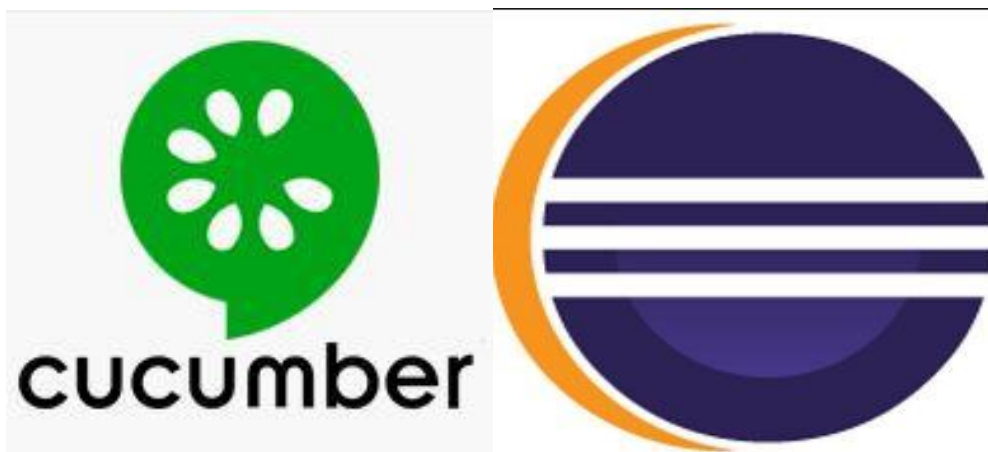




Méthodes Agiles - TP4

Scénario d'intégration agile



Binôme :

Khellaf DJEBBAR

Massinissa BOULKARIA

1) Rédiger les User stories :

Une user story (un scénario utilisateur), donc pour l'écrire il faut se mettre à la place de l'utilisateur et décrire ce que l'utilisateur attend de l'application, qui va permettre de guider le développeur dans l'atteinte du besoin clients.

Et pour le faire il faut mettre en place des scénarios suivis de l'user story :

User story – Scénario réservation :

En tant que vendeur de billets

Je veux gérer la vente des billets de chaque match disputé à domicile, avec la date du match, les deux clubs qui se confrontent, le stade, ainsi le numéro de la place du billet.

Scénario Outline : la réservation d'un billet comprenant le numéro de la place

Given : la création d'un billet comprenant le numéro de la place, la date du match, et les deux équipes confrontées.

When : L'acheteur valide.

Then : rendre la place indisponible.

Exemples :

La date	Club1 VS Club2	Stade	Num de place
01/08/2020	PSG VS OM	Paris saint Germain	200
10/07/2020	PSG VS OL	Groupama Staduim	104

Scenario Outline : Impossibilité de réserver une place deux fois.

Given : la création d'un billet comprenant le numéro de la place, la date du match, et les deux équipes confrontées.

When : L'acheteur valide.

Then : Le système refuse avec le message d'erreur <cette place est indisponible>

Exemple :

La date	Club1 VS Club2	Stade	Num de place
01/08/2020	PSG VS OM	Paris saint Germain	200
01/08/2020	PSG VS OM	Paris saint Germain	200 <place indisponible>

User story – Scénario annulation :

En tant que vendeur de billets

Je veux gérer la vente des billets de chaque match disputé à domicile, avec la date du match, les deux clubs affrontés, le stade, ainsi le numéro de la place du billet.

Scénario Outline : Annuler un billet déjà réservé.

Given : la suppression d'un billet comprenant le numéro de la place, la date du match, et les deux équipes affrontées.

When : L'acheteur valide.

Then : rendre la place disponible.

Exemple :

Annuler la place 200

La date	Club1 VS Club2	Stade	Num de place
01/08/2020	PSG VS OM	Paris saint Germain	200
01/08/2020	PSG VS OM	Paris saint Germain	200 La place 200 est disponible

Scenario Outline : Impossibilité d'annuler une place disponible.

Given : la suppression d'un billet comprenant le numéro de la place, la date du match, et les deux équipes affrontées.

When : L'acheteur valide.

Then : Le système refuse avec le message d'erreur <cette place est disponible>

Exemple :

La place 200 est disponible :

La date	Club1 VS Club2	Stade	Num de place
01/08/2020	PSG VS OM	Paris saint Germain	200 <cette place est disponible>

User story – Scénario modification :

En tant que vendeur de billets

Je veux gérer la vente des billets de chaque match disputé à domicile, avec la date du match, les deux clubs affrontés, le stade, ainsi le numéro de la place du billet.

