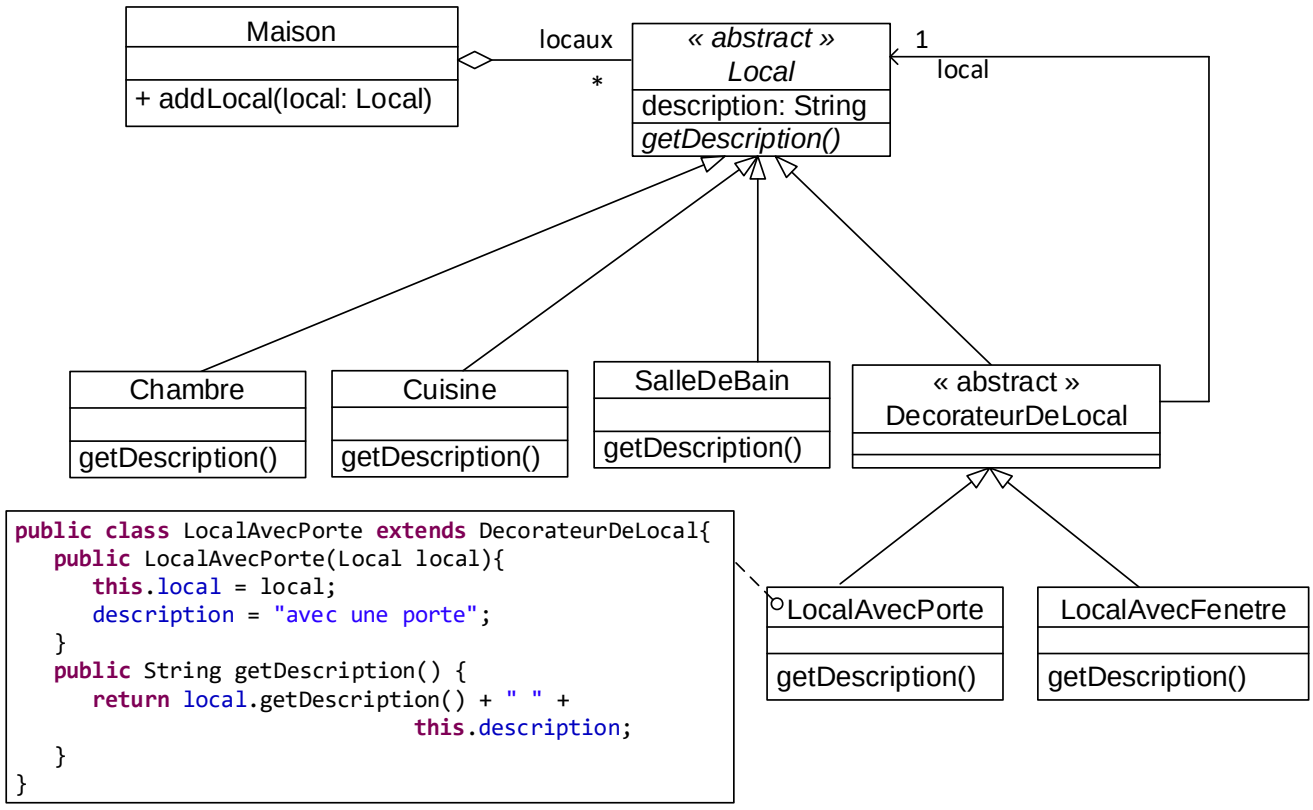
Solution des exercices sur le patron Décorateur

Exercice 1

Diagramme de classes



Code source

La classe Local :

```
public abstract class Local {
	protected String description;
	public abstract String getDescription();
}
```

La classe Chambre :

```
public class Chambre extends Local{

	public Chambre(){
		description = "Une simple chambre";
	}
```

```
	public String getDescription() {
		return description;
	}

}
```

La classe DecorateurDeLocal :

```
public abstract class DecorateurDeLocal extends Local{

	protected Local local;

}
```

La classe LocalAvecFenetre :

```
public class LocalAvecFenetre extends DecorateurDeLocal{


	public LocalAvecFenetre(Local local){
		this.local = local;
		description = "avec une fenêtre";
	}

	public String getDescription() {
		return local.getDescription() + " " + this.description;
	}

}
```

La classe LocalAvecPorte :

```
public class LocalAvecPorte extends DecorateurDeLocal{

	public LocalAvecPorte(Local local){
		this.local = local;
		description = "avec une porte";
	}

	public String getDescription() {
		return local.getDescription() + " " + this.description;

	}

}
```

La classe Maison :

```
import java.util.ArrayList;
import java.util.Iterator;
```

```
public class Maison {

	ArrayList<Local> locaux = new ArrayList<Local>();

	public void addLocal(Local local){
		locaux.add(local);
	}

	Iterator<Local> getIterator(){
		return locaux.iterator();
	}

}
```

Une classe pour tester le tout :

```
import java.util.Iterator;

public class ConstructeurDeMaison {

	public static void main(String args[]) {

		System.out.println("========Voilà ma première maison===============");
		Maison maMaison1 = new Maison();

		// je veux une chambre avec une fenêtre et une porte
		Chambre maChambre1 = new Chambre();
		LocalAvecFenetre local1_f = new LocalAvecFenetre(maChambre1);
		LocalAvecPorte local1_P = new LocalAvecPorte(local1_f);
		maMaison1.addLocal(local1_P);

		// je veux une salle de bain avec une porte
		SalleDeBain maSalle1 = new SalleDeBain();
		LocalAvecPorte local2_P = new LocalAvecPorte(maSalle1);

		maMaison1.addLocal(local2_P);

		// je veux une chambre avec une fenêtre et deux portes
		Chambre maChambre2 = new Chambre();
		LocalAvecFenetre local3_f = new LocalAvecFenetre(maChambre2);
		LocalAvecPorte local3_P = new LocalAvecPorte(local3_f);
		LocalAvecPorte local3_PP = new LocalAvecPorte(local3_P);
		maMaison1.addLocal(local3_PP);

		Iterator<Local> iterateurDesLocaux = maMaison1.getIterator();

		while(iterateurDesLocaux.hasNext()){
			System.out.println(iterateurDesLocaux.next().getDescription());
		}
	}
}
```

Exercice 2 (Exercices 5.12 et 5.13 du livre de Horstman)

La solution de cet exercice est extraite du site du livre :
http://www.horstmann.com/oodp2/solutions/solutions.html

Le code zippé de la solution se trouve sur le site du cours (sous l'onglet « Exercices », le fichier zip nommé « encrypt_decrypt »).