

SAE 3.01 Développement d'applications

Liste des fonctionnalités :

- Affichage de l'arborescence d'un fichier donné dans un TreeView.
- Drag and Drop de l'arborescence au diagramme.
- Déplacement d'une classe sur le diagramme
- Affichage de plusieurs classes.
- Affichage des relations entre les classes (héritage,implémentation..) avec les flèches.
- Le déplacement d'une classe.
- La suppression de la classe lorsqu'elle est glissée sur l'arborescence.
- Affichage d'un menu contextuel lorsqu'un clic droit est effectué
- Suppression d'une ou de toutes les classes (boutons menu contextuel)
- Sauvegarde du diagramme en image depuis le menu contextuel
- Sauvegarde du diagramme en fichier plantuml
- Générer le squelette du code source correspondant au diagramme
- Possibilité d'afficher les classes externes notamment les classes héritant des éléments graphiques JavaFX avec le menu contextuel.
- Ajouter une classe directement au diagramme depuis le menu contextuel.
- Ajouter une relation entre plusieurs classes (et donc une flèche) depuis le menu contextuel
- Ajouter des attributs et méthodes à une classe depuis le menu contextuel
- Affichage d'un diagramme si tout un dossier est sélectionné (clic droit sur le dossier).
- Bouton d'options pour la sauvegarde, le chargement du diagramme, la modification des paramètres d'affichage, ainsi que le choix du répertoire courant de la TreeView.
- Onglets (vue diagramme javafx ou vue PlantUml)
- Menu contextuel au clic droit



Répartition des tâches :

Léo :

- Gestion des classes : attributs, méthodes...
- Relations entre les classes
- Affichage des flèches entre les classes
- Menu déroulant sur une classe
- Déplacement d'une classe
- Choix du répertoire de travail
- Ajouter des nouvelles classes/methodes/...
- Montrer les relations externes

Quentin :

- Export de l'introspection (créée par Elias) dans un fichier
- Sauvegarde du diagramme en fichier plantuml
- Sauvegarde du diagramme en image depuis le menu contextuel
- Onglets (vue diagramme javafx ou vue PlantUml)
- Suppression d'une ou de toutes les classes (boutons menu contextuel)
- Menu contextuel au clic droit
- Générer le squelette du code source correspondant au diagramme

Elias :

- Affichage graphique d'une classe
- Génération de l'introspection avec l'affichage textuel de tous les éléments d'une classe
- Gestion des coordonnées d'une classe (affichage avec un clic)
- Affichage de plusieurs classes
- déplacement basique des classes
- Gestion des dimensions d'une boîte
- Diagramme de classe à partir d'un dossier
- Possibilité d'afficher/masquer certains paramètres
- (dezoom)

Eloi :

- Sauvegarde d'un diagramme
- Drag and drop (arborescence->diagramme)
- Suppression avec drag and drop(diagramme->arborescence)

Tâche dont nous sommes fiers :

Léo : J'ai fait l'affichage des flèches sur la vue graphique, je suis content de la manière dont j'ai réussi à gérer les placements des flèches et l'affichage des différents types de flèches. J'ai fait en sorte que les flèches soient décalées lorsqu'il y en a plusieurs entre les mêmes classes pour ne pas qu'il y ait de flèches superposées.

Quentin : Ma deuxième fonctionnalité préférée est la génération du squelette du code source correspondant au diagramme. J'aurais bien aimé vous parler de la Vue PlantUML intégrée dans un onglet de l'application. Malheureusement, cela ne fonctionne que sur mon ordinateur par souci de version PlantUML, c'est pourquoi je ne la considère pas comme entièrement valide.

Elias : J'ai ajouté un prompt pour choisir d'afficher ou masquer les flèches, les attributs hérités, ainsi que les getters/setters/constructeurs. Cette option permet de diminuer la taille du graphique et m'a demandé de modifier la logique derrière certaines méthodes notamment pour les attributs, qui sont désormais stockés dans deux listes différentes : une contenant les attributs hérités, l'autre non.

Eloi : J'ai effectué la fonctionnalité de la sauvegarde qui permet d'enregistrer le diagramme dans un fichier là où on le souhaite. La fonctionnalité a été faite pour sauvegarder la hashmap des classes présentes dans la vue et les relations entre les classes. Une fois tout récupéré, tout est mis dans le fichier à l'emplacement que l'utilisateur souhaite avoir le fichier de sauvegarde. Et pour importer une sauvegarde, il suffit de sélectionner le fichier correspondant. Une fois sélectionné, le programme va tirer les classes du fichier. Si le fichier a été créé dans l'application, on va récupérer le type de la classe puis la faire passer dans une méthode pour la créer. Dans le cas contraire, on ajoute la classe. Une fois toutes les classes faites on reprend toutes les relations.