Scénario Outline : Modifié un billet déjà réservé.

Given : la modification d'un billet comprenant le numéro de la place, la date du match, et les deux équipes affrontées.

When : L'acheteur valide.

Then : rendre l'ancienne place disponible, et la nouvelle indisponible.

Exemple :

Modifié la place 200 avec la place 201 :

La date	Club1 VS Club2	Stade	Num de place
01/08/2020	PSG VS OM	Paris saint Germain	200 disponible
01/08/2020	PSG VS OM	Paris saint Germain	201 indisponible

Scenario Outline : Impossibilité de modifier une place avec une place indisponible.

Given : la modification d'un billet comprenant le numéro de la place, la date du match, et les deux équipes affrontées.

When : L'acheteur valide.

Then : Le système refuse avec le message d'erreur <cette place est indisponible>

Exemple :

Modifié la place 200 avec la place 201 qui est indisponible :

La date	Club1 VS Club2	Stade	Num de place
01/08/2020	PSG VS OM	Paris saint Germain	200
01/08/2020	PSG VS OM	Paris saint Germain	201 <cette place est indisponible>

2) Planification des Itérations- Sprint Planning (Quoi/Quand) :

Le sprint agile comme étant le cœur des méthodes Agile. Cette qualification lui correspond plutôt bien, puisque tous les développements incrémentaux menant petit à petit au produit final du projet sont réalisés au sein des sprints.

Dans notre cas donc nous allons planifier nos itérations par périmètre de développement, affilié à une date limite, où quand la conclusion d'un sprint arrive à sa fin un autre sera déclenché, ainsi de suite jusqu'à ce qu'on obtienne un résultat final qui répond parfaitement au besoin client.

Quand ? : À la première semaine :

Quoi : l'équipe développement doit mettre en place les user stories, implémenter la Demo pour ensuite faire des tests et s'assurer de la justesse de l'algorithme implémenter.

Quand ? : à la deuxième semaine :

Quoi : l'équipe développement doit s'assurer de l'implémentation intégral du code qui répond au besoin du client, ainsi l'assurance des tests unitaires et fonctionnels, et la mise en place de l'exécution et testes automatiques.

Quand ? : au début de la troisième semaine :

Quoi : signalement et correction des anomalies.

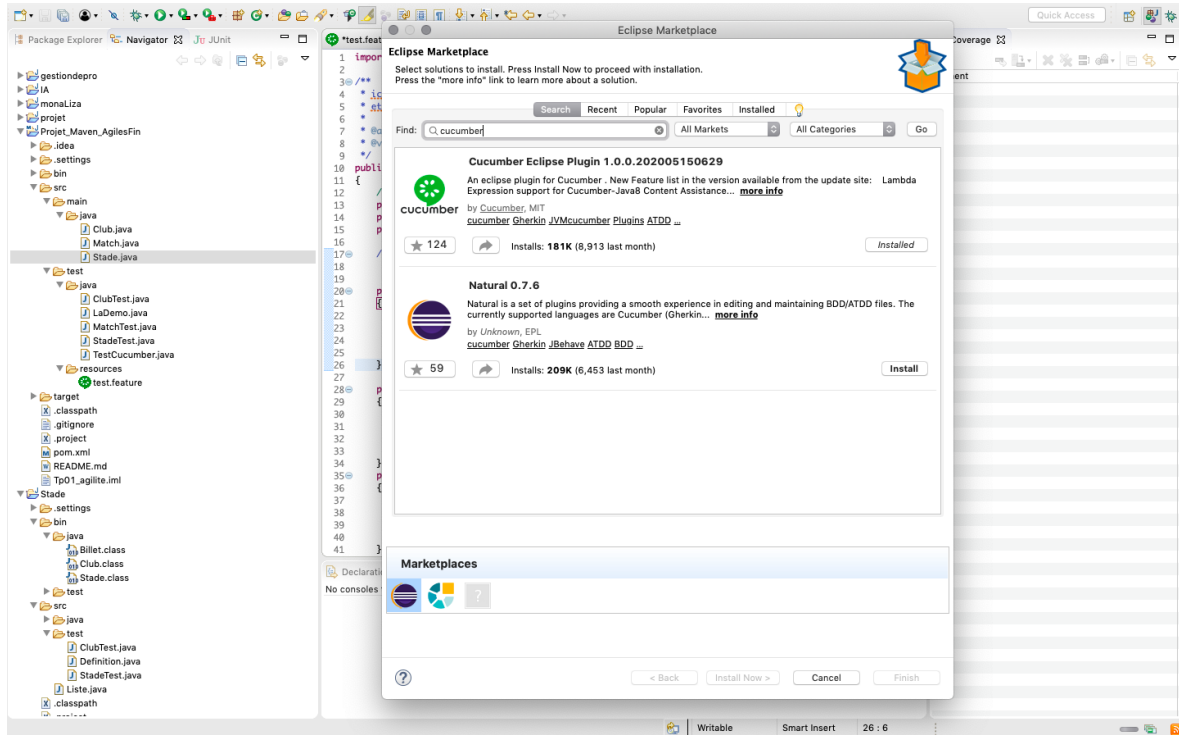
Quand ? : à la fin de la 3 ème semaine :

