getSalgadinho()** : para retornar o atributo salgadinho
    - **@Override public String toString()**: para imprimir a mensagem:

“Massa: “ imprima a massa;

“Molho: “ imprima o molho;

“Recheio: “ imprima o recheio;

Crie as fábricas em um pacote de fábricas e os modelos em um pacote de modelos. Para testar o código, crie 4 fábricas de salgados, uma para cada tipo de salgado e região.

Imprima as informações conforme exemplo a seguir:

Coxinha de Goias

Massa: Fina

Molho: Pequi

Recheio: Frango

Coxinha do Ceara

Massa: Grossa

Molho: Pimenta

Recheio: Carne Seca

Empada do Goias

Massa: Podre

Molho: Tomate

Recheio: Calabresa

Empada do Ceara

Massa: Crepioca

Molho: Camarao

Recheio: Peixe