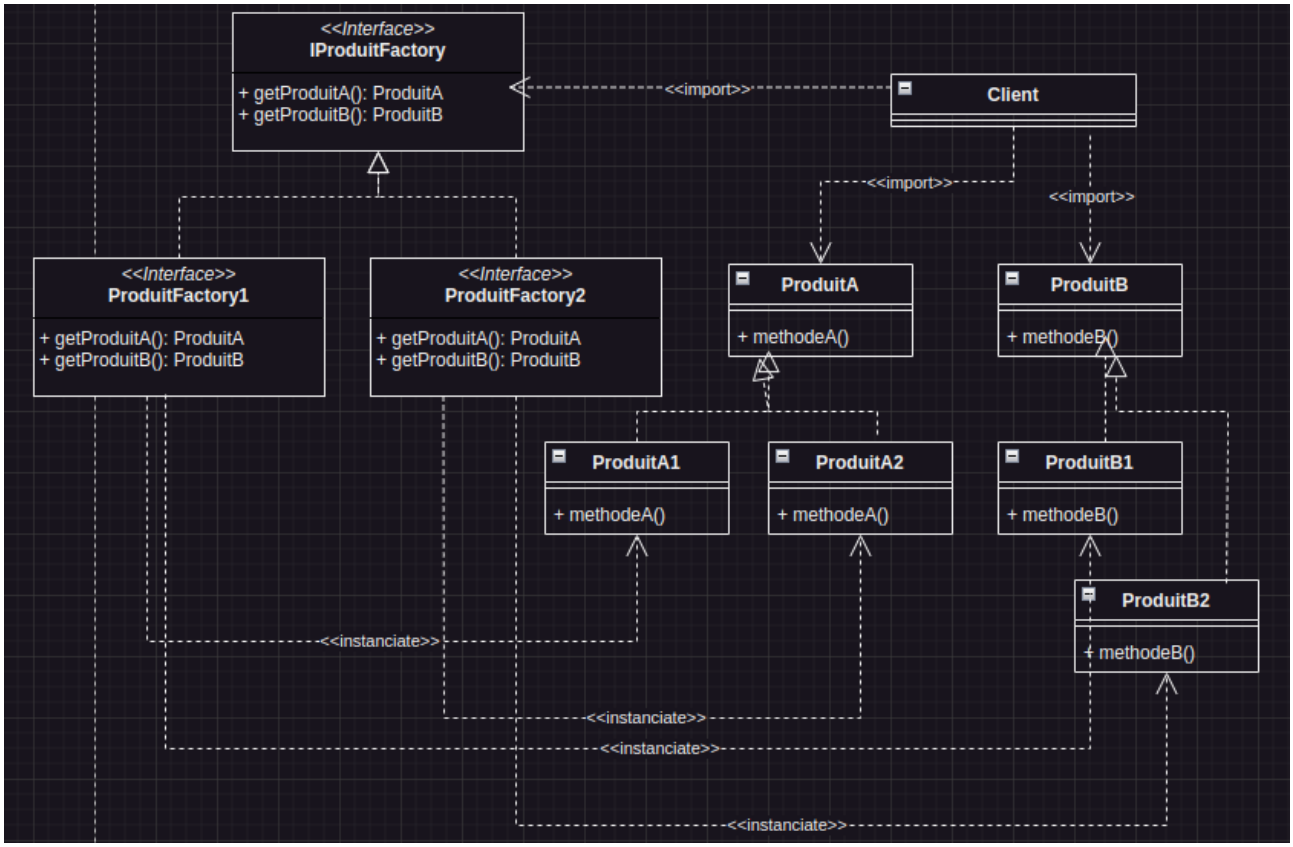
Vendredi 06 Octobre 2023

Abstract Factory

Model générique :



Code source :

Client :

```java
public class Client {
Run | Debug
public static void main(String[] args) {
IProduitFactory produitFactory1 = new ProduitFactory1();
IProduitFactory produitFactory2 = new ProduitFactory2();
ProduitA produitA = null;
ProduitB produitB = null;
System.out.println("Utilisation de la premiere fabrique");
produitA = produitFactory1.getProduitA();
produitB = produitFactory1.getProduitB();
produitA.methodeA();
produitB.methodeB();
System.out.println("Utilisation de la seconde fabrique");
produitA = produitFactory2.getProduitA();
produitB = produitFactory2.getProduitB();

produitA.methodeA();
produitB.methodeB();

    }
}
```

IProduitFactory :

```
public interface IProduitFactory {
public ProduitA getProduitA();
public ProduitB getProduitB();
}
```

ProduitFactory1 :

```
public class ProduitFactory1 implements IProduitFactory {
public ProduitA getProduitA() {
return new ProduitA1();
}
public ProduitB getProduitB() {
return new ProduitB1();
}
}
```