Quoi : Livraison du produit.

3) implémentation du code :

- Installation du plugin cucumber sur eclipse :

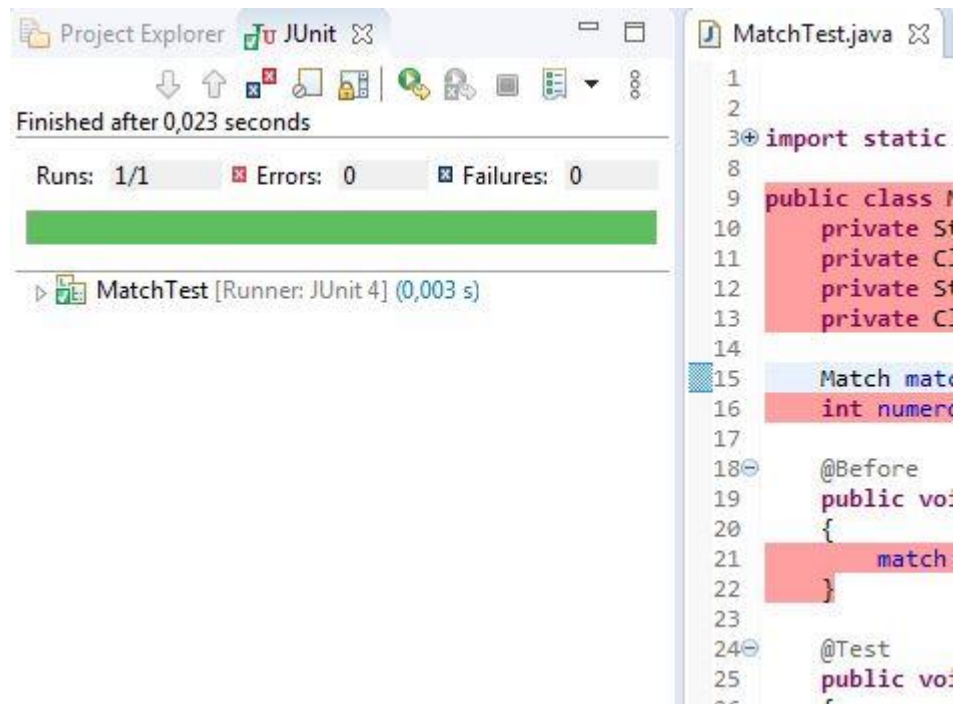
Pour faire les tests d'acceptance nous allons avoir besoin d'installer un outil logiciel comme cumcumber sur Eclipse, il suffit de le télécharger sur la Mareketplace.



