



Lisez tout l'énoncé, et en particulier les informations pratiques avant de commencer.



### Informations pratiques.

- I) Lors de ce TP nous utiliserons le programme StarUML (sous Windows). Vous pouvez utiliser votre propre machine si vous le souhaitez.
- II) On utilisera dans ce TP un seul projet et un seul modèle (mais bien sûr plusieurs diagrammes).
- III) N'oubliez pas de sauvegarder régulièrement.
- IV) Si à l'issue du temps imparti le diagramme sur lequel vous êtes en train de travailler n'est pas fini, l'indiquer par une note sur le diagramme.
- V) Le fichier .uml ainsi que le projet doivent être nommés avec votre nom de famille.
- VI) Chaque diagramme devra être exporté comme image. Le nom des images doit se conformer au schéma suivant : `numéro_de_question-votrenom.jpg`.
- VII) À l'issue du TP, le sauvegarder puis l'envoyer par e-mail : `mathieu.sassolas@u-pec.fr` avec comme objet « [L3P – UML] TP solo ». Attendre l'accusé de réception avant de se déconnecter et de quitter le poste et la salle.
- VIII) Le TP est un examen, il est donc individuel. Les notes de cours sont autorisées. Aucun autre document n'est autorisé.

### Exercice. Une application mobile de « e-santé »

Un assureur (malhonnête) décide d'utiliser