TP 1

GL52

stephane.galland@utbm.fr

Introduction à Eclipse et Implantation d’une pile

00 | Objectifs

Les objectifs de cette session de TP sont de vous initier à l'outil Eclipse qui est la référence d'environnement de développement pour GL52, et de créer un structure de pile en appliquant les principes d'implantation issus du génie logiciel.

01 | Exercices

01.1 Initiation à Eclipse

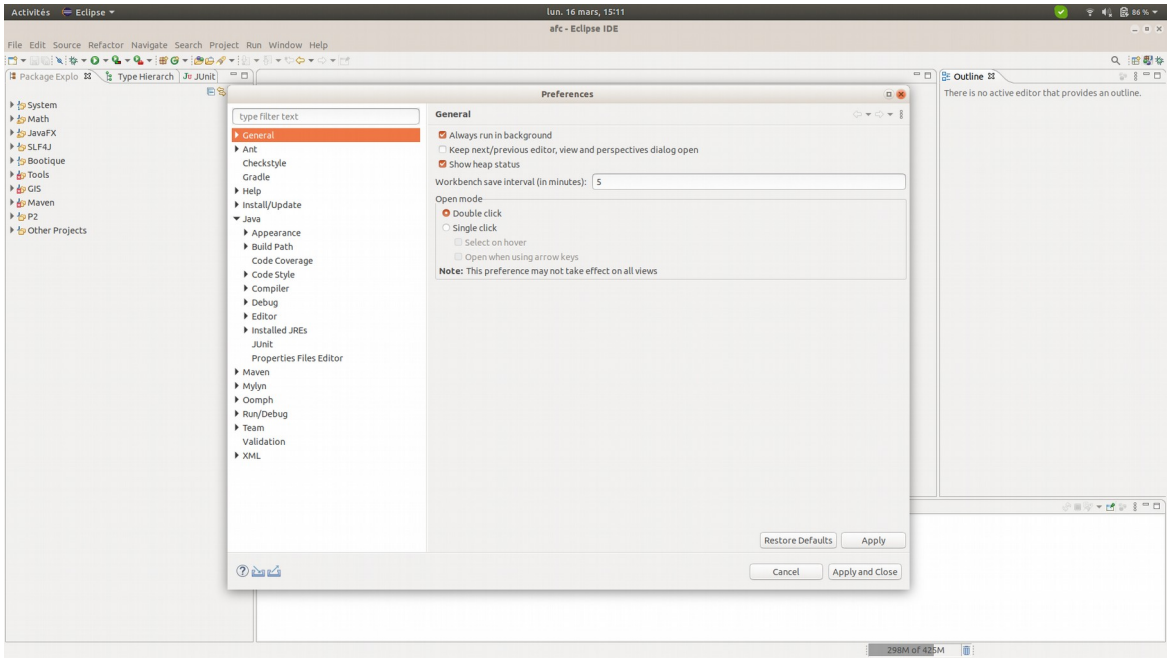
Travail à faire:

- Installer Java JDK 8 ou supérieur sur votre ordinateur
- Installer Eclipse 2019-12
- Lancer Eclipse
- Suivre le tutoriel : https://perso.telecom-paristech.fr/bellot/CoursJava/tps/tp1.html

