
8PRO102 – Langages de programmation

Laboratoire #4

Session A-2017

Date de remise : 13 décembre 2017

Remise par courriel : eric_dallaire@uqac.ca

Sujet : Remise du laboratoire 4 : Nom Prénom

À remettre : Un dossier contenant tous les fichiers du projet (.java et .class) et le document contenant les captures d'écran

Aéroport

À l'aide du langage Java, programmer la solution proposée et décrite par le diagramme UML ci-dessous et les descriptions associées.

- Réutiliser le modèle de classes fourni pour vous guider dans votre création. Toutefois, vous pouvez en ajouter des nouvelles.
- Des exigences de bases sont fournies avec les classes et vous déterminez vous-même comment implémenter le fonctionnement proposé

IDE

Vous pouvez utiliser l'éditeur de votre choix pour réaliser ce projet

- Lien pour **Eclipse**
<http://www.eclipse.org/downloads/packages/eclipse-ide-java-developers/oxygen1a>
- Lien pour **IntelliJ**
<https://www.jetbrains.com/idea/download/>
- Lien pour **Netbeans**
<http://www.oracle.com/technetwork/articles/javase/jdk-netbeans-jsp-142931.html>
- Lien pour **JDK** (Si pas inclus dans l'IDE)
<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

Que doit faire le programme

CONTRÔLE EN CONSOLE (processus principal)

