TP2 et TP3
« Astéroïde 325»

Ce TP vous emmène en voyage sur l'astéroïde 325. Ce petit bout de cailloux fait partie d'un des décors de l'histoire du *Petit Prince* d'Antoine de Saint-Exupéry. Il possède un seul habitant, un roi pas très marrant :

> *- Ah ! Voilà un sujet, s'écria le roi quand il aperçut le Petit Prince.*
> *Et le Petit Prince se demanda :*
> *- Comment peut-il me connaître puisqu'il ne m'a encore jamais vu !*
> *Il ne savait pas que, pour les rois, le monde est très simplifié. Tous les hommes sont des sujets.*

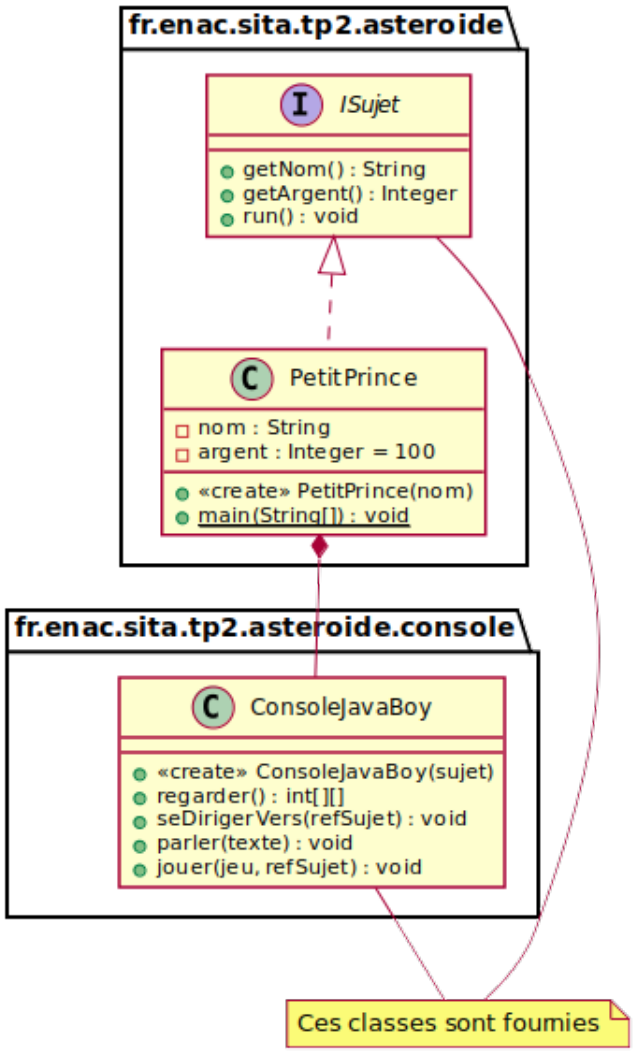
Nous vous proposons pour quelques heures de changer un peu l'histoire, en particulier de peupler cet astéroïde de petits princes virtuels, et de les faire interagir entre eux et avec le roi par l'intermédiaire d'un moyen de communication moderne : la console « JavaBoy » !

1ère étape : Connecter tout le monde

A. de Saint-Exupéry l'a bien compris, on peut simplifier ce monde et considérer que toute personne marchant sur l'astéroïde 325 est un sujet. Cela permet au roi (qui est en fait un programme serveur partagé pour toute la salle de TP) d'interagir avec un sujet quelle que soit son implémentation (vous aurez tous des Petits Princes différents !), tant qu'il possède les méthodes spécifiées dans l'interface `Isujet`.

Chaque sujet arrivant sur l'astéroïde 325, peut utiliser les services de la console JavaBoy prêtée par le roi.

Le diagramme de classes UML ci-contre montre comment s'articulent les classes de votre application.