01.2 Configuration d'Eclipse pour GL52

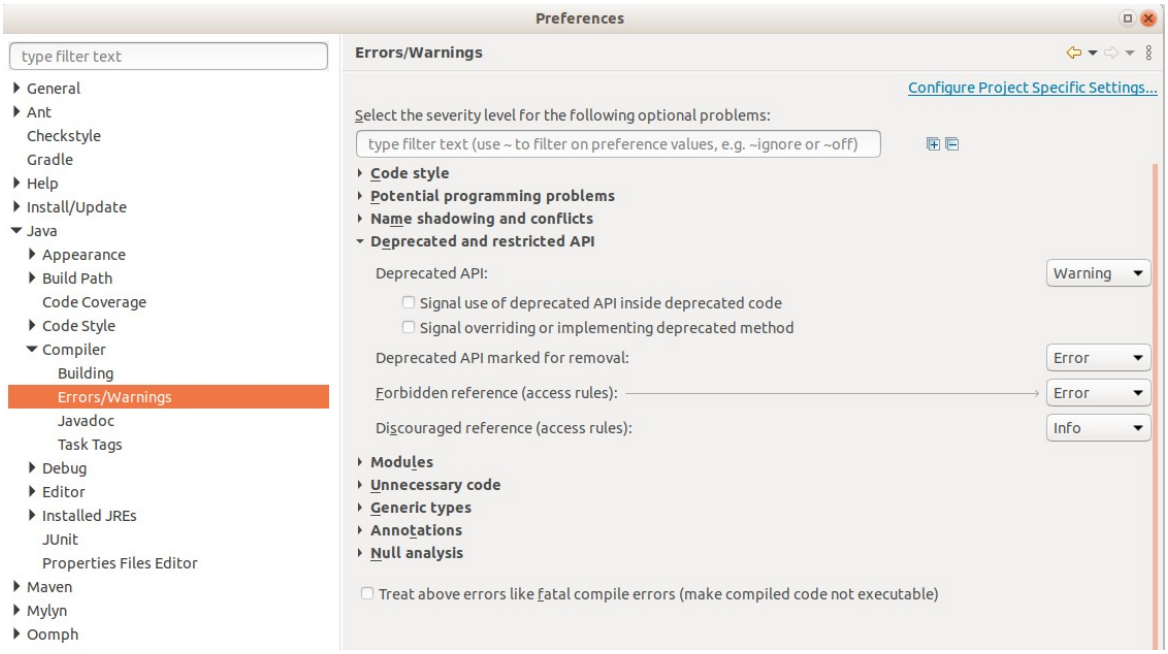
L'objectif est de configurer votre environnement Eclipse afin qu'il vous indique un maximum de messages d'anomalie, qui correspondent à des bonnes pratiques en génie logiciel :

- Lancer Eclipse
- Ouvrir la boite de dialogue des préférences : Menu « Windows » / « Préférences »



- Ouvrir la section « Java » / « Compiler » / « Errors/Warnings »

- Dans la partie droite de la fenêtre s'affiche la liste des erreurs reconnues par le compilateur Java et leur niveau d'urgence (error, warning, ignore).



- Ouvrir chaque partie (e.g. « Code Style ») et placer chaque erreur sur « Error » ou « Warnings ». Il ne doit plus y avoir de « Ignore ».

- Ouvrir la section « Java » / « Compiler » / « Javadoc », et réaliser la même tâche de configuration de précédemment.

