

Copies Écran pour le Laboratoire 3 du cours Algorithmes et Modèles de Programmation

Présentation du Résultat

```
Construction d'une maison en bois  
Construction étage en bois terminée  
Construction murs en bois terminée  
Construction toit en bois terminée  
  
Construction d'une maison en béton  
Construction étage en béton terminée  
Construction murs en béton terminée  
Construction toit en béton terminée
```

Classe App (comprends le 'main')

```
public class App {  
    public static void main(String[] args) {  
        new ClientMaison();  
    }  
}
```

Classe ClientMaison (le 'client')

```
public class ClientMaison {

    public FabriqueMaison uneFabrique = new FabriqueMaison();

    public MonteurMaisonBois unMonteurMaisonEnBois = new MonteurMaisonBois();
    public MonteurMaisonBeton unMonteurMaisonEnBeton = new MonteurMaisonBeton();

    public Maison uneMaison;

    ClientMaison(){
        uneFabrique.setMonteurMaison(unMonteurMaisonEnBois); // La 'fabrique' (en réalité le directeur) reçoit le plan (un monteur concret) pour
                                                                // construire le produit, soit une maison en bois.

        uneFabrique.construireMaison();                        // La fabrique ordonne la construction d'une maison en bois (le processus s'enclenche).

        uneMaison = uneFabrique.getMaison();                 // On récupère l'objet complété

        System.out.println(uneMaison.getMaisonRepresentation()); // Affichage à la console du getRepresentation de la Maison (lui-même composé
                                                                    // des getRepresentations de ses composants (etage,murs,toit).

        System.out.println();

        uneFabrique.setMonteurMaison(unMonteurMaisonEnBeton); // Construction d'une maison en béton cette fois-ci.
        uneFabrique.construireMaison();
        uneMaison = uneFabrique.getMaison();
        System.out.println(uneMaison.getMaisonRepresentation());
    }
}
```

Classe FabriqueMaison (le 'directeur')

```
public class FabriqueMaison {  
    private MonteurMaison monteur;  
  
    public void setMonteurMaison(MonteurMaison unMonteur) {  
        this.monteur = unMonteur;  
    }  
  
    public Maison getMaison() {  
        return this.monteur.getMaison();  
    }  
  
    public void construireMaison() {  
        this.monteur.creerNouvelleMaison();  
        this.monteur.choisirMateriel();  
        this.monteur.monterEtage();  
        this.monteur.monterMurs();  
        this.monteur.monterToit();  
    }  
}
```

Classe MonteurMaisonBois (un 'monteur concret')

```
public class MonteurMaisonBois extends MonteurMaison {  
    @Override  
    public void monterEtage() {  
        maison.setEtage("étage en bois");  
    }  
  
    @Override  
    public void monterMurs() {  
        maison.setMurs("murs en bois");  
    }  
  
    @Override  
    public void monterToit() {  
        maison.setToit("toit en bois");  
    }  
  
    @Override  
    public void choisirMateriel() {  
        maison.setMateriel("bois");  
    }  
}
```

Classe MonteurMaisonBeton (un 'monteur concret')

```
public class MonteurMaisonBeton extends MonteurMaison {  
  
    @Override  
    public void monterEtage() {  
        maison.setEtage("étage en béton");  
    }  
  
    @Override  
    public void monterMurs() {  
        maison.setMurs("murs en béton");  
    }  
  
    @Override  
    public void monterToit() {  
        maison.setToit("toit en béton");  
    }  
  
    @Override  
    public void choisirMateriel() {  
        maison.setMateriel("béton");  
    }  
  
}
```

Classe MonteurMaison (le 'monteur abstrait')

```
public abstract class MonteurMaison {  
  
    protected Maison maison;  
  
    public Maison getMaison() {  
        return this.maison;  
    }  
  
    public void creerNouvelleMaison() {  
        this.maison = new Maison();  
    }  
  
    public abstract void choisirMateriel();  
    public abstract void monterEtage();  
    public abstract void monterMurs();  
    public abstract void monterToit();  
  
}
```

Classe Maison (le 'produit')

```
public class Maison implements IEtage, IMurs, IToit {

    protected String unMateriel;
    protected String unEtage;
    protected String desMurs;
    protected String unToit;

    public void setMateriel(String unMateriel) {
        this.unMateriel = unMateriel;
    }

    public void setEtage(String unMateriel) {
        this.unEtage = unMateriel;
    }

    public void setMurs(String unMateriel) {
        this.desMurs = unMateriel;
    }

    public void setToit(String unMateriel) {
        this.unToit = unMateriel;
    }

    @Override
    public String getEtageRepresentation() {
        return "Construction "+this.unEtage+" terminée\n";
    }

    @Override
    public String getMursRepresentation() {
        return "Construction "+this.desMurs+" terminée\n";
    }
}
```

```
@Override
public String getToitRepresentation() {
    return "Construction "+this.unToit+" terminée\n";
}

public String getMaisonRepresentation() {
    String rep = "Construction d'une maison en "+this.unMateriel+"\n";
    rep += this.getEtageRepresentation();
    rep += this.getMursRepresentation();
    rep += this.getToitRepresentation();
    return rep;
}
```