Eléments modifié par rapport à l'étude préalable :

Dans l'étude préalable, nous pensions que les flèches seraient mieux gérées avec un patron stratégie, pour séparer les différents types de flèches.

En les implémentant, nous nous sommes rendus compte que les inconvénients de cette méthode pesaient plus que les avantages de ce patron dans ce contexte.

Nous n'avons également pas mis la possibilité de déplacer les flèches, nous avons préféré nous concentrer sur les autres fonctionnalités.

Patron de conception mise en oeuvre :

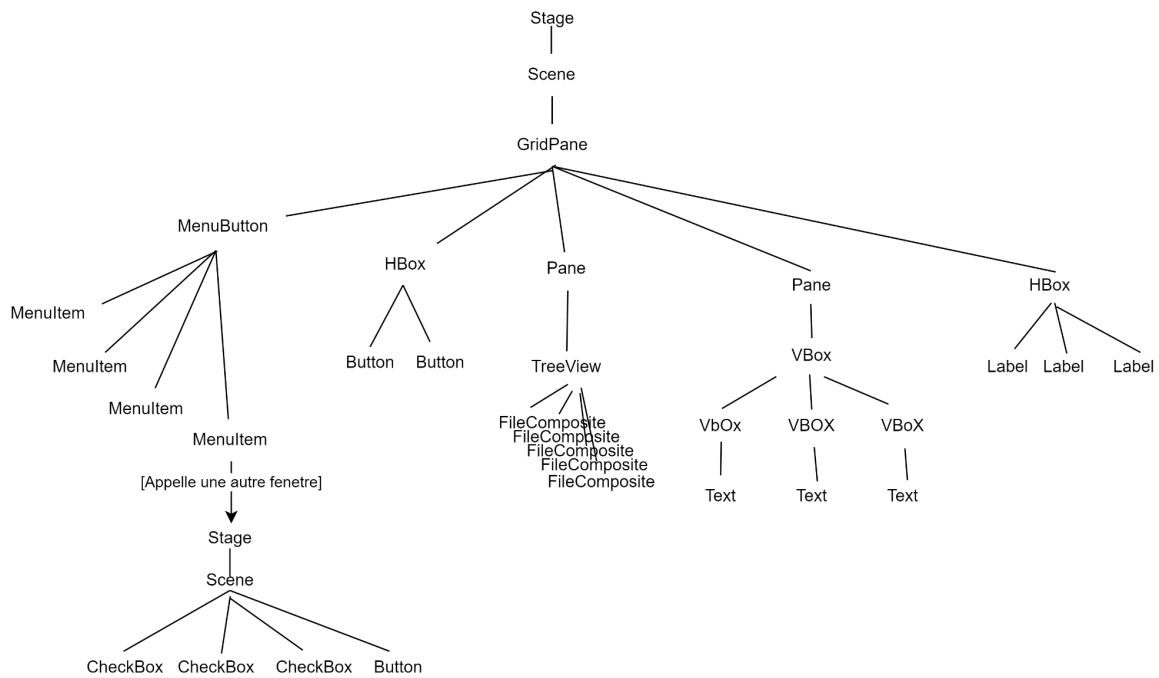
MVC

Introspection

Composite

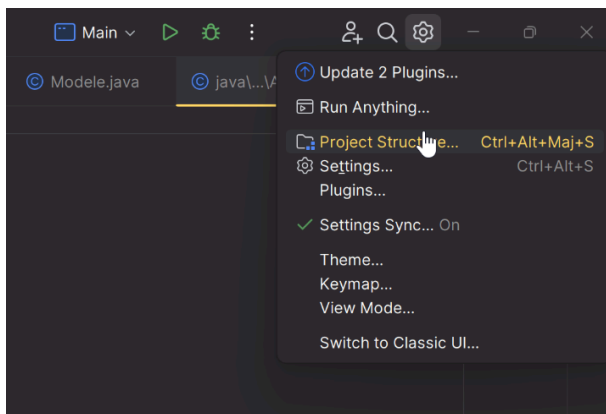
Stratégie (FileComposite avec Fichier et Dossier)