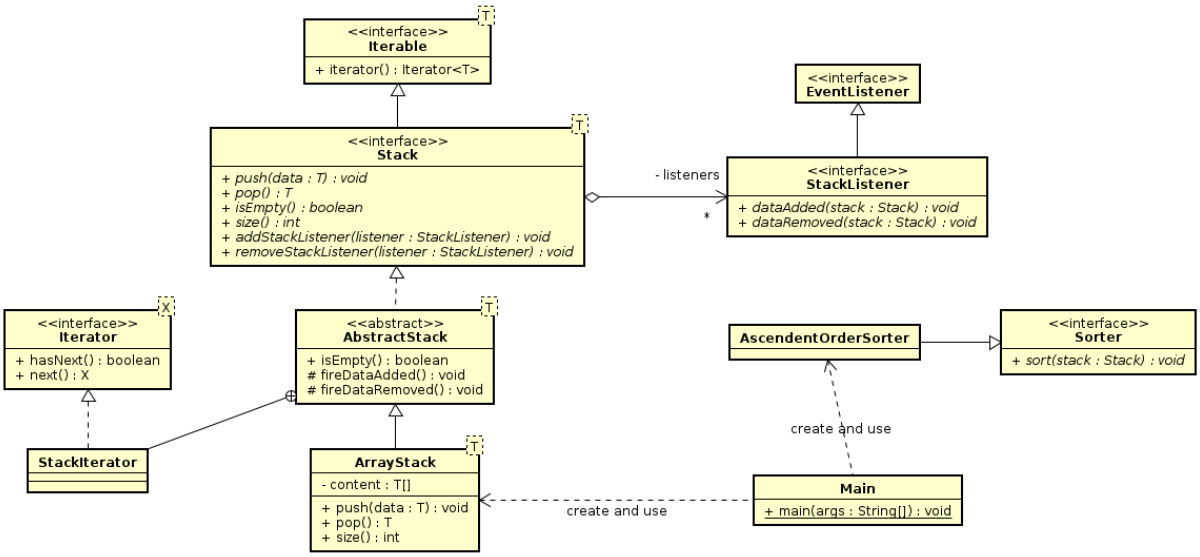
- Enregistrer la configuration

01.2 Réalisation d'une implantation de pile

Une pile est une structure de données permettant d'empiler et de dépiler des données.

Une pile peut contenir des données d'un type spécifié en type générique.

Le diagramme de classes UML ci-dessous vous donne la structure et les relations entre les types à implanter à la fin du TP.



Les interfaces « EventListener », « Iterable » et « Iterator » sont déjà fournis par l'API de Java. Les autres interfaces et classes doivent être implantées par vous.

Vous devrez respecter les bases des principes de la programmation inspirées des bonnes pratiques du génie logiciel, à savoir :

1. Utiliser des noms clairs et en anglais, car votre code doit pouvoir être lu par tout développeur, quelque soit son langage maternel.

2. Corriger le code pour qu'il n'y ait plus d'erreur ni de warning car chaque erreur ou warning indique une anomalie en relation avec les bonnes pratiques de programmation.

3. Ecrire une documentation claire et en anglais (vous aurez des warnings pour vous aider).

4. Appliquer le plus possible les design patterns : https://fr.wikipedia.org/wiki/Patron_de_conception

Travail à faire:

A) Structure de pile

- Écrire l'interface Stack représentant une pile. ATTENTION : il est interdit d'utiliser l'interface Stack déjà proposé dans l'API de Java.

- Écrire la classe AbstractStack représentant une implantation commune entre toutes les différentes implantations de piles. Pourquoi est-elle abstraite ?

- Écrire la classe ArrayStack qui implante les principes d'une pile en utilisant un tableau pour stocker les éléments dans la pile.

B) Design Pattern : Observateur

- Écrire l'interface « StackListener » représentant un observateur sur une pile. Les deux fonctions déclarées correspondent aux deux événements observables dans une pile.

- Modifier les types « AbstractStack » et « ArrayStack » pour stocker la liste des observateurs et les notifier lorsque la pile est modifiée.

C) Design Pattern : Iterateur

- Ecrire l'iterateur sur la pile « StackIterator ». Un iterateur à pour rôle de retourner en séquence la liste des données dans la pile, sans modifier le contenu de la pile (voir la documentation de « Iterator »).

- Modifer l'interface « Stack » et la classe « AbstractStack » pour intégrer la fonction d'itération sur la pile.

C) Travail optionnel : Outil de tri d'une pile

- Ecrire l'interface « Sorter » et la classe « AcendentOrderSorter » représentant respectivement un outils de tri des données dans une pile, et une implantation triant par ordre croissant.

D) Programme de test

- Ecrire une function « main » réalisant les actions suivantes :

(a) création d'une pile d'entiers

(b) ajout d'un nombre aléatoire d'entiers aléatoires dans la piles

(c) Afficher le contenu de la pile

(d) Trier la pile

(e) Afficher le contenu de la pile « triée »

A LA FIN DU TP, ENVOYEZ UN FICHIER ZIP CONTENANT LE CODE JAVA QUE VOUS AVEZ PRODUIT DURANT LE TP EN UTILISANT LA PLATE-FORME MOODLE.UTBM.FR