

IPST-CNAM
Intranet et Designs patterns
NSY 102
Vendredi 9 Avril 2021

Durée : **1 h 30**
Enseignants : LAFORGUE Jacques

1ère Session NSY 102

NOM :
PRENOM :
NUMERO D'INSCRIPTION :

CONSIGNES :

1/ Vous venez de télécharger le sujet d'examen de NSY 102 1^{ère} session 2020-2021.
Ce fichier est un fichier word. Il doit rester dans ce format ou si vous utilisez LibreOffice le format .odt est accepté.

Vous renommez, tout de suite, ce fichier avec votre nom et prénom suivant le format "NOM PRENOM.doc".

2/ Vous écrivez, tout de suite, ci-dessus, votre nom, prénom et numéro d'inscription.

3/ Vous répondez, en insérant vos textes et vos diagrammes en-dessous de chaque énoncé.

Vous ne modifiez pas les énoncés.

4/ Au bout de 1 heures et 30 minutes, vous déposez votre copie dans : **DEPOT DES COPIES.**

Attention : Après avoir remis votre copie, vous restez en ligne afin que je vous dise que la séance d'examen est terminée car il me faut vérifier que tout le monde ait bien remis sa copie.

En cas de panne de courant, pensez à enregistrer régulièrement votre travail (surtout si vous n'utilisez pas un ordinateur portable ou n'utilisez pas la batterie).

Si vous avez perdu votre connexion internet et que vous ne pouvez pas déposer votre copie alors la procédure à suivre est la suivante :

- vous m'envoyez un SMS au 06 88 73 25 65 afin de me prévenir de cette difficulté

- vous m'envoyez par SMS une photo de chaque page de votre copie (précisez votre nom dans le SMS) (pour preuve)

- dès que vous pourrez, vous déposerez votre copie comme prévu, et vous m'enverrez un mail pour me prévenir du dépôt.

Au total, ce document est constitué de 4 pages. Veuillez vérifier que vous avez bien 4 pages et que la dernière page se termine bien par la phrase « Fin du sujet ».

J'ai volontairement mis les pages, après la description du problème, devant contenir vos réponses, en mode paysage afin que les copiés/collés de vos diagrammes soient plus visibles.

PROBLEME [50 points] (1h30)

On veut réaliser le Système d'Information permettant de gérer la mise à jour des panneaux d'affichage des vols d'un grand aéroport.

Toutes les compagnies aériennes déposent les plans de vol (PLN) dans une base de données unique (BDPLN), au plus tard la veille de la date de départ de chacun des vols. (On ne traite pas les compagnies aériennes dans ce sujet).

Un poste IHM d'administration [COMPOSANT 3] de l'aéroport permet de configurer un serveur [COMPOSANT 1] afin qu'il récupère de la BDPLN, automatiquement, par exemple à 4h du matin, tous les plans de vols prévus pour la journée.

Les plans de vols récupérés sont stockés en mémoire du serveur. Le serveur est propre à l'aéroport.

Pendant la journée, à tout moment, un administrateur peut utiliser ce poste pour demander au serveur de mettre en standby, annuler, modifier l'heure de départ d'un vol ou modifier toute autre information du plan de vol. Il peut aussi créer, exceptionnellement, un nouveau plan de vol.

Le poste affiche tous les plans de vol et permet de faire toutes les commandes nécessaires sur un plan de vol.

Le rôle de ce serveur est aussi de gérer tous les panneaux d'affichage [COMPOSANT 2] répartis dans l'aéroport.

Il existe différents types de panneaux entre ceux destinés au public et ceux destinés à des techniciens (enregistrement, douane, bagagerie, ...). Ainsi, en fonction de son type, chaque panneau n'affiche pas les mêmes informations. (Pour simplifier l'écriture de vos diagrammes, prenez 2 types de panneaux possibles)

Pour simplifier à notre niveau, un panneau est vu comme un composant d'IHM graphique lourd qui s'abonne, dès son lancement, au serveur en lui indiquant sa nature.

En fonction de sa nature, le serveur devra lui notifier les informations qu'il attend.

Pour fixer les idées (inutile de mettre tous ces attributs dans votre diagramme, cela étant implicite), voici un exemple des informations contenues dans un plan de vol : aéroport de départ, aéroport d'arrivée, heure de départ, heure d'arrivée, identifiant du vol, le type d'avion, le nom de la compagnie, le nombre de passager prévus, la nature du frète si exceptionnel, le niveau global de risque potentiel pour la douane, le nombre de bagages, une demande de révision ou d'entretien pour l'avion, un texte libre d'information, ...

Dynamiquement, à chaque fois que le serveur met à jour une information d'un plan de vol, les panneaux d'affichage sont aussitôt notifiés de la modification.

Le serveur contient tout un ensemble de traitements algorithmiques (recherche, planification, filtre, tri, ...). Ces traitements utilisent directement les plans de vol mémorisés dans le serveur et peuvent donc aussi les modifier.

(Suite du sujet page suivante)

PARTIE 1 [15 points] :

Faites le diagramme de communication de ce Système d'Information [10 points].

Commentez votre schéma (rôles des composants, comportement dynamique général, échanges des informations, localisation des données) [5 points].

Nous rappelons que ce schéma doit permettre de connaître vos choix d'organisation des composants et des composants internes (sous-composants) de chacun des trois COMPOSANTS de ce SI, et vos choix d'appel entre les composants et sous-composants.

<Ecrire ici votre diagramme de communication>

<Ecrire ici votre texte explicatif>

PARTIE 2 [35 points] : (cette partie contient 3 sous-parties)

1/ Faites le diagramme de classe UML du [COMPOSANT 1] en mettant en évidence les Designs Patterns vus en cours. Commentez le diagramme.

<Ecrire ici votre diagramme de classe avec son commentaire>

2/ Ecrire le pseudo-code ou le code Java des traitements suivants du COMPOSANT 1 :

- la création d'un nouveau PLN
- l'abonnement d'un panneau au serveur
- la notification des panneaux, suite à la modification d'un plan de vol.

<Ecrire ici vos pseudo-codes ou codes Java>

(Suite du sujet page suivante)

3/ Faites le diagramme de classe UML du [COMPOSANT 2] en mettant en évidence les Designs Patterns vus en cours.
Commentez chacun de ces diagrammes.

<Ecrire ici votre diagramme de classe avec son commentaire>

Conseils :

Un composant logiciel [COMPOSANT X] correspond à une JVM ou process. Cela signifie que les COMPOSANTS X communiquent sur le réseau à travers des interfaces distantes.

Ainsi, pour une description précise de vos diagrammes de classe, on fait le choix que toutes les communications distantes entre les composants sont réalisées en RMI (utilisation de la classe URO = UnicastRemoteObject et de l'interface Remote).

Fin du sujet