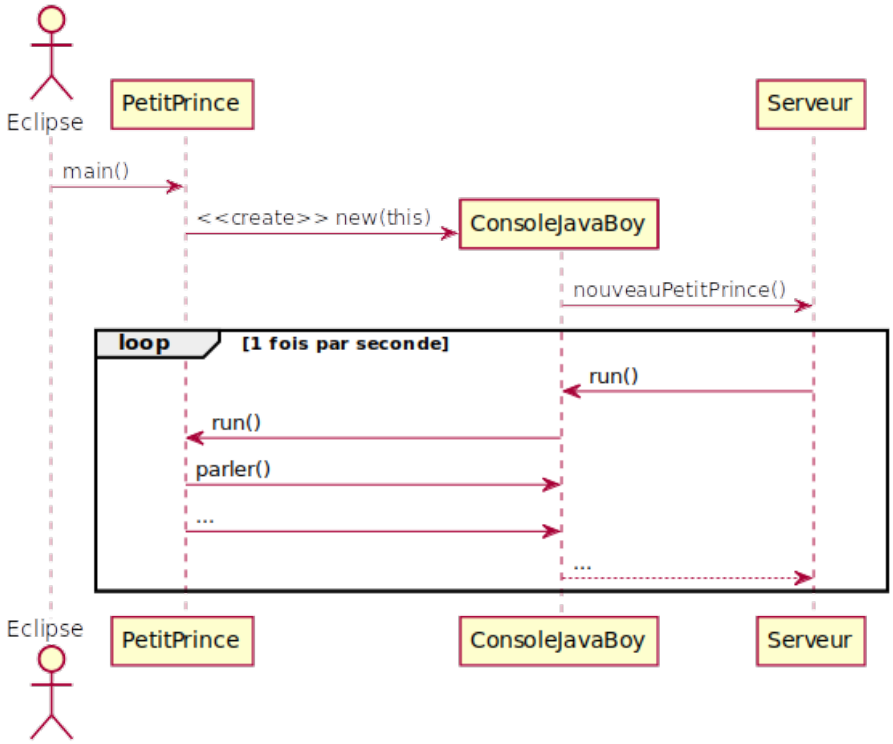
⇒ Créez un nouveau projet Java

- *Vous ferez particulièrement attention à respecter les noms de paquetages qui sont indiqués dans le sujet.*
- Téléchargez depuis e-campus l'archive « asteroide325.zip ». Décompressez la et collez en le contenu à la racine de votre projet.
- Cela fait apparaître un répertoire « doc » qui contient de la documentation Javadoc, ainsi qu'un squelette de code (paquetage et interface ISujet) dans le répertoire « src », un répertoire « lib » qui contient une archive jar, dans laquelle se trouve plusieurs classes Java compilées (dont la classe `ConsoleJavaBoy`) et un fichier de configuration « config.ini ».
- Pour permettre à NetBeans de « voir » les classes de l'archive jar, vous pouvez faire la manipulation suivante :
  - Faites un clic droit sur le nom du projet,
  - sélectionnez l'item de menu nommé « Properties »,

- sélectionnez la catégorie « Librairies »,
- appuyez sur le bouton « + » en regard de l'inscription « Classpath »
- choisissez l'item de menu « *Add JAR/Folder* »,
- sélectionnez la librairie « asteroide325.jar » du dossier « lib ».

- Pour permettre la connexion au serveur ouvrez le fichier « config.ini » depuis un éditeur de texte et modifiez l'adresse IP du serveur (votre enseignant vous donnera l'adresse de la machine où il aura lancé le serveur).

Le fonctionnement de l'interaction entre le serveur (roi) et votre client (petit prince) est le suivant : une fois que vous avez connecté votre console JavaBoy au serveur, celui-ci appelle une fois par seconde la méthode `run()` de votre sujet. Cela vous permet de faire tout ce que vous souhaitez à condition que vous terminiez l'exécution de la méthode run() en moins d'une seconde (sinon vous serez déconnecté !).