ProduitFactory2 :

```
public class ProduitFactory2 implements IProduitFactory {
public ProduitA getProduitA() {
return new ProduitA2();
}
public ProduitB getProduitB() {
    return new ProduitB2();
}
}
```

ProduitA :

```
public abstract class ProduitA {
    public abstract void methodeA();
}
```

ProduitB :

```java
public abstract class ProduitB {
    public abstract void methodeB();
}
```

ProduitA1 :

```java
import construction.abstractFactory.ProduitA;

public class ProduitA1 extends ProduitA{

    public void methodeA(){
        System.out.println("ProduitA1.methodeA()");
    }
}
```

ProduitA2 :

```java
public class ProduitA2 extends ProduitA {
public void methodeA() {
System.out.println("ProduitA2.methodeA()");
    }
}
```

ProduitB1 :

```java
import construction.abstractFactory.produitB;

public class ProduitB1 extends ProduitB{

    public void methodeB(){
        System.out.println("ProduitB1.methodeB()");
    }
}
```

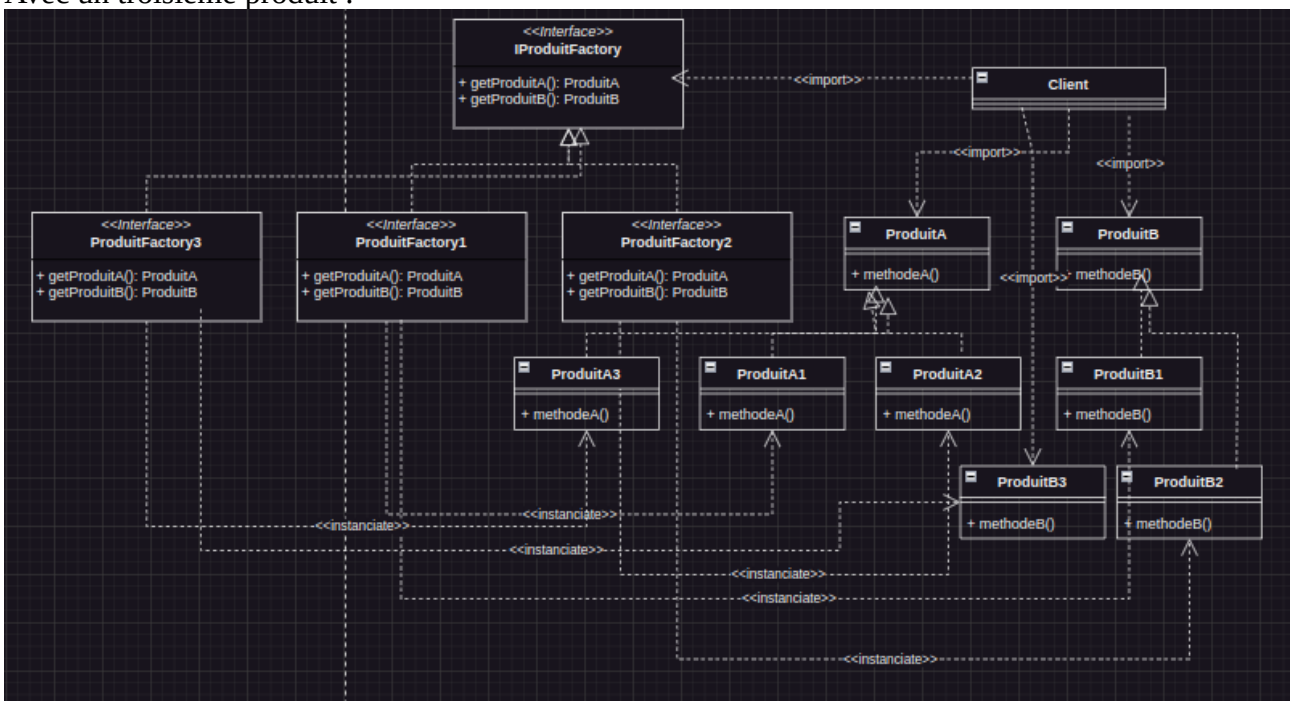
ProduitB2 :

```java
import construction.abstractFactory.produitB;

public class ProduitB2 extends ProduitB{

    public void methodeB(){
        System.out.println("ProduitB2.methodeB()");
    }
}
```

Avec un troisième produit :



ProduitA3 :

```java
public class ProduitA3 extends ProduitA {
public void methodeA() {
System.out.println("ProduitA3.methodeA()");
    }
}
```

Client :

