

IPST-CNAM
Intranet et Designs patterns
NSY 102
Mercredi 15 Avril 2020

Durée : **2 h 15**
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 2019-2020.
Ce fichier est un fichier word. Il doit rester dans ce format.

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

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

3/ Vous répondez à tous les énoncés, en insérant vos textes et vos diagrammes en-dessous de
chaque énoncé. Vous ne modifiez pas les énoncés.

4/ Au bout de 2 heures et 15 minutes, c'est à dire à **16h00**, 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 me téléphonez au afin de me prévenir de cette difficulté
- vous m'envoyez par téléphone une photo de chaque page de votre copie (précisez votre
nom dans le SMS)
- dès que possible, vous déposerez votre copie comme prévu, et vous m'enverrez un
message par SMS pour me prévenir du dépôt (précisez votre nom dans le SMS).

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 suivantes en mode paysage afin que les copiés/collés de vos
diagrammes soient plus visibles.

1. **EXERCICE 1** [10 Points] (15mn)

Expliquez avec vos propres mots (ne pas faire de copié/collé avec le cours) pourquoi il est souhaitable d'utiliser les « Designs Patterns » dans la conception d'un Système d'Information, et quelles sont les difficultés que l'on peut rencontrer. Soyez convaincant, explicatif et démonstratif dans vos arguments.

<Ecrire ici votre réponse>

2. **PROBLEME** [50 points] (2h)

Nous envisageons de réaliser un Système d'Information (SI) générique de supervision d'éléments informatiques.

Ce SI est composé de 3 types de COMPOSANTS :

- le [COMPOSANT 1], qui est unique, appelé SERVEUR, qui crée, en mémoire, les Objets de Supervision.
- les [COMPOSANT 2] appelés IHM qui affichent et animent les Objets de Supervision qui sont situés sur le SERVEUR.
- les [COMPOSANT 3] appelés APP qui changent l'état des Objets de Supervision qui sont situés sur le SERVEUR.

Tous les Objets de Supervision sont des instances de classes C1, C2, ... qui héritent d'une même classe abstraite ou hérite d'une classe qui hérite de cette classe abstraite. Cette classe abstraite implémente l'interface **ObjetSupervision** (cette interface n'hérite pas de Remote).

Cette interface est la suivante :

String getIdent() : retourne l'identification de l'objet de supervision.

void setAlarme(int numeroAlarme, String texte, boolean etatAlarme, boolean clignotante) : crée, ou change si existe déjà, le texte et l'état de l'alarme d'un numéro donné et précise si l'alarme doit clignoter ou pas

void setTexte(int numeroTexte, String texte) crée, ou change si existe déjà, la valeur d'un texte d'un numéro donné de l'objet de supervision

ImageOS getImage() retourne une "image" de l'objet de supervision dans un format dédié XML.

Un fichier de configuration permet, au SERVEUR, de savoir, pour chaque classe Cn, combien d'objets de supervision il doit créer. L'identification d'un objet de supervision est le nom de la classe suivi de "#001", "#002", ...

Les classes C1, C2, ... sont déposées dans un répertoire du SERVEUR (fichiers .class).

Dès qu'une nouvelle classe Cn est déposée dans ce répertoire, le SERVEUR charge la classe et remplace toutes les instances existantes de cette classe par autant de nouvelles instances.

Ainsi, il n'est pas utile d'arrêter le SERVEUR pour ajouter (ou supprimer) de nouveaux objets de supervision.

A chaque fois qu'un setteur d'un objet de supervision est utilisé, l'objet de supervision maintient une "image" de représentation graphique de son état dans un format XML spécifique. Ainsi chaque classe d'objet de supervision sait comment se représenter graphiquement (voyant, jauge, thermomètre, ...). La méthode **getImage** permet de récupérer cette "image".

La classe **ImageOS** correspond à l'implémentation d'un format XML spécifique de description de l'objet de supervision qui sera ensuite interprété par les composants IHM pour visualiser les objets de supervision.

Un composant APP demande au SERVEUR la liste des Objets de Supervision qu'il est en train d'exécuter. APP peut alors mettre à jour les objets de supervision de son choix en utilisant leurs setteurs.

Les composants IHM se déclarent au SERVEUR. Ils sont notifiés par le SERVEUR dès qu'un objet de supervision est modifié.

Un composant IHM (si elle est concernée par cette notification) affiche alors l'image calculée par l'objet de supervision. Il est à noter que l'"image" peut contenir des boutons, zones de saisie, ... permettant à l'opérateur de l'IHM d'agir sur la représentation graphique, et même modifier l'objet de supervision situé sur le serveur.

Lors de son initialisation, l'IHM demande au SERVEUR la liste des objets de supervision qu'elle souhaite afficher, et affiche leurs images.

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] :

1/ Faites le diagramme de classe UML du [COMPOSANT 1] en mettant en évidence les Designs Patterns vus en cours [20 points].
Commentez ce diagramme [5 points].

<Ecrire ici votre diagramme de classe>

<Ecrire ici votre texte explicatif>

2/ Faites le diagramme de classe UML du [COMPOSANT 2] en mettant en évidence les Designs Patterns vus en cours [7 points].
Commentez ce diagramme [3 points].

<Ecrire ici votre diagramme de classe>

<Ecrire ici votre texte explicatif>

Conseils pour le PROBLEME :

Un composant logiciel [COMPOSANT X] correspond à une JVM ou process. Cela signifie que les COMPOSANTS X communique 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