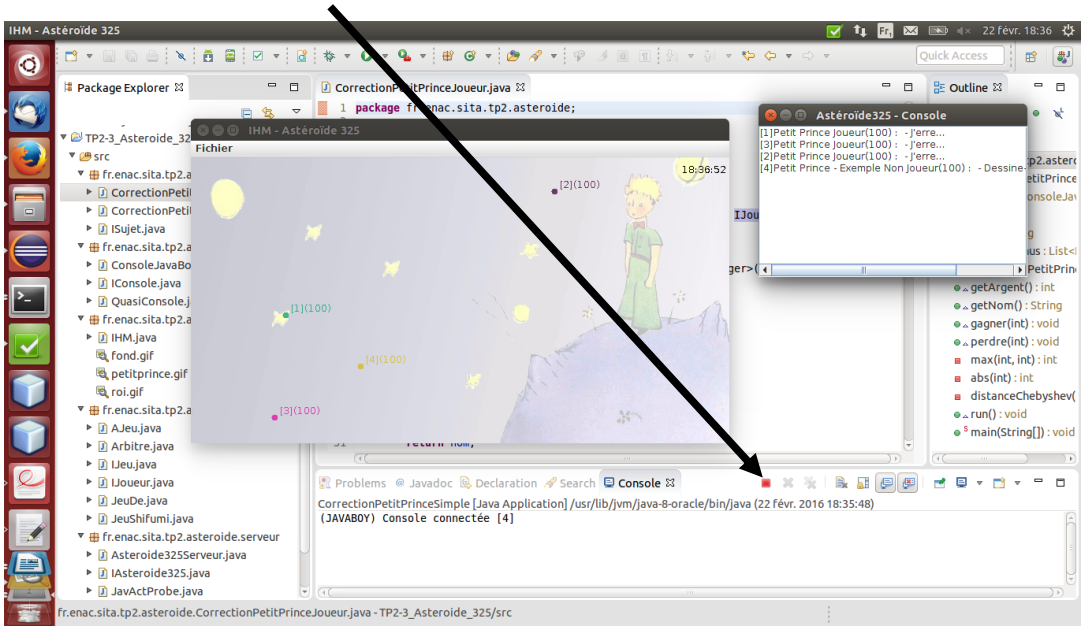


=> Codez votre Petit Prince

- Dans la méthode `run()`, dans un premier temps, vous pourrez faire s'exprimer votre sujet en utilisant la méthode `parler()` de la console, puis le faire se déplacer aléatoirement en utilisant la méthode `seDirigerVers(0)`.
- Ajoutez un point d'entrée (méthode `main`), qui se contente de construire une instance de votre personnage.
- Pour visualiser l'astéroïde 325 et tous ses sujets, vous avez à votre disposition une interface graphique empaquetée dans un jar exécutable (asteroide325IHM.jar). Lancez-la par exemple dans une console avec la commande : `java -jar asteroide325IHM.jar`
- Depuis votre IDE, lancez une exécution de votre sujet. Vous pouvez observer sur la console un message vous donnant un identifiant de console ; ce numéro vous permet de repérer votre sujet par la suite.

Remarque : en cas de défaillance de votre programme (temps de réponse trop long, inactivité, etc.) et dans certains cas d'erreur, le roi (le serveur) vous renverra de son astéroïde, en vous donnant la raison.

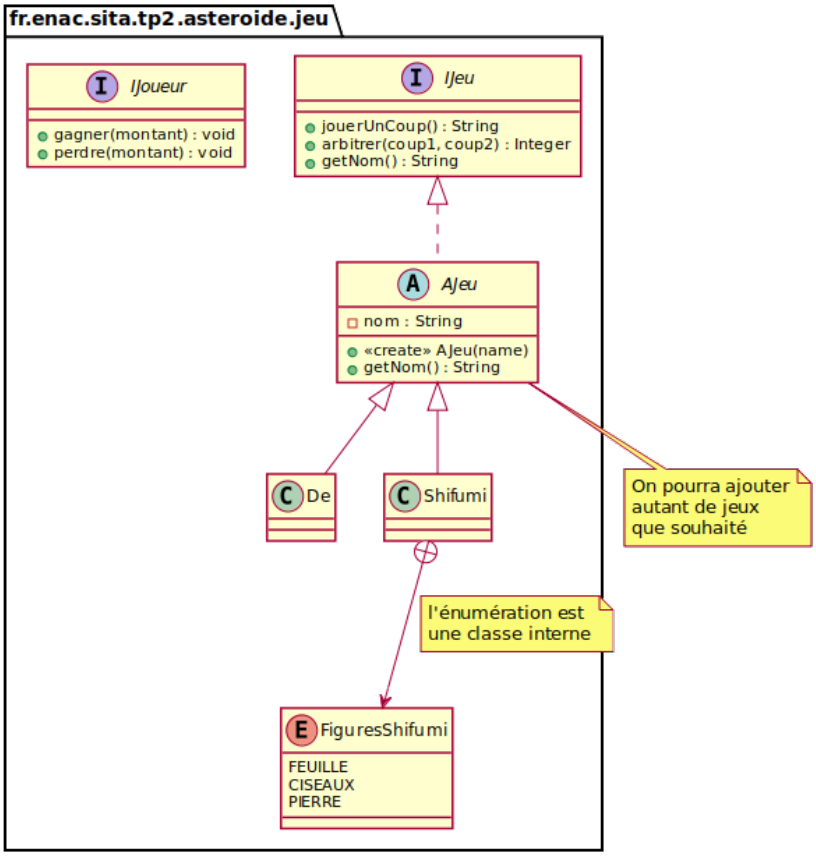
Vous pouvez également faire quitter l'astéroïde à votre Petit Prince en terminant volontairement l'exécution de votre sujet (clic sur le bouton carré rouge par exemple, cf. illustration ci-dessous).



2ème étape : le démon du jeu

Votre Petit Prince est maintenant capable de se déplacer sur l'astéroïde 325, mais comme marcher n'est pas son fort, il se dit qu'il programmerait bien sa console pour jouer avec les autres sujets du royaume.

Le Petit Prince imagine des jeux simples comme « pile ou face », des « jeux de dé », ou encore jouer avec les mains par exemple à « Pierre-Feuille-Ciseau » (jeu du Shifumi). En y réfléchissant, il se dit que tous ces jeux ont des points communs et pour factoriser les codes, il dessine *vite fait* dans le sable le diagramme de classes suivant.