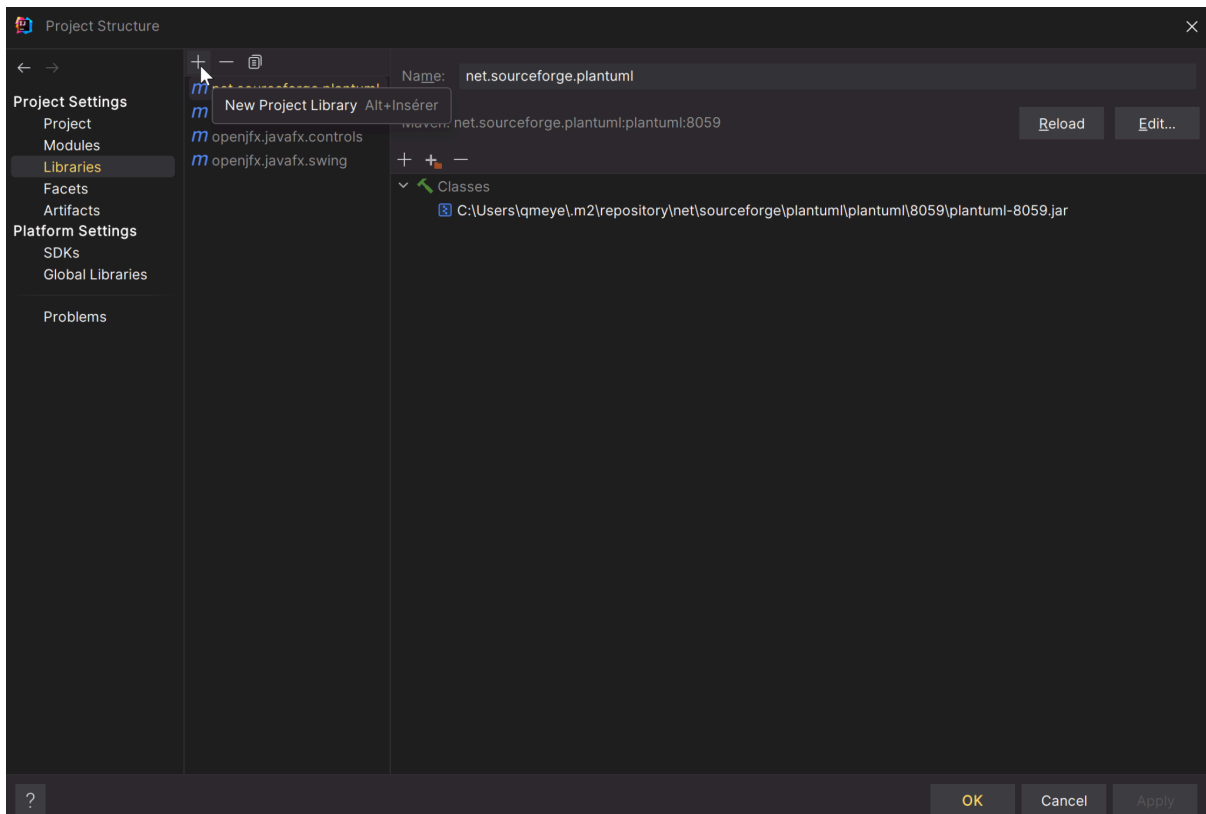
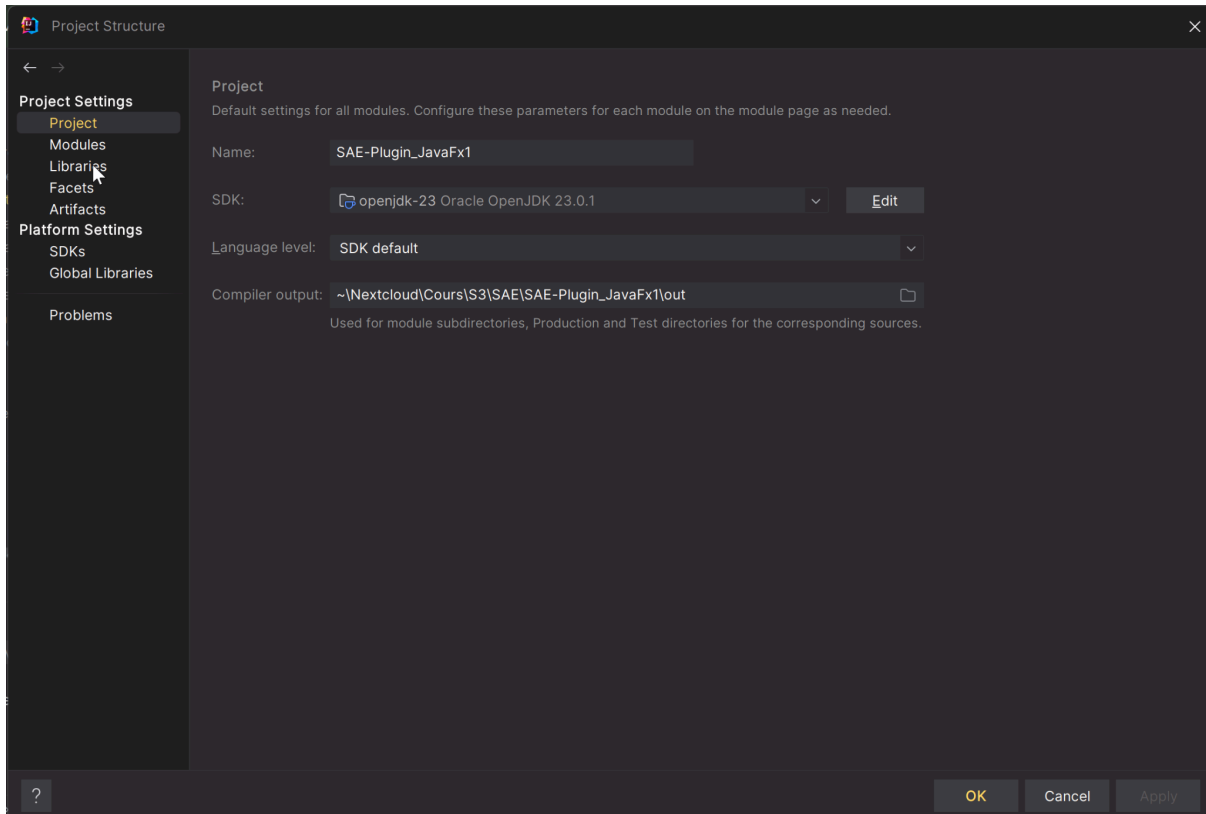
Graphe de scène de l'interface graphique :