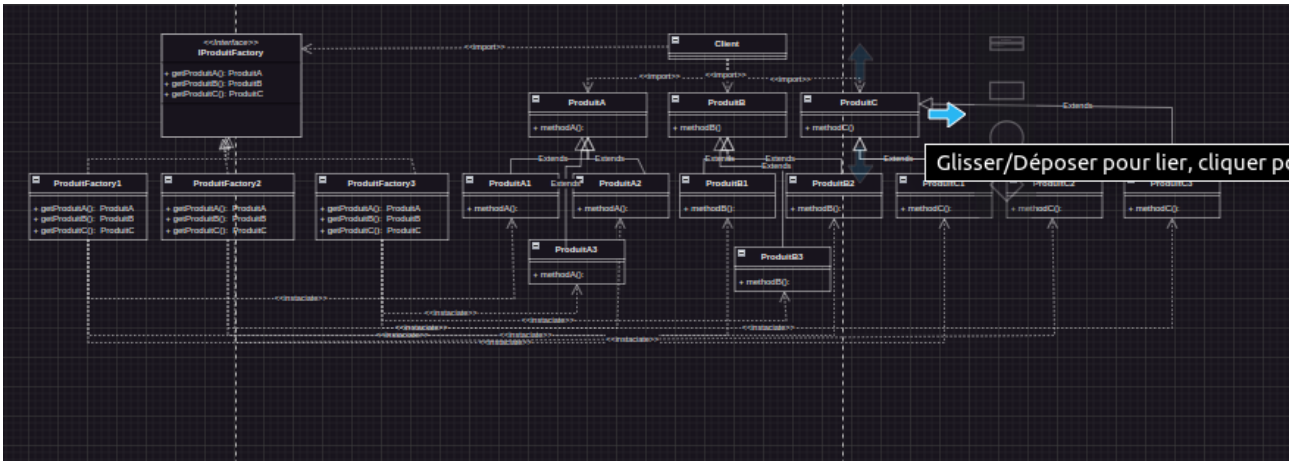
```java
public class Client2 {
Run | Debug
public static void main(String[] args) {
IProduitFactory produitFactory1 = new ProduitFactory1();
IProduitFactory produitFactory2 = new ProduitFactory2();
IProduitFactory produitFactory3 = new ProduitFactory3();
ProduitA produitA = null;
ProduitB produitB = null;

System.out.println("Utilisation de la premiere fabrique");
produitA = produitFactory1.getProduitA();
produitB = produitFactory1.getProduitB();
produitA.methodeA();
produitB.methodeB();
System.out.println("Utilisation de la seconde fabrique");
produitA = produitFactory2.getProduitA();
produitB = produitFactory2.getProduitB();
produitA.methodeA();
produitB.methodeB();

System.out.println("Utilisation de la troisième fabrique");
produitA = produitFactory3.
produitB = produitFactory3.getProduitB();
produitA.methodeA();
produitB.methodeB();
    }
}
```

ProduitB IProduitFactory.getProduitB()

Abstract Factory 3ème famille :



code ProduitC :

```java
public abstract class ProduitC {
    public abstract void methodeC();
}
```

Code Client 3 avec insertion de la 3 ème famille :

```java
public class Client3 {

    Run | Debug
    public static void main(String[] args) {
        IProduitFactory produitFactory1 = new ProduitFactory1();
        IProduitFactory produitFactory2 = new ProduitFactory2();
        IProduitFactory produitFactory3 = new ProduitFactory3();
        ProduitA produitA = null;
        ProduitB produitB = null;
        ProduitC produitC = null;

        System.out.println("Utilisation de la premiere fabrique");
        produitA = produitFactory1.getProduitA();
        produitB = produitFactory1.getProduitB();
        produitC = produitFactory1.getProduitC();
        produitA.methodeA();
        produitB.methodeB();
        produitC.methodeC();
        System.out.println("Utilisation de la seconde fabrique");
        produitA = produitFactory2.getProduitA();
        produitB = produitFactory2.getProduitB();
        produitC = produitFactory2.getProduitC();
        produitA.methodeA();
        produitB.methodeB();
        produitC.methodeC();

        System.out.println("Utilisation de la troisième fabrique");
        produitA = produitFactory3.getProduitA();
        produitB = produitFactory3.getProduitB();
        produitC = produitFactory3.getProduitC();
        produitA.methodeA();
        produitB.methodeB();
```

```java
produitA.methodeA();
produitB.methodeB();
produitC.methodeC();
System.out.println("Utilisation de la seconde fabrique");
produitA = IProduitFactory produitFactory2 - Client3.main(String[])
produitB = produitFactory2.getProduitB();
produitC = produitFactory2.getProduitC();
produitA.methodeA();
produitB.methodeB();
produitC.methodeC();

System.out.println("Utilisation de la troisième fabrique");
produitA = produitFactory3.getProduitA();
produitB = produitFactory3.getProduitB();
produitC = produitFactory3.getProduitC();
produitA.methodeA();
produitB.methodeB();
produitC.methodeC();

    }
}
```