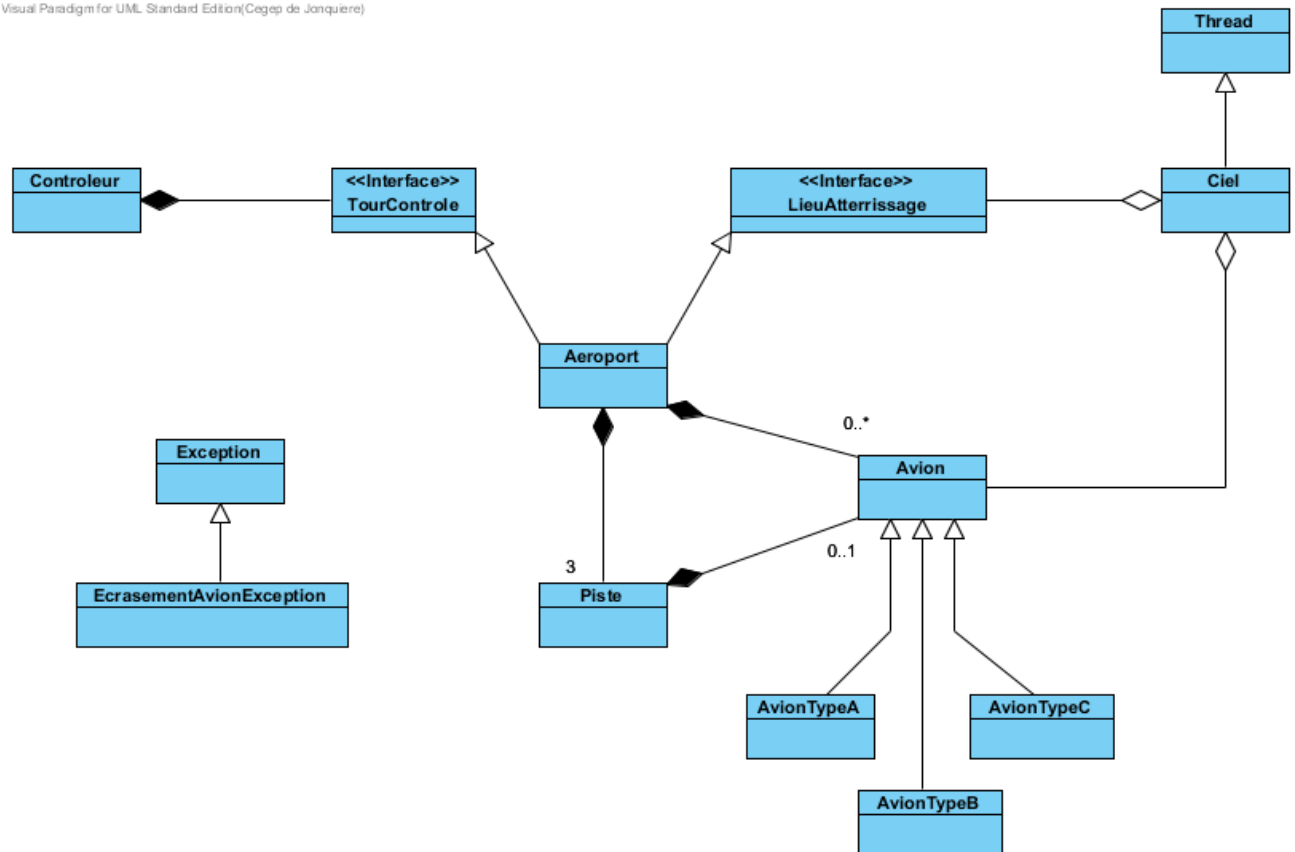
Le contrôleur peut réaliser 3 actions

1. Faire survoler les avions
 - Chaque avion dans le ciel perd 1 PT d'essence
 - Si un avion n'a plus d'essence, elle s'écrase!!!!
 - Chaque avion atterri sur une piste avance d'un TOUR
 - Si l'avion est là depuis 5 tours, la piste se libère (l'avion est « éliminé »)
2. Faire atterrir un avion
 - On doit choisir une piste et un numéro de série d'avion
 - Si une piste est déjà occupée, l'avion s'écrase
 - Autrement, l'avion arrête de survoler l'aéroport puis se pose sur la piste
 - Après l'atterrissage, on fait survoler les autres avions (voir point #1)
3. Quitter

CONTRÔLE AUTOMATIQUE (thread)

La méthode « run » du ciel exécute en boucle les opérations suivantes

1. Créer un nouvel avion
 - On instancie des avions des types « AvionTypeA », « AvionTypeB » et « AvionTypeC ».
 - L'utilisation de l'instanciation dynamique (avec la classe « Class ») donnera un petit bonus.
2. Ajouter cet avion au lieu d'atterrissage
3. Attendre (sleep) une durée de temps variable
4. Attention : Le ciel continue d'envoyer des avions indépendamment de votre vitesse de traitement en console!!!!



Les classes (quelques contraintes minimum)

Avion	<ul style="list-style-type: none"> • Classe abstraite qui possède 3 attributs : <ul style="list-style-type: none"> ○ La quantité d'essence ○ Le type d'avion ○ Le numéro de série • Il y a un seul constructeur et il reçoit les 3 paramètres
AvionTypeA AvionTypeB AvionTypeC	<ul style="list-style-type: none"> • Les noms des 3 avions peuvent varier (vous décidez) • Les 3 avions héritent de la classe abstraite Avion • Il n'y a qu'un seul constructeur et il ne reçoit aucun paramètre. Il aura pour rôle de fournir les bons paramètres au constructeur du parent.
EcrasementAvionException	<ul style="list-style-type: none"> • Cette classe hérite de la classe java.lang.Exception (API) • Surcharger la méthode « getMessage »
Ciel	<ul style="list-style-type: none"> • Cette classe hérite de la classe Thread (API) • Elle a un attribut de type « LieuAtterrissage » qui permet au ciel d'ajouter des avions. • N'oublier pas de surcharger la méthode run() et de « démarrer le ciel » dans l'aéroport (via le constructeur?)
Piste	<ul style="list-style-type: none"> • La piste possède 3 attributs <ul style="list-style-type: none"> ○ Un avion (la référence ou NULL) ○ Le nombre de tours depuis l'arrivée de l'avion ○ Le numéro de piste • Lorsqu'un avion est sur la piste, elle y reste 5 TOURS. • Après 5 tours, elle est éliminée (au hangar ;-)
LieuAtterrissage	<ul style="list-style-type: none"> • Cette interface ne possède qu'une seule fonction « survoler » • Avec cette fonction, le ciel pourra « ajouter » des avions à l'aéroport
TourControle	<ul style="list-style-type: none"> • Cette interface possède 3 fonctions principales <ul style="list-style-type: none"> ○ Afficher l'aéroport ○ Faire atterrir un avion ○ Faire survoler les avions (passer un tour)
Controleur	<ul style="list-style-type: none"> • Cette classe contient le « main » et un lien vers la tour de contrôle (en fait l'aéroport) • Ici, on peut effectuer les actions du système <ul style="list-style-type: none"> ○ Faire atterrir un avion ○ Passer un tour! (faire survoler les avions) ○ Afficher l'état de l'aéroport ○ Quitter
Aéroport	<ul style="list-style-type: none"> • La classe principale de l'application • L'aéroport contient 3 pistes et une tous les avions qui survolent dans le ciel