

Licence Informatique 3ème année

Modélisation Objet

Examen de 2ème session – 26/06/2017 – 2h

Les notes de cours, TD et TP sont autorisées.



La SNCF veut développer un logiciel de suivi des trains de voyageurs pour que les cheminots et les voyageurs puissent savoir à tout moment où sont les trains.

Question 0 : Le logiciel doit permettre aux voyageurs de connaître l'état d'un train, en fournissant le numéro du train. Les chefs de gare peuvent également connaître l'état d'un train, mais ils peuvent aussi indiquer qu'un train vient d'être constitué dans une gare (c'est-à-dire que les wagons et locomotives ont été assemblés à quai), qu'un train vient de démarrer d'une gare ou qu'un train vient d'arriver à une gare. Un chef de gare peut également indiquer qu'un train a été annulé. Il peut aussi indiquer un retard (en minutes) pour un train donné. Un chef de gare doit toujours donner le numéro du train pour effectuer une des actions précédentes. Finalement, dans certaines gares, un dispositif automatique est mis en place avec des capteurs à l'entrée et en sortie de gare, qui peuvent indiquer au logiciel qu'un train est arrivé en gare ou sorti d'une gare en fournissant son numéro (le numéro est obtenu automatiquement par le capteur).

Réalisez un diagramme de cas d'utilisation correspondant à ces fonctionnalités. (4 points)

Question 1 : Un train possède un numéro (entier) et un retard (en minutes). Il suit un itinéraire constitué de gares (au moins deux), dans un certain ordre. On connaît pour chaque gare où le train s'arrête l'horaire d'arrivée et l'horaire de départ du train (on peut utiliser la classe `java.util.Date` du langage Java). Parmi ces gares, il y a une gare de départ, pour laquelle il n'y a pas d'horaire d'arrivée et une gare terminus pour laquelle il n'y a pas d'horaire de départ. Chaque gare possède un nom. Un train a un état qui peut prendre pour valeur : PRÉVU, EN_GARE, EN_ROUTE, ANNULÉ. On peut connaître par un appel de méthode le retard d'un train en minutes (éventuellement 0 minute si le train est à l'heure). De plus, on sait à tout moment quelle est la dernière gare traversée par le train (il peut s'agir de la gare où le train se trouve s'il est dans l'état EN_GARE).

Réalisez un diagramme de classes d'analyse correspondant à cette description (4 points)