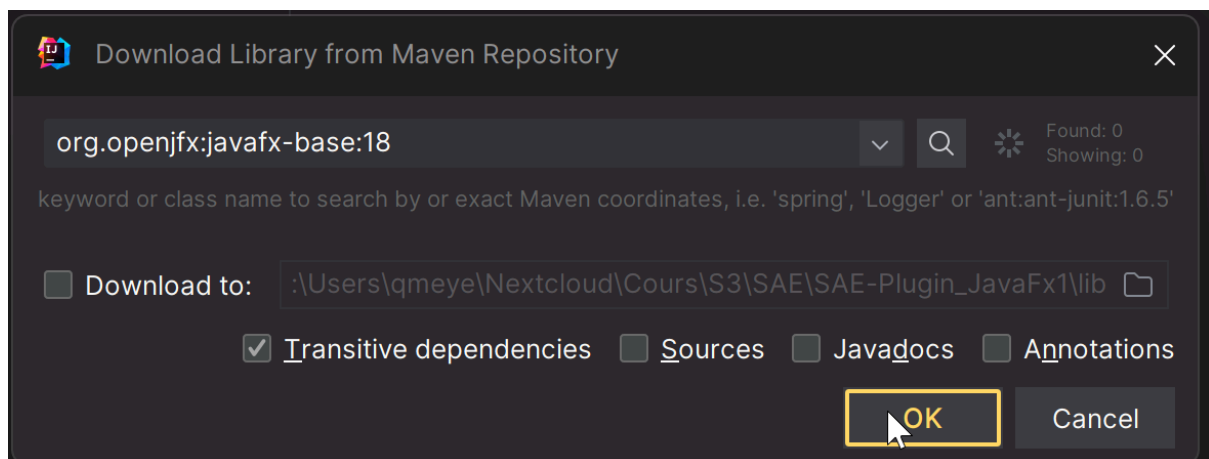
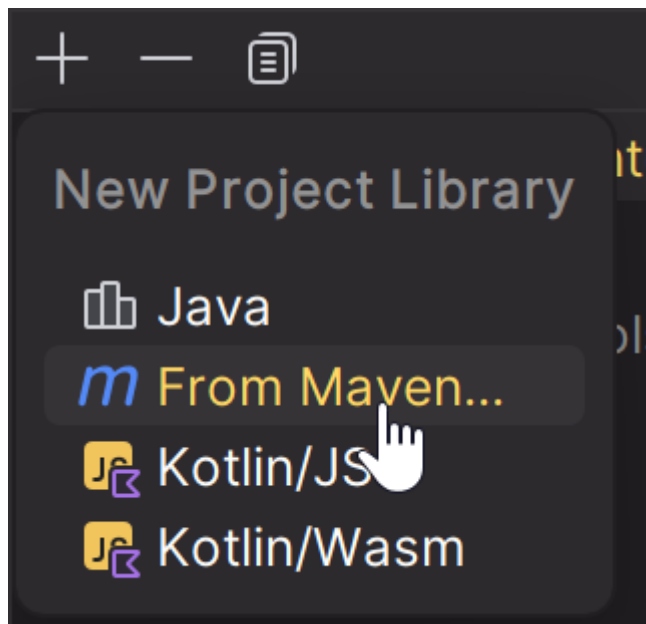