=> Rappelez l'intérêt d'une classe abstraite, d'une interface et de leur complémentarité.

=> Créez un paquetage nommé précisément `fr.enac.sita.tp2.asteroide.jeu`, dans lequel vous placerez les classes correspondant à cette partie du sujet.

- Écrivez l'interface `IJeu` et la classe abstraite `AJeu` comme spécifié ci-dessus.
- Codez la classe `De` : pour jouer un tour, vous retournerez une chaîne de caractères contenant un chiffre tiré au hasard entre 1 et 6. Pour la méthode `arbitrer()`, le gagnant peut être (à votre convenance) le plus grand ou le plus petit chiffre passé en paramètre. Ainsi vous retournerez -1 si `coup1` est le gagnant, +1 si c'est `coup2`, et 0 en cas d'égalité.
- Pour ``Pierre-Feuille-Ciseau'' (qui s'appelle aussi le jeu de Shifumi), vous devrez utiliser une énumération pour représenter les différentes figures possibles. Écrivez la classe `Shifumi`. Pour jouer un tour, vous retournerez une chaîne de caractères correspondant à l'un des coups possible du Sifumi, tiré aléatoirement. Consultez la documentation Java en ligne pour trouver comment utiliser les énumérations.
- Ajoutez l'interface Joueur, puis utilisez-là pour exprimer que vos Petits Princes sont des joueurs (faites les modifications nécessaires).

3ème étape : Jouons ensemble

Maintenant que les jeux sont en place, vous allez pouvoir les intégrer dans le monde de l'Astéroïde 325.

Pour cela, vous devrez d'abord essayer de vous rapprocher d'autres sujets. Ensuite, lorsqu'ils seront distants d'une seule case, vous pourrez essayer de jouer avec eux (via la méthode `jouer(JeuSimple jeu, int refSujet)`). Le roi (le serveur) se fera un plaisir d'arbitrer la partie, et de déclarer éventuellement le gagnant en appelant les méthodes `gagner` et `perdre` de chacun.

> ***Attention*** *: vous ne pouvez jouer qu'avec des sujets qui n'ont jamais joué avec vous, et qui sont également des joueurs.*

Pour mesurer la distance qui vous sépare d'un autre sujet, vous pouvez ajouter une méthode privée.

Vous pouvez utiliser ce genre d'algorithme pour coder le comportement de votre sujet :

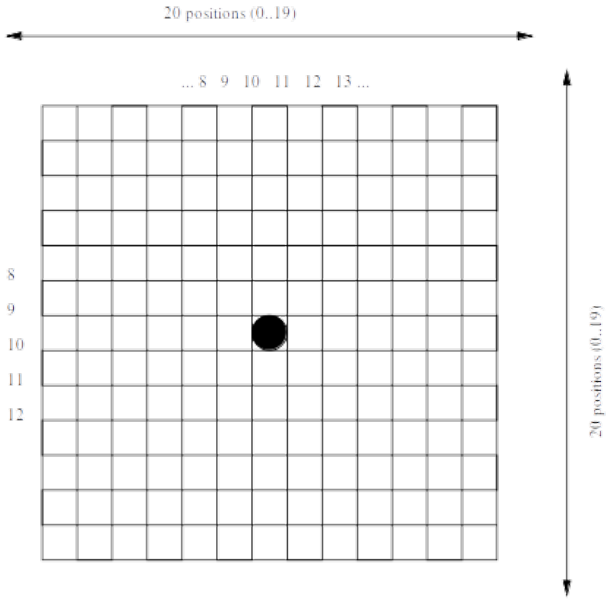
```
V = regarder
S = chercher le plus proche sujet dans V


SI (distance(s)<=1) ALORS
  jouer avec S a un jeu (au hasard parmi ceux implantes)
SINON
  SI (s est null) ALORS se diriger au hasard
  SINON se diriger vers S
  FINSI
FINSI
```

=> Modifiez le code de votre méthode `run`() pour poursuivre un sujet avec lequel vous n'avez jamais joué et qui se trouve dans votre champ de vision afin de lui proposer l'un de vos jeux quand vous serez assez proche.

=> Si vous avez fini en avance, ajoutez un jeu de votre choix, compatible avec le modèle proposé à l'étape 2

ANNEXE



Vision du Petit Prince :

La vision d'un Petit Prince est donnée par la console. Celle-ci retourne un tableau d'entiers (20x20) contenant soit 0 si aucun sujet n'occupe la case soit une référence identifiant un sujet. La position [10,10] correspond à votre sujet. Deux sujets ne peuvent jamais occuper une même case. Votre référence n'est pas renvoyée.

La classe `consoleJavaBoy` est documentée (Javadoc) dans le répertoire doc de votre projet.