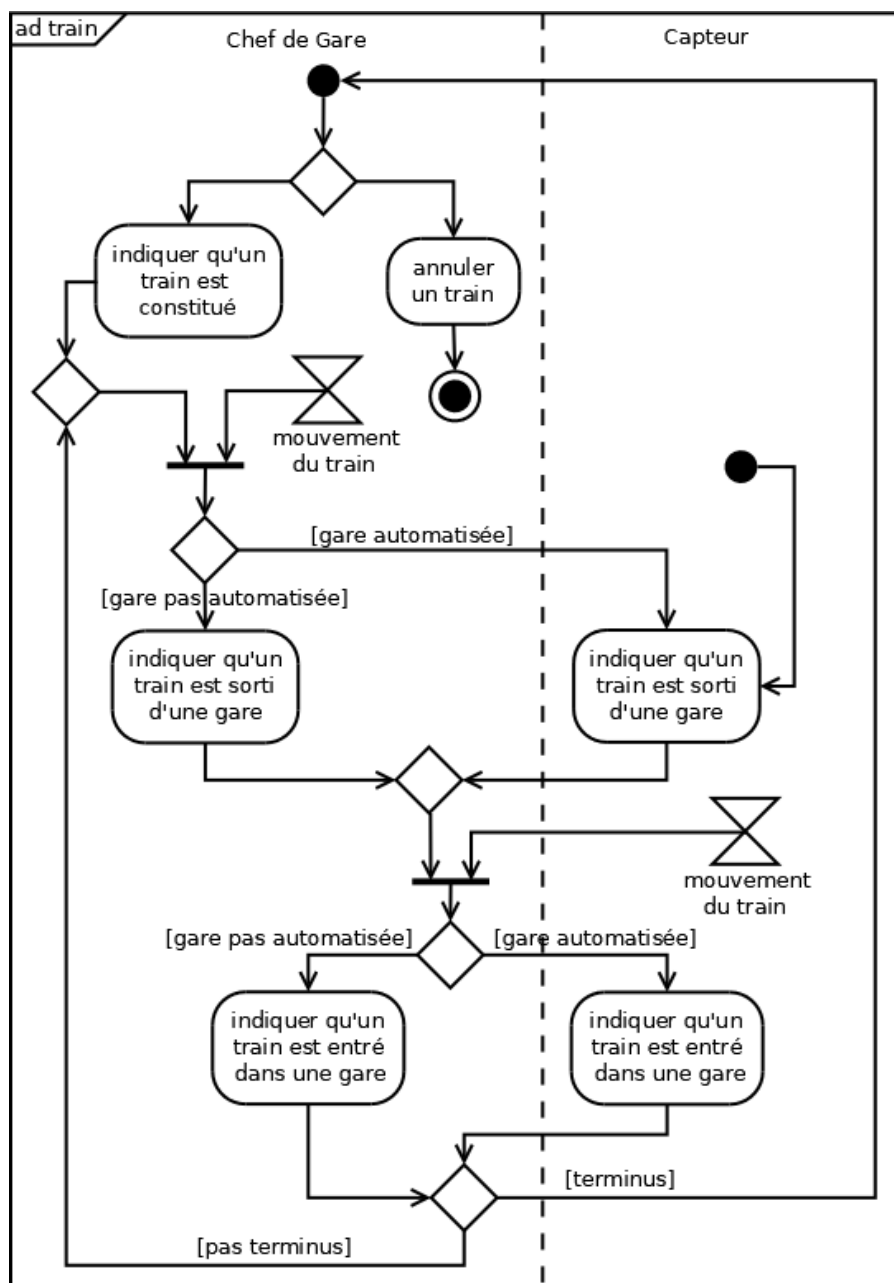
Question 2 : On veut préciser la façon dont les trains passent d'un état à l'autre. Quand on crée un objet train, il est dans l'état PRÉVU. Quand un chef de gare indique que le train est constitué, le train passe dans l'état EN_GARE. Quand un chef de gare ou un capteur indique qu'un train quitte une gare et que le train est EN_GARE, il passe dans l'état EN_ROUTE. Quand un chef de gare ou un capteur indique qu'un train arrive dans une gare et que le train est EN_ROUTE, il passe dans l'état EN_GARE. Mais si le train arrive dans la gare terminus, il repasse dans l'état PRÉVU. Si, dans l'état PRÉVU, un chef de gare annule le train, il passe dans l'état ANNULÉ. Si, dans l'état ANNULÉ, il est passé minuit, le train redevient PRÉVU (on considère que chaque train roule tous les jours, week-end compris).

Réalisez un diagramme d'états correspondant à ce fonctionnement. (4 points)