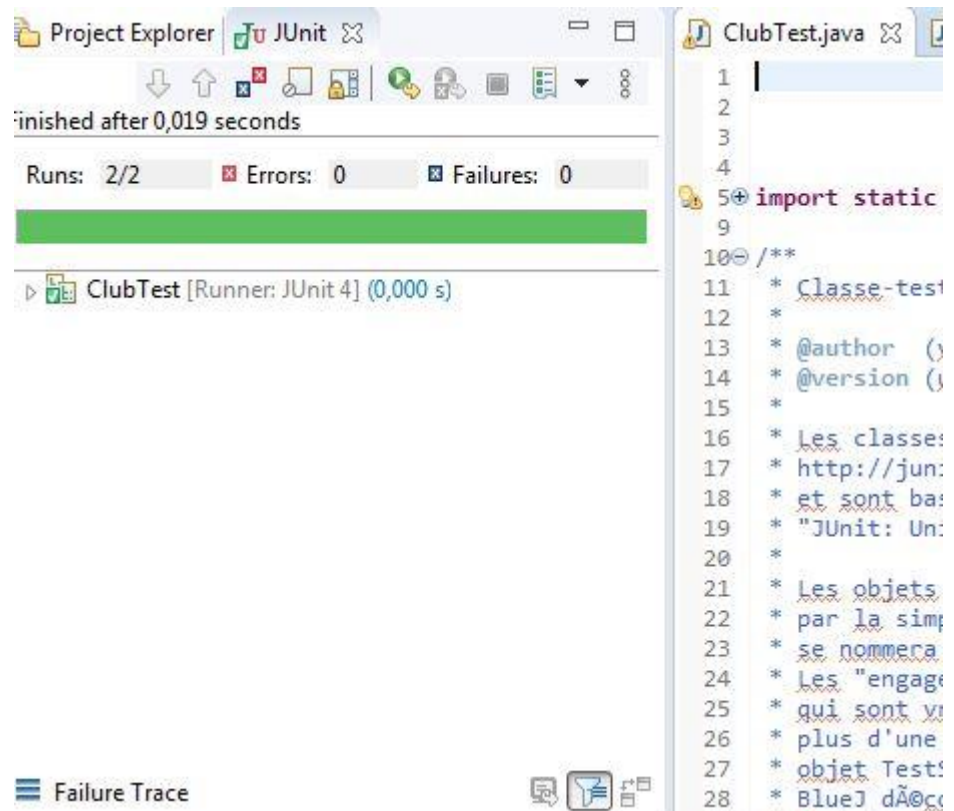
- Les tests unitaires :

Nous avons réalisé des tests sur les différentes classes, pour s'assurer du bon fonctionnement du code.

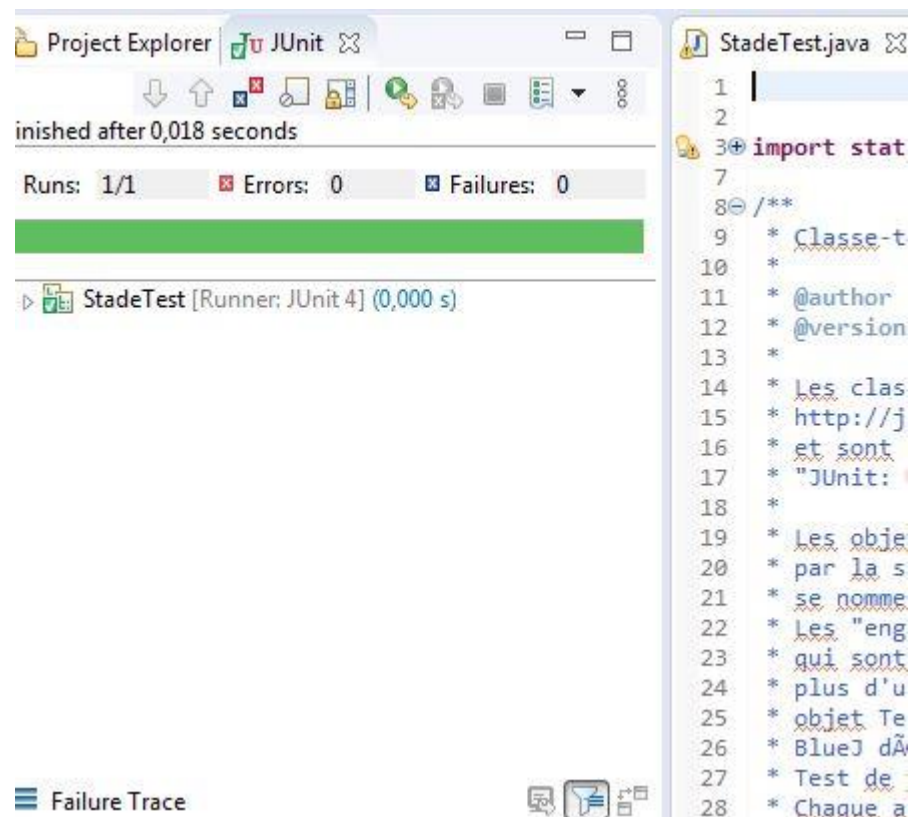
Classe "Match" :



Classe "Club" :

