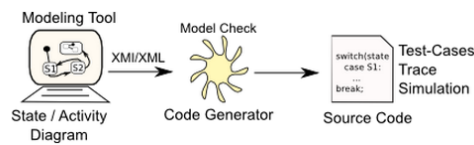
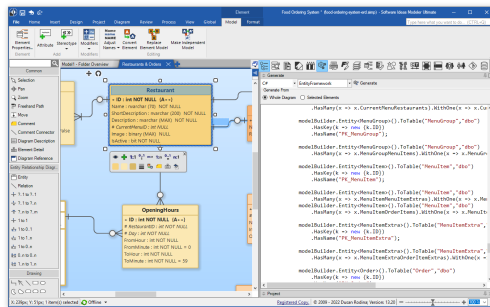


# Exemple de Génération et Rétro-Conception avec UML

La génération et la rétro-conception sont des processus importants dans le cycle de vie du développement logiciel. La génération consiste à créer du code à partir de modèles UML, tandis que la rétro-conception consiste à créer des modèles UML à partir de code existant. Voici un exemple pour illustrer ces concepts.

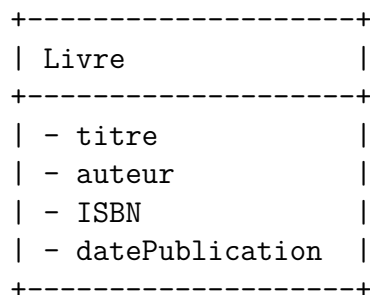


## 1. Génération de Code à partir de Modèles UML



### a. Modèle UML : Diagramme de Classes

Nous avons un diagramme de classes pour un système de gestion de bibliothèque.



```

| + ajouterLivre()    |
| + supprimerLivre() |
| + mettreAJourLivre() |
+-----+
|
|
|
+-----+
| Emprunt             |
+-----+
| - dateEmprunt        |
| - dateRetour         |
| - livreID            |
| - membreID           |
+-----+
| + enregistrerEmprunt() |
| + enregistrerRetour()  |
+-----+
|
|
|
+-----+
| Membre               |
+-----+
| - nom                |
| - identifiant        |
| - adresse            |
| - numéroTéléphone    |
+-----+
| + ajouterMembre()    |
| + supprimerMembre()  |
| + mettreAJourMembre() |
+-----+

```

## b. Génération de Code

Nous utilisons un outil comme **Umbrello** pour générer du code Java à partir du diagramme de classes.

```

// Livre.java
public class Livre {
    private String titre;
    private String auteur;
    private String ISBN;
    private Date datePublication;

    public void ajouterLivre() {
        // Implémentation de l'ajout d'un livre
    }

    public void supprimerLivre() {
        // Implémentation de la suppression d'un livre
    }

    public void mettreAJourLivre() {
        // Implémentation de la mise à jour d'un livre
    }
}

// Emprunt.java
public class Emprunt {
    private Date dateEmprunt;
    private Date dateRetour;
    private int livreID;
    private int membreID;

    public void enregistrerEmprunt() {
        // Implémentation de l'enregistrement d'un emprunt
    }

    public void enregistrerRetour() {
        // Implémentation de l'enregistrement d'un retour
    }
}

// Membre.java
public class Membre {

```

```

private String nom;
private int identifiant;
private String adresse;
private String numeroTelephone;

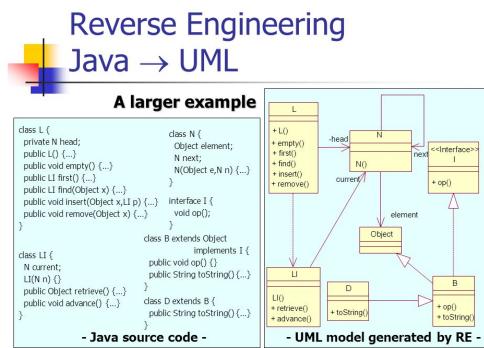
public void ajouterMembre() {
    // Implémentation de l'ajout d'un membre
}

public void supprimerMembre() {
    // Implémentation de la suppression d'un membre
}

public void mettreAJourMembre() {
    // Implémentation de la mise à jour d'un membre
}
}

```

## 2. Rétro-Conception à partir du Code



### a. Code Existant

Nous avons un code Java existant pour le système de gestion de bibliothèque.

```

// Livre.java
public class Livre {
    private String titre;
    private String auteur;

```

```

    private String ISBN;
    private Date datePublication;

    public void ajouterLivre() {
        // Implémentation de l'ajout d'un livre
    }

    public void supprimerLivre() {
        // Implémentation de la suppression d'un livre
    }

    public void mettreAJourLivre() {
        // Implémentation de la mise à jour d'un livre
    }
}

// Emprunt.java
public class Emprunt {
    private Date dateEmprunt;
    private Date dateRetour;
    private int livreID;
    private int membreID;

    public void enregistrerEmprunt() {
        // Implémentation de l'enregistrement d'un emprunt
    }

    public void enregistrerRetour() {
        // Implémentation de l'enregistrement d'un retour
    }
}

// Membre.java
public class Membre {
    private String nom;
    private int identifiant;
    private String adresse;
    private String numeroTelephone;

```

```

public void ajouterMembre() {
    // Implémentation de l'ajout d'un membre
}

public void supprimerMembre() {
    // Implémentation de la suppression d'un membre
}

public void mettreAJourMembre() {
    // Implémentation de la mise à jour d'un membre
}
}

```

## b. Rétro-Conception

Nous utilisons un outil comme **Umbrello** pour analyser le code et générer automatiquement des diagrammes UML. Voici à quoi pourrait ressembler le diagramme de classes généré :

```

+-----+
| Livre          |
+-----+
| - titre        |
| - auteur       |
| - ISBN         |
| - datePublication |
+-----+
| + ajouterLivre() |
| + supprimerLivre() |
| + mettreAJourLivre() |
+-----+
|
|
|
+-----+
| Emprunt        |
+-----+

```

```

| - dateEmprunt      |
| - dateRetour       |
| - livreID          |
| - membreID         |
+-----+
| + enregistrerEmprunt() |
| + enregistrerRetour()  |
+-----+
|
|
|
+-----+
| Membre              |
+-----+
| - nom               |
| - identifiant       |
| - adresse           |
| - numeroTelephone   |
+-----+
| + ajouterMembre()   |
| + supprimerMembre() |
| + mettreAJourMembre() |
+-----+

```

## Conclusion

La génération de code à partir de modèles UML permet de passer rapidement de la conception à l'implémentation, tandis que la rétro-conception permet de visualiser et de documenter le code existant sous forme de modèles UML. Les outils comme **Umbrello** facilitent ces processus en offrant des fonctionnalités pour la génération de code et l'ingénierie inverse, aidant ainsi les développeurs à maintenir une bonne documentation et une conception cohérente tout au long du cycle de vie du développement logiciel.