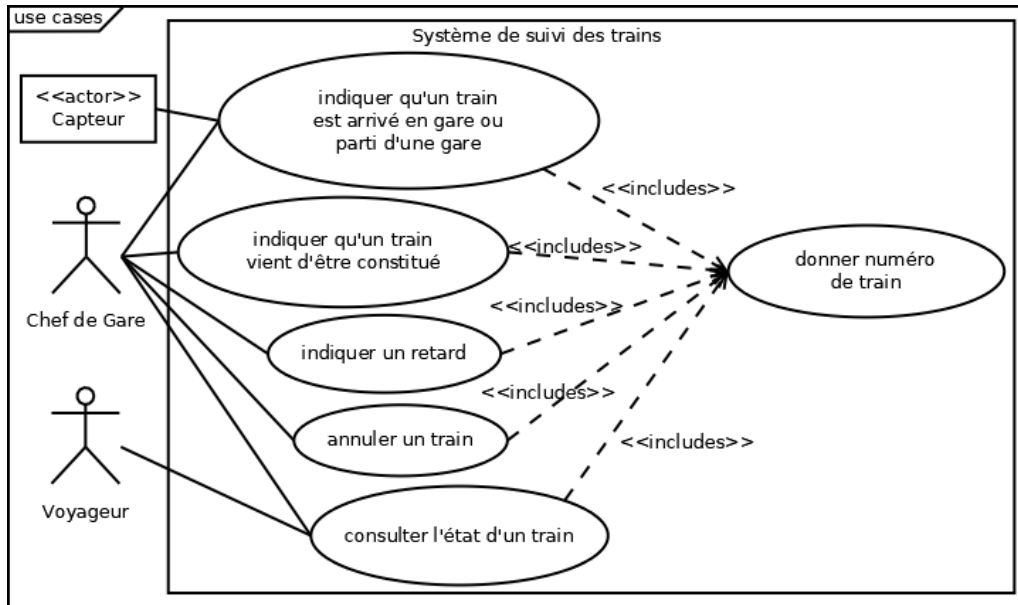
Question 3 : On considère maintenant qu'on est rendu à l'étape de **conception** du logiciel. Il faut donc détailler le fonctionnement **interne** du logiciel à l'aide d'un scénario. On suppose ici que les objets représentant les différents trains existent déjà dans le logiciel et que les actions des utilisateurs ne font que modifier les états et données des trains. Le scénario est le suivant : un chef de gare indique qu'un train a été constitué en fournissant son numéro, ce qui initialise l'état du train et entraîne une modification interne à l'objet train qui initialise la dernière gare traversée. Puis un capteur indique que le train en question a quitté la gare. Le train passe alors dans l'état EN_ROUTE et recalcule lui-même son retard (en comparant l'heure de départ prévue et l'heure de départ effective). Finalement, un chef de gare (différent du premier) indique que le train est entré en gare, ce qui entraîne la mise à jour de l'état du train (qui passe dans l'état EN_GARE). Cette mise à jour entraîne une modification interne à l'objet train qui modifie la dernière gare traversée (on suppose que cette modification entraîne le recalcul du retard). Si cette gare est le terminus, l'objet train passe dans l'état TERMINUS.

Réalisez un diagramme de séquences qui décrit ce scénario. (5 points)

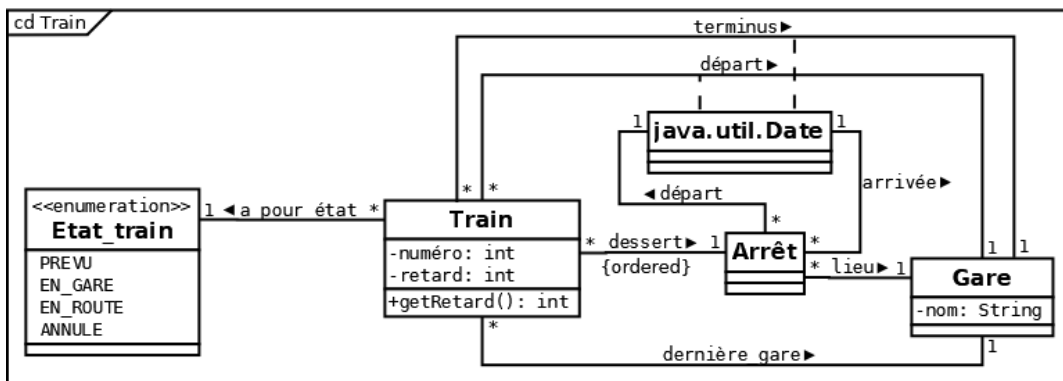
Question 4 : Le diagramme d'activités suivant indique comment les chefs de gare et les capteurs indiquent au système les mouvements d'un train. Indiquez les erreurs contenues dans le diagramme en justifiant pourquoi ce sont des erreurs (il s'agit d'erreurs liées au langage UML, et non d'erreurs sur le fond). (3 points)



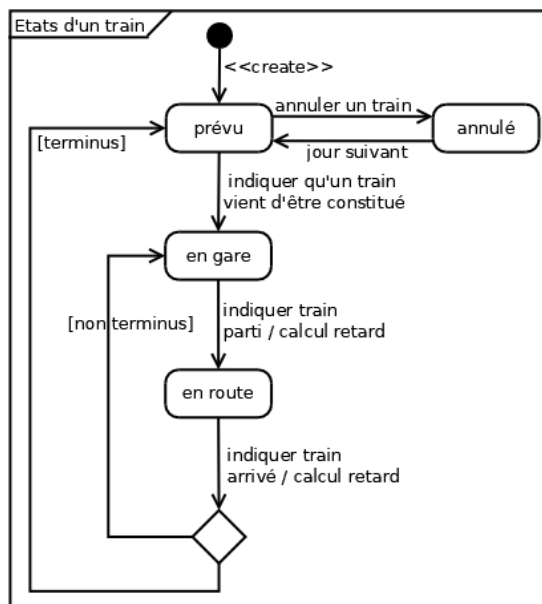
Question 0 :