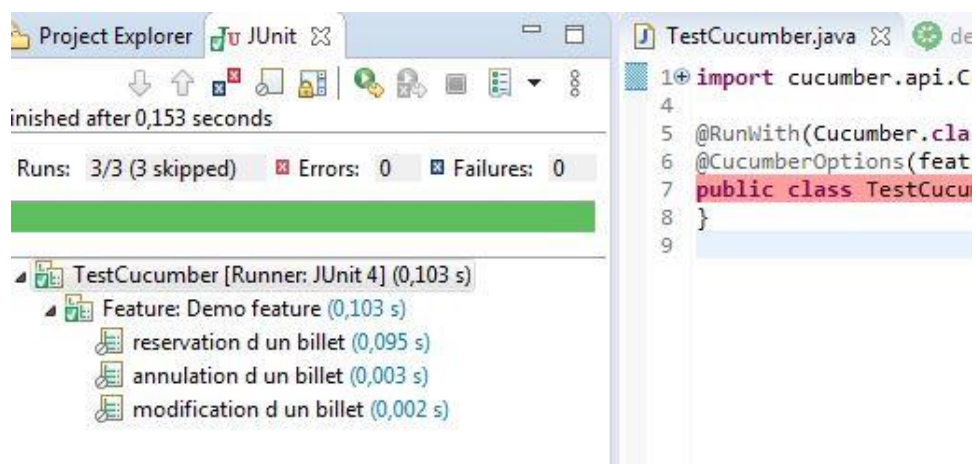


Classe "Stade" :



- Les tests fonctionnels :

Pour lancer le test on doit exécuter la classe « TestCucumber » en utilisant le framework de test jUnit.



4) Archivage des implémentations :

Pour archiver les implémentations on peut le faire de deux manières différentes :

1 : en ligne de commande : mvn clean install

mvn compile

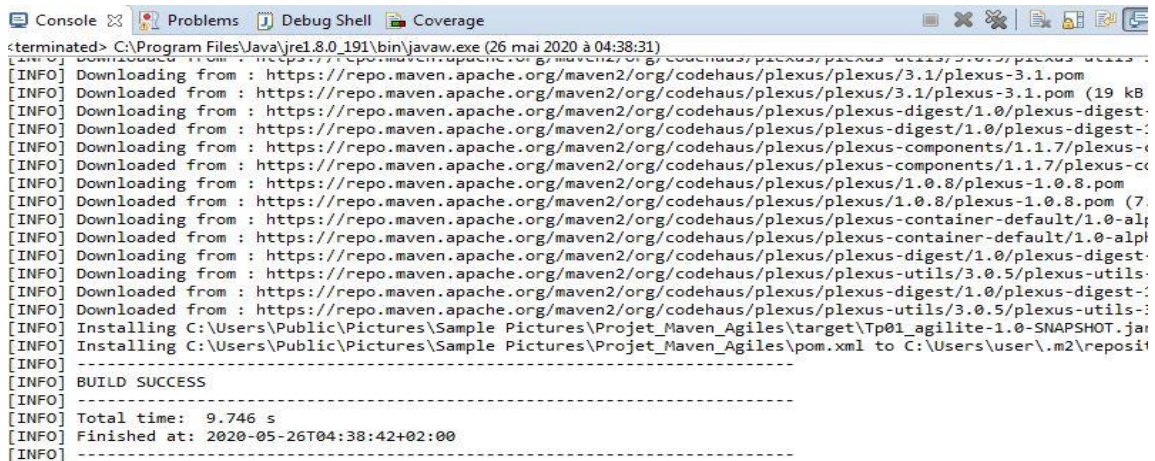
Mvn package

java -cp target/nom-du-jar nomduprojet

2 : depuis Eclipse ou on peut aussi le faire de deux manières :

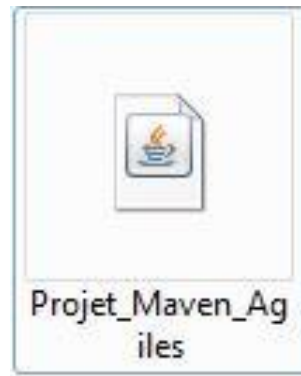
A) clic droit sur le projet, ensuite Export -> Java -> JAR File.

