

TP 3 : diagramme de classe

I. Création du projet

1. Créez un projet TP3 et cochez la case Java Project. Vérifiez que le module « JavaDesigner » est dans le catalogue des modules (menu Configuration > Modulecataglogue). S'il ne l'est pas rajoutez-le.

2. Créez un diagramme de classe.

II. Création d'une classe

3. Créez une classe « Personne ».

4. Créez un attribut dans la classe en cliquant-déposant l'icône (A : Attribute) sur la classe dans le diagramme.

5. Renommez l'attribut en « num ».

6. Changez son type à « integer » (sélectionner l'attribut et changer le type sur l'onglet Properties).

III. Génération de code Java

7. Dans le diagramme, sélectionnez la classe « Personne » et dans l'onglet Java de la partie inférieure de la fenêtre de l'application, sélectionnez Java element. Ceci indique que la classe fera l'objet d'une génération de code. Un petit symbole apparaît dans la classe pour signifier que cet élément du diagramme donne lieu à une génération de code. On peut également cocher cette case au niveau du paquetage (sélectionnable dans l'arborescence) pour que tous les éléments du paquetage fassent l'objet d'une génération de code.


8. Sélectionnez l'attribut « num : integer » et indiquez dans l'onglet Java qu'à cet attribut devront correspondre des accesseurs en Java (cochez Getter et Setter). Les accesseurs en question apparaissent dans le modèle (voir l'arborescence) mais pas dans le diagramme.

9. Pour faire apparaître les accesseurs, glissez-les de l'arborescence vers le diagramme.

10. Sélectionnez la classe (dans le diagramme ou l'arborescence) et au moyen du menu contextuel, générez un code Java, (Java Designer -> Generate).

11. Sélectionnez la classe (dans le diagramme ou l'arborescence) et au moyen du menu contextuel, éditer un code Java, (Java Designer -> Edit).

12. Le code source généré par Modelio apparaît dans une fenêtre. Analysez le code et comparez-le avec la représentation de la classe dans le diagramme.

Remarque: Sélectionnez la classe, dans l'onglet Java des icônes apparaissent (en haut à droite ). Ces icônes offrent les mêmes fonctionnalités que le menu contextuel JavaDesigner.

IV. Reverse engineering

Ce terme désigne la possibilité que les modifications dans les modèles aient un impact direct sur le code et que simultanément les modifications sur le code aient un impact sur les modèles. De cette manière, on peut coder et modéliser en même temps.

13. Dans le code, définissez les accesseurs comme publics. Vous pouvez ajouter d'autres attributs (exemples : nom, age, etc.) et d'autres méthodes.

14. Lorsque vous sauvegardez, le modèle est automatiquement mis à jour. Vous pouvez aussi utiliser le menu contextuel ou le bouton de l'onglet Java pour mettre à jour le modèle à partir des sources.

V. Associations et code source

15. Créez une autre classe « Adresse » avec les attributs suivants et générez son code.

- `private String numero;`
- `private String rue;`
- `private String codePostale;`
- `private String ville;`
- `private String pays;`

16. Créez une Association entre la classe « Personne » et la classe « Adresse ». Générez le code pour prendre en compte l'association. Analysez les deux classes « Personne » et « Adresse ». Que remarquez-vous ?

17. Changez les cardinalités de l'association pour permettre à une Personne d'avoir plusieurs adresses. Générez à nouveau le code et analysez-le. Qu'est-ce qui a changé ?

18. Sélectionnez l'association et indiquez dans l'onglet Java que des accesseurs sont requis. Générez à nouveau le code et analysez-le.

VI. Ajout de méthode à partir des sources

19. Ouvrez le fichier Personne.java et ajoutez un constructeur.

20. Mettez à jour le modèle pour que la nouvelle méthode soit intégrée au modèle.

VII. Élaboration du diagramme de classe

21. Représentez le diagramme de classe d'une entreprise qui emploie plusieurs personnes. Chaque personne employée a un supérieur hiérarchique et peut avoir des adjoints.

Note : les classes « Personne » et « Adresse » font partie du diagramme.