

Devoir Maison

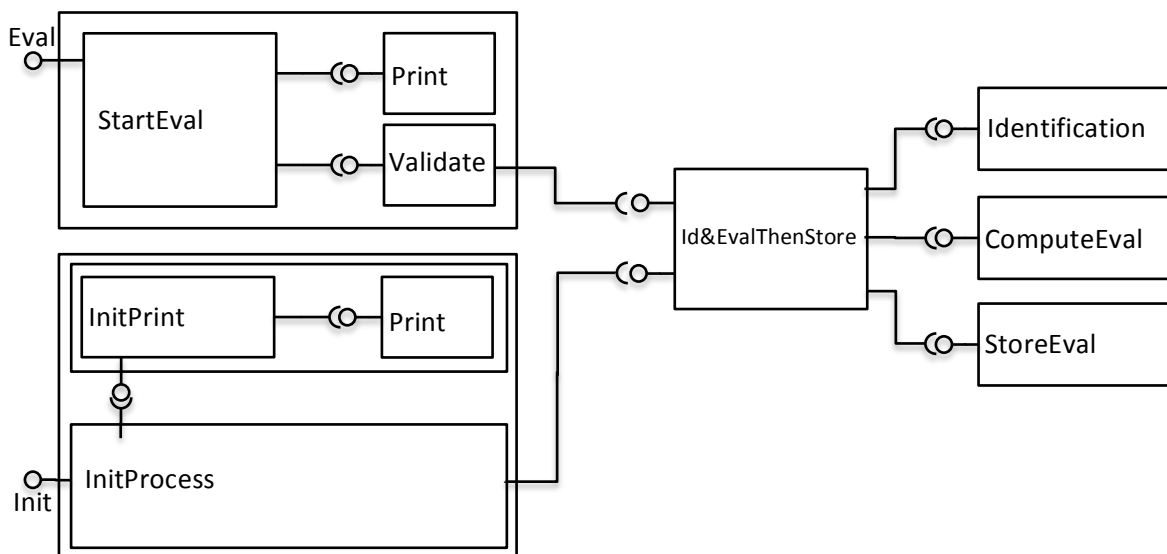
Programmation par composants

L'objectif de cette petite épreuve est de vous donner les moyens de montrer vos compétences en ce qui concerne la programmation par composants. Pour cela, comme en TP, vous devrez utiliser le plug-in MAY.

Ce devoir n'est pas très volumineux et ne devrait pas vous demander plus d'une heure de travail. Pour vous aider, vous avez à disposition toutes les ressources de cours/TD/TP disponible sur le site Moodle du module (<http://moodle.univ-tlse3.fr/course/view.php?id=1503>). Par ailleurs, nous y avons préparé un forum sur lequel vous pourrez y poser des questions et chercher de l'aide de la part de vos collègues ou des enseignants de TP.

Sujet

Le sujet consiste à implanter un assemblage à réaliser sous forme d'un composant composite, composé de plusieurs composants primitifs ou composites eux-mêmes. L'assemblage est représenté par la figure ci-dessous qui utilise la syntaxe graphique de SpeADL.



Il ne vous est pas demandé de le réaliser entièrement. En revanche, pour en comprendre globalement la logique, sachez que cet assemblage nous servira à calculer et à enregistrer automatiquement votre note. Pour vous donner une idée générale du comportement de cet assemblage, voici quelques éléments :

- Le composant *Id&EvalThenStore* est un composant à état qui peut (d'abord) récupérer et mémoriser l'identité d'un candidat. Ce composant peut (ensuite) provoquer l'évaluation du candidat et retourner une note. Dès que cette note est connue, le composant déclenche l'enregistrement de la note. Pour ces trois opérations, les composants *Identification*, *ComputeEval* et *StoreEval* sont successivement utilisés.
- Par l'interface *Init*, l'assemblage permet de démarrer le processus d'évaluation. Cela consiste à demander l'identité du candidat pour ensuite réaliser un affichage qui signale le début du processus pour le candidat identifié.
- L'interface *Eval* de l'assemblage offre la possibilité de déclencher l'évaluation à proprement parler. Un affichage signale lorsque le processus est terminé et quelle est la note obtenue.

Un composant *Validate* intervient lors de l'implantation de ce service pour s'assurer que l'évaluation obtenue donne une note valide.

Pour vous donner une base, vous trouverez sur Moodle deux archives de projets à importer dans lesquels se trouvent des types de services (sous la forme d'interfaces Java) et des définitions et implantations de composants. Pour vous guider dans votre construction, veuillez suivre les questions ci-dessous. Pour implanter plus facilement les dépendances entre les paquetages, nous vous conseillons de réaliser chaque question dans un (ou plusieurs) projet(s) différent(s) pour lequel (lesquels) vous pourrez préciser des dépendances spécifiques.

1. Implantations de composant primitif

Etant donnée la définition de composant *Identification* présente dans le fichier de spécification *DefQ1.speedl*, donnez deux implantations de ce composant :

- l'une qui donne comme chaîne résultat votre nom, votre prénom et le numéro de votre groupe de TP (dans le module AL), dans cet ordre et séparés par un « / ».
- l'autre qui donne comme chaîne résultat votre nom et votre numéro de carte étudiant (8 chiffres), dans cet ordre et séparés par un espace.

2. Complément d'une définition de composite

Etant donné le fichier de spécification *DefQ2.speedl* et les implantations de *Print* et de *CompositeQ2*, complétez l'assemblage selon les 3 recommandations suivantes :

- Donnez une spécification de composant *InitPrint* pour valider l'assemblage.
- Donnez une implantation de ce même composant qui étant donné un identifiant passé en paramètre, construit un message signalant le début du processus d'évaluation pour ce candidat puis en délègue l'affichage au composant *Print*.
- Donnez une autre implantation du composant *Print* qui stocke le message passé en paramètre dans un fichier de log ainsi qu'une implantation d'un composite qui utilise votre implantation de *InitPrint* et cette implantation de *Print*.

3. Composition et réutilisation

Etant données les définitions des composants *Print*, *Validate* et *StartEval* dans les fichiers de spécification *DefQ2.speedl* *DefQ3.speedl* ainsi que leur implantation suivez les 2 recommandations suivantes :

- Donnez la spécification de composant qui les englobe et qui offre le service *Eval*.
- Donnez une implantation de ce même composant.

4. Réutilisation et extension

Dans les paquetages *question4* des projets fournis se trouvent tous les éléments nécessaires à la définition et l'implantation du composant *InitProcess* selon le schéma. Il vous est demandé de définir et d'implanter un composant du même type que *InitProcess* mais qui l'englobe dans un traitement consistant à afficher un message avant et un message après.

5. Assemblage total

Etant données les types des interfaces de service des composants *Identification*, *ComputeEval* et *StoreEval*, proposez des implantations basiques (et pas forcément réalistes) permettant de réaliser l'assemblage complet et de le faire fonctionner.