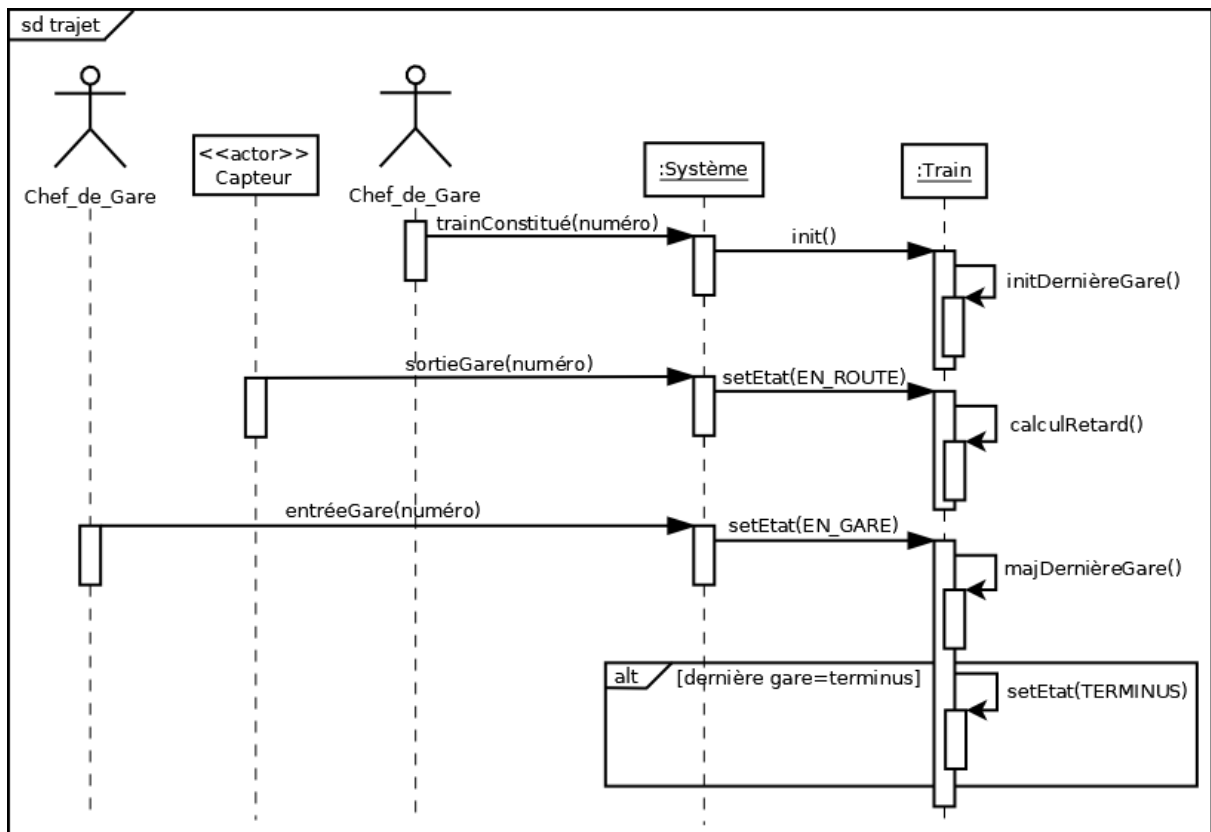
Question 1 :



Question 2 :



Question 3 :



Question 4 :

L'état initial coté capteur ne devrait pas exister. Sur le point de choix le plus haut, il manque des conditions pour que le diagramme soit déterministe. La transition qui sort du dernier point de choix ne doit pas retourner à l'état initial mais à l'état final.