B) clic droit sur le projet, Run As -> Maven Install.



```
<terminated> C:\Program Files\Java\jre1.8.0_191\bin\javaw.exe (26 mai 2020 à 04:38:31)
[INFO] Downloading from : https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/3.0.9/plexus-utils-3.0.9.pom
[INFO] Downloaded from : https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus/3.1/plexus-3.1.pom (19 kB)
[INFO] Downloading from : https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-digest/1.0/plexus-digest-1.0.pom
[INFO] Downloaded from : https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-digest/1.0/plexus-digest-1.0.pom
[INFO] Downloading from : https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-components/1.1.7/plexus-components-1.1.7.pom
[INFO] Downloaded from : https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-components/1.1.7/plexus-components-1.1.7.pom
[INFO] Downloading from : https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus/1.0.8/plexus-1.0.8.pom
[INFO] Downloaded from : https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus/1.0.8/plexus-1.0.8.pom (7.7 kB)
[INFO] Downloading from : https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-container-default/1.0-alpha-9/plexus-container-default-1.0-alpha-9.pom
[INFO] Downloaded from : https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-container-default/1.0-alpha-9/plexus-container-default-1.0-alpha-9.pom
[INFO] Downloading from : https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-digest/1.0/plexus-digest-1.0.pom
[INFO] Downloaded from : https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-digest/1.0/plexus-digest-1.0.pom
[INFO] Downloading from : https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/3.0.5/plexus-utils-3.0.5.pom
[INFO] Downloaded from : https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/3.0.5/plexus-utils-3.0.5.pom
[INFO] Installing C:\Users\Public\Pictures\Sample Pictures\Projet_Maven_Agiles\target\Tp01_agilite-1.0-SNAPSHOT.jar to C:\Users\user\.m2\repository\org\codehaus\plexus\plexus-utils\3.0.5\plexus-utils-3.0.5.jar
[INFO] BUILD SUCCESS
[INFO] Total time: 9.746 s
[INFO] Finished at: 2020-05-26T04:38:42+02:00
[INFO]
```

Le résultat du JAR se trouvera dans le dossier sélectionné :



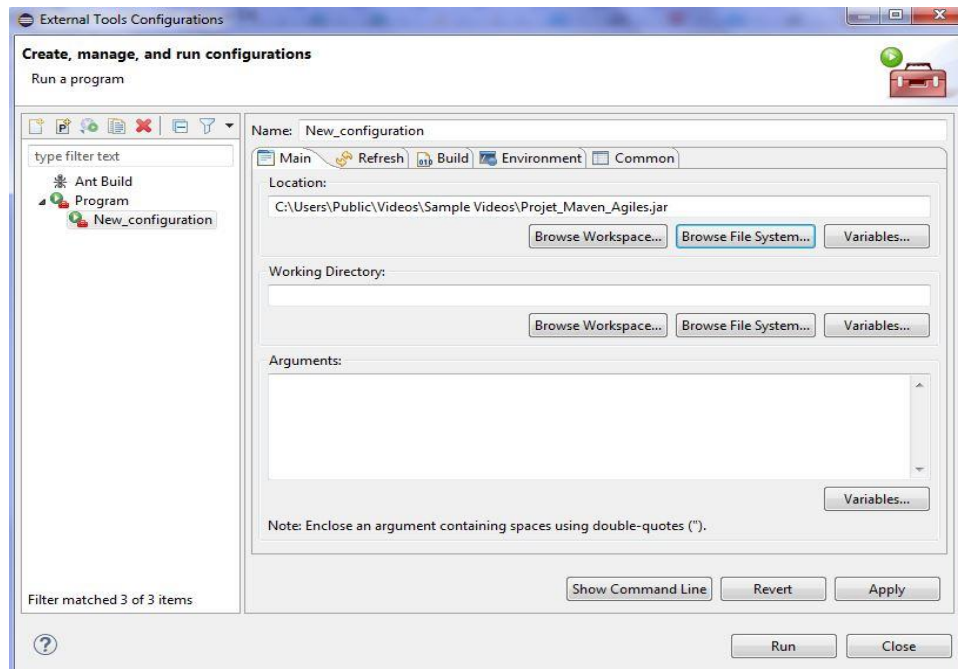
5) exécuter et tester automatiquement :

Pour le faire nous disposons de deux manières possibles :

1- ligne de commande : `java -jar nom-du-jar.jar`

2- sur Eclipse : Run > External Tools > External tools Configurations.

En selectionnant le JAR :



6) Signaler et corriger les anomalies et suivre l'avancement :

Nos anomalies initiales qui nous ont pris beaucoup de temps à corriger étaient à la construction de la classe SetDefinition, ou nous avons rencontré un problème java, ou les erreurs étaient du type conversion d'un entier à un Integer, qui nous a pris du temps pour effectuer nos tests fonctionnels.

Nous avons réussi à corriger cela en changeant le type de notre variable en Integer dans notre code initial.