Mode d'emploi de l'application :

- Ajouter les librairies au project structure :







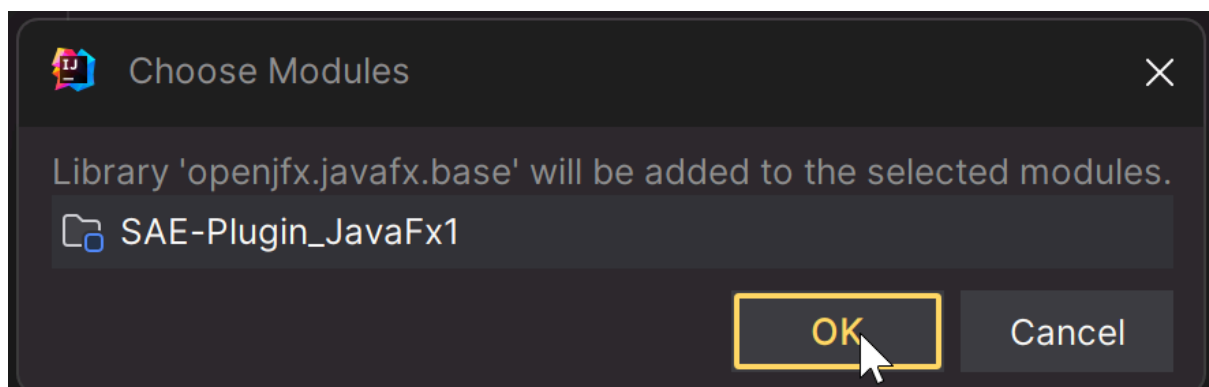
Faire ceci pour toutes ces libraires :

