Université Abou Beckr Belkaid
Master 2 Génie Logiciel
Année Universitaire 2018/2019

Examen Final

—

Les technique de construction d'architectures logicielles avancées

—

Remarques :

- Les documents ne sont pas autorisés ainsi que les appareils électroniques (PC, Tablette, téléphone..).
- La lisibilité et la clarté de vos réponses et de votre code sont très importantes. Une réponse pas claire ne sera pas prise en compte lors de la correction.

Exercice 1 :

Nous souhaitons maitre en place un système de gestion d'adresse et de numéro de téléphone. Pour chaque adresse et numéro de téléphone il existe deux formats :

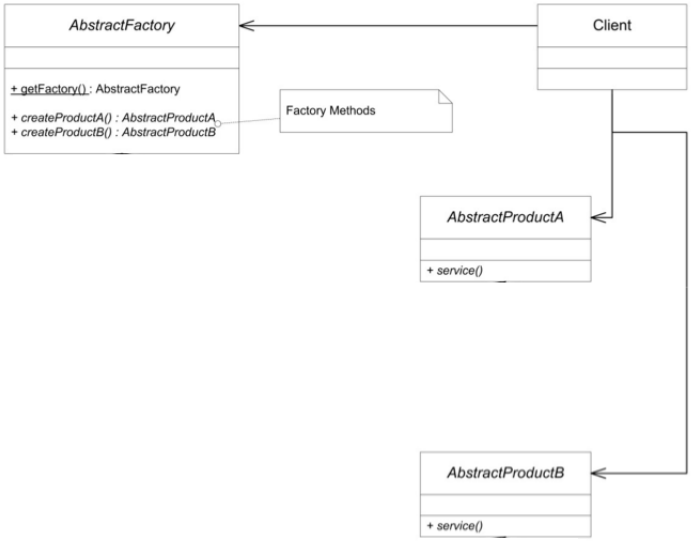
- Le format algérien :
  - Adresse : Rue, Code postal, Ville, Pays.
  - Téléphone : 213.XX.XX.XX.XX
- Le format américain :
  - Adresse : Rue, Ville, Région, Code postal, Pays.
  - Téléphone : (1).XXX.XXX.XXXX

Afin d'abstraire les différents types d'adresses et de numéros de téléphone aux clients, nous proposons d'utiliser le design pattern Abstract Factory.

Questions :

1 A quelle groupe de design pattern appartient Abstract Factory. Donner deux avantages de ce pattern.

2 En utilisant Abstract Factory, proposer une conception à ce système en complétant la conception suivante et en remplaçant AbstractProductA et AbstractProductB par des noms de classes adéquates :



.

3 Proposer une implémentation de la classe *AbstractFactory* et une implémentation du main liée à la classe *Client*.

Exercice 2 :

Nous avons une classe Java simple nommée *Point*. Un objet de type *Point* est représenté par deux coordonnées *x* et *y*. Le code de la classe point est donné comme suite :

```
class Point {

  protected int x = 0;
  protected int y = 0;

  public int getX() {
    return x;
  }

  public int getY() {
    return y;
  }

  public void setRectangular(int newX, int newY) {
    setX(newX);
    setY(newY);
  }

  public void setX(int newX) {
    x = newX;
  }

  public void setY(int newY) {
    y = newY;
  }

  public void offset(int deltaX, int deltaY) {
    setRectangular(x + deltaX, y + deltaY);
  }

  public String toString() {
    return "(" + getX() + ", " + getY() + ")" ;
  }
}
```

.

Le but de l'exercice est de transformer la classe Point en une classe Java bean en utilisant la programmation orientée aspect.

Questions :

1 Donner la définition de la programmation orientée aspect.
2 Donner les caractéristiques d'une classe Java Bean.
3 Quelles sont les propriétés à ajouter à la classe Point pour la transformer en une classe Java Bean ?
4 En utilisant la programmation orientée aspect, proposer une solution à ce problème (en donnant le code associé).
3 En utilisant la programmation orienté aspect, transformer les attributs x et y en des propriétés liées (en donnant le code associé).

Bon courage et bonne continuation.