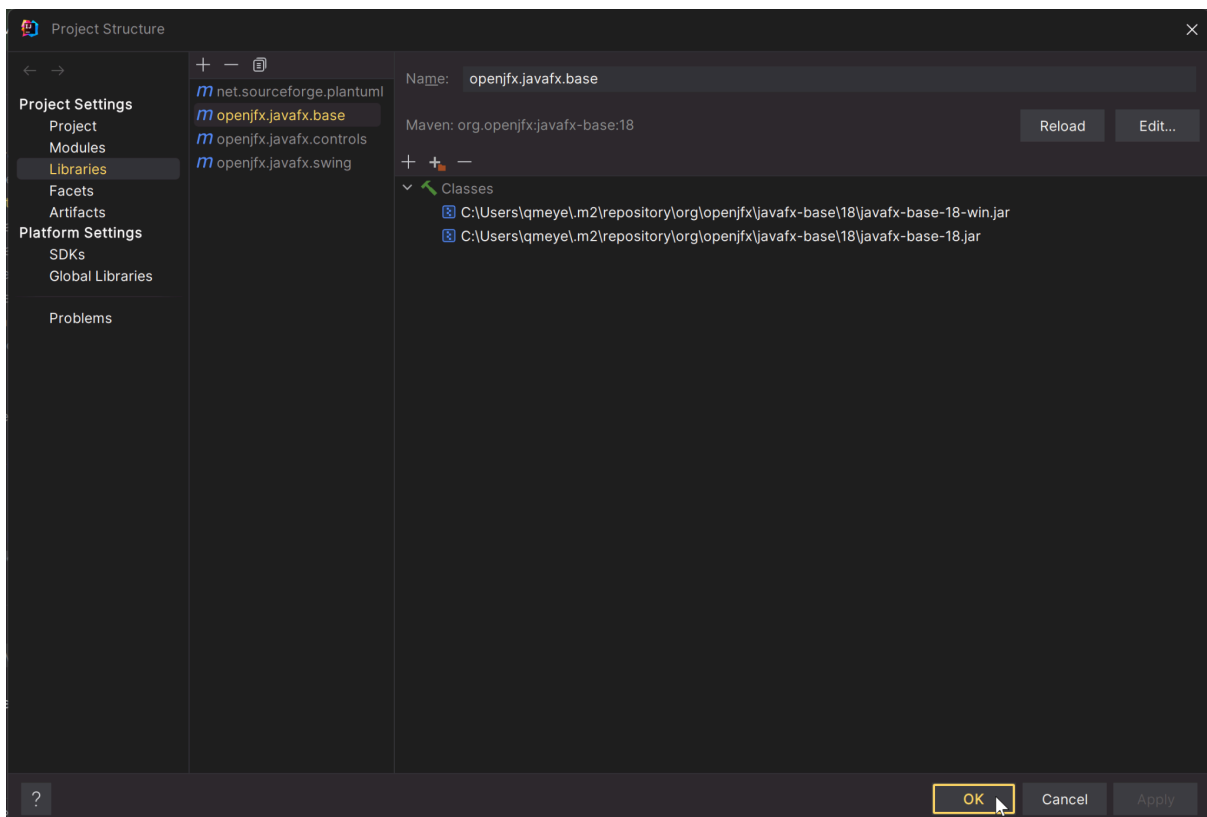
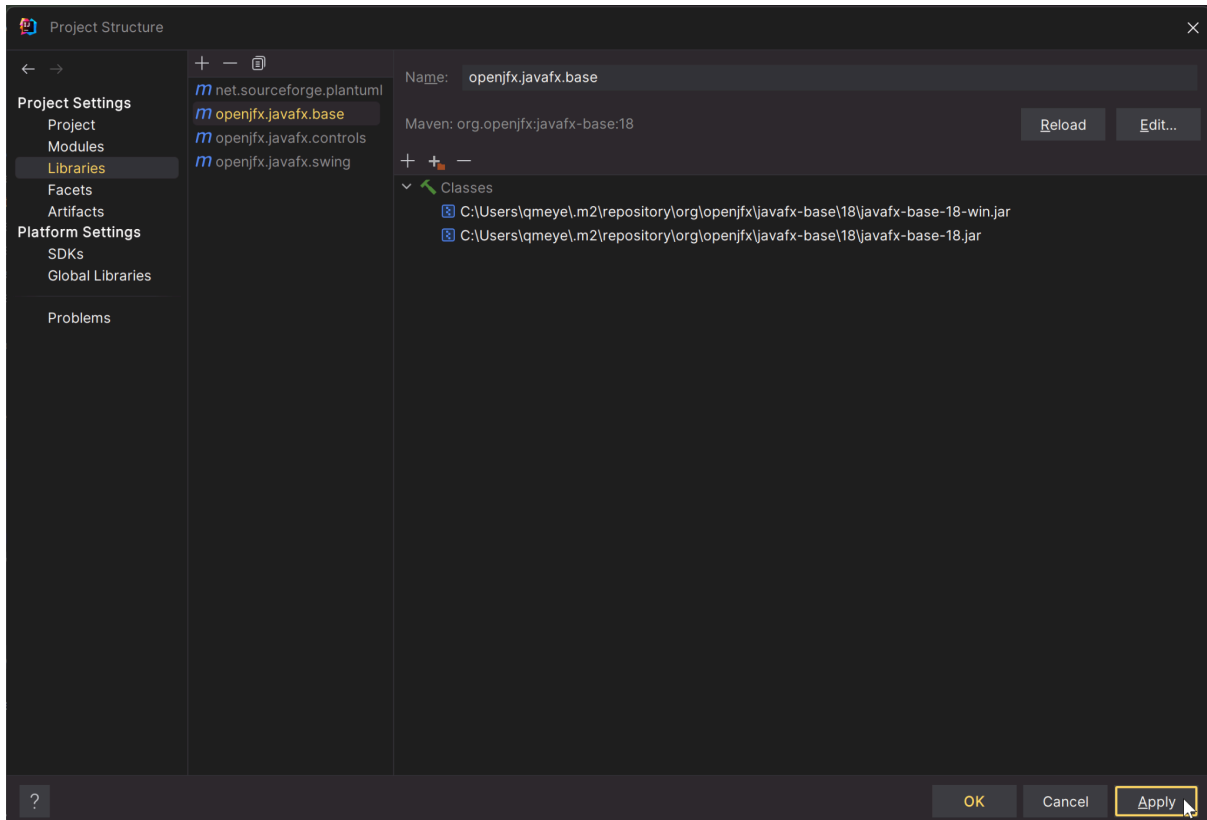
org.openjfx:javaafx-base:18

org.openjfx:javaafx-controls:18

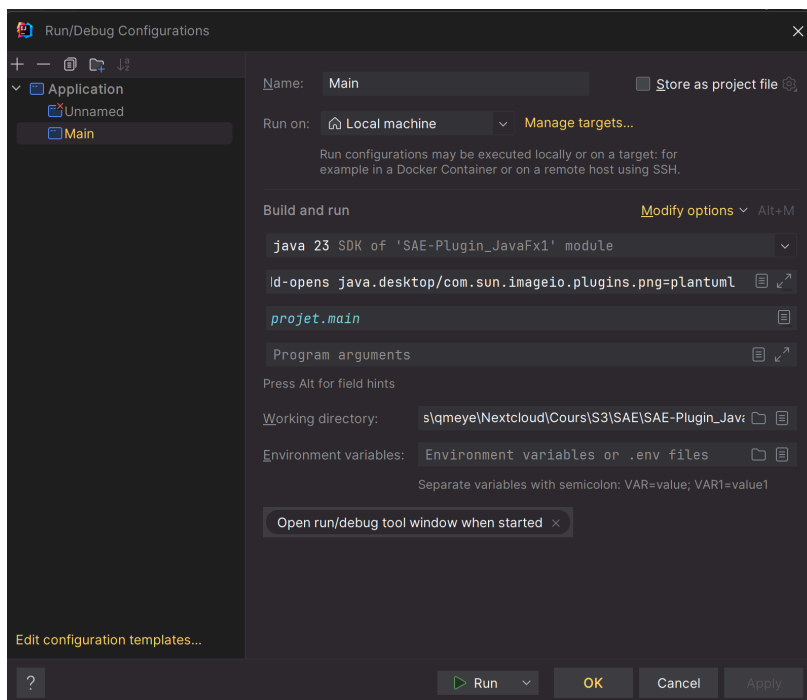
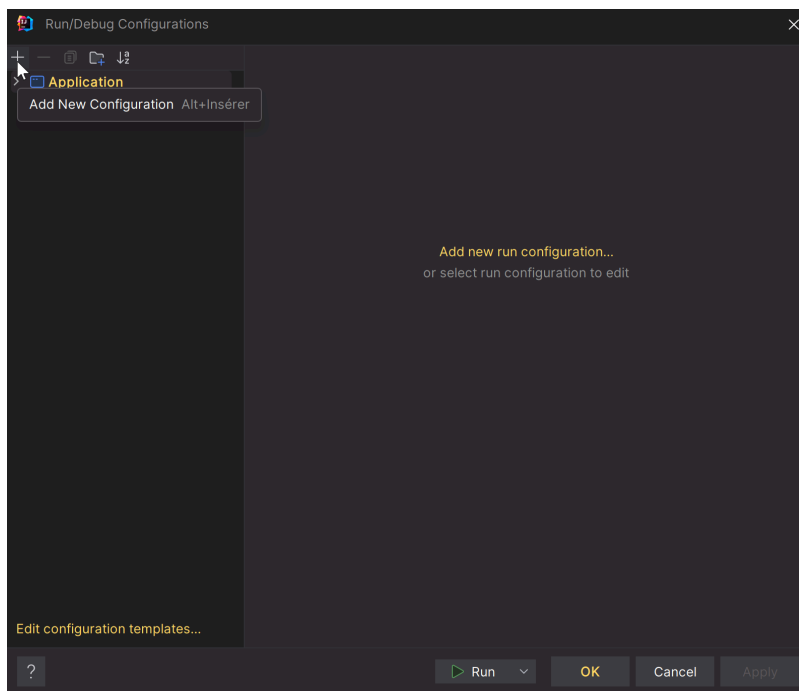
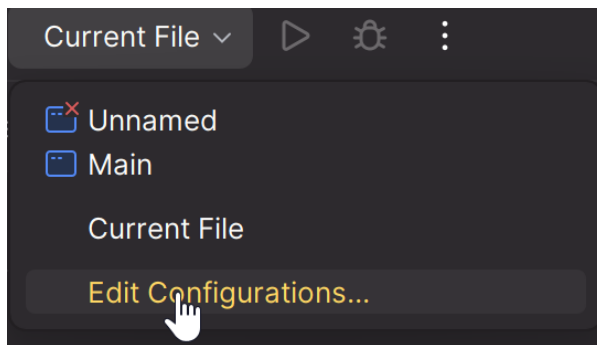
org.openjfx:javaafx-swing:18

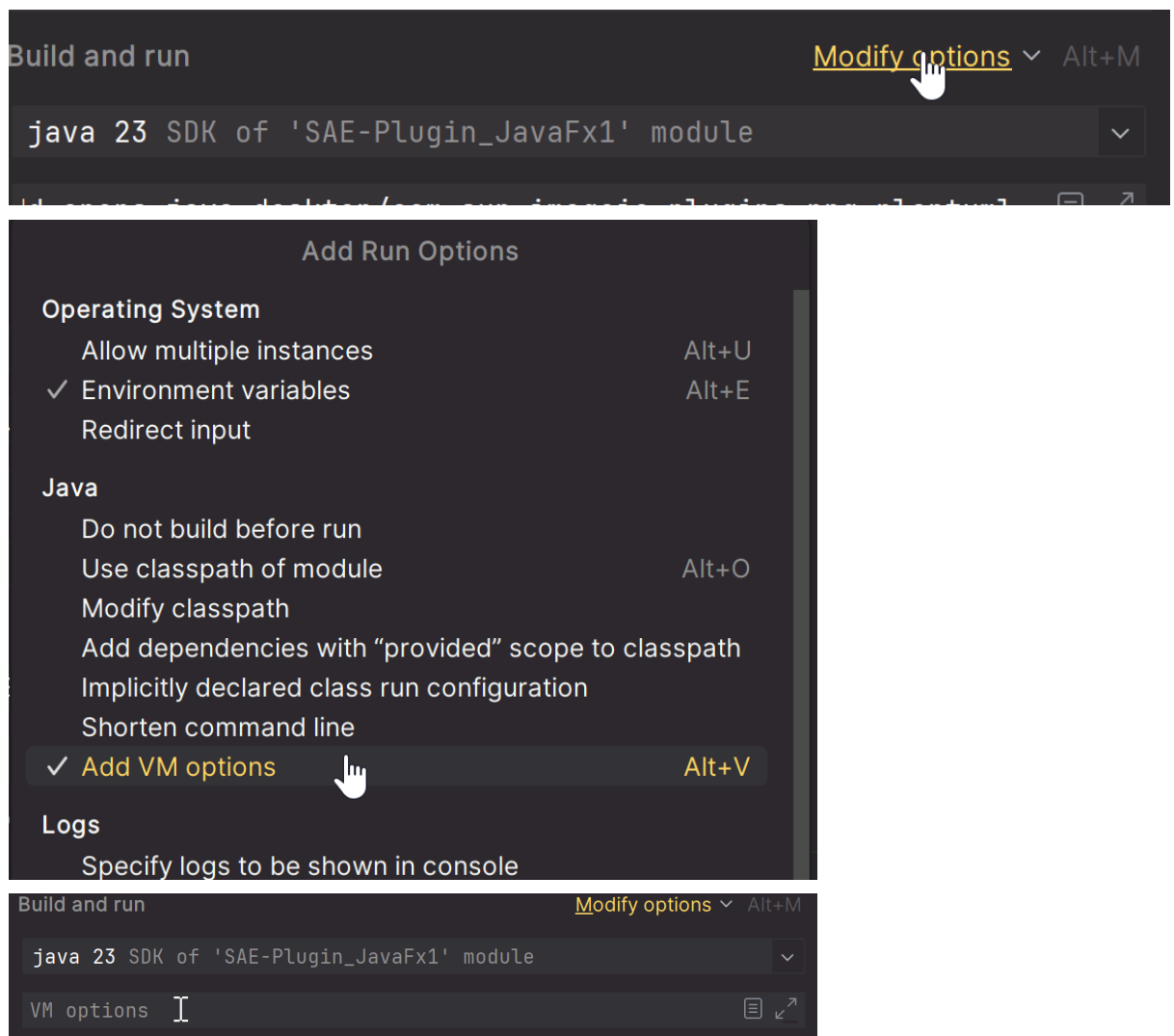
net.sourceforge.plantuml:plantuml:8059



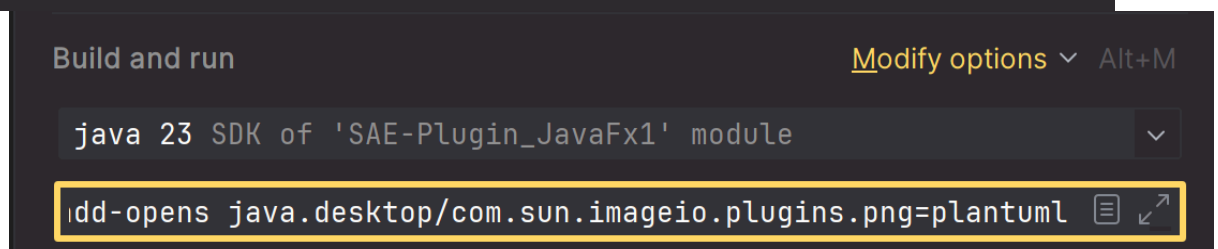


- Créer une configuration de lancement :





- Ajouter cette ligne en argument de la machine virtuelle java : `--add-opens java.desktop/com.sun.imageio.plugins.png=plantuml`



Notes :

Le plantUML est sauvegardé dans le fichier `Diag.puml`, placé à la racine du projet.
L'exportation en image est sauvegardée dans le fichier `capture.png`, placé à la racine du projet.
L'exportation en image plantUML est sauvegardée dans le fichier `Diag.png`.

Khodjaoui Elias, Fontaine Leo, Meyer Quentin, Reignier Eloi

La création du code source du diagramme se fait dans un dossier nommé squelette présent à la racine du projet. Les packages sont respectés dans ce dossier et les classes ajoutées manuellement ne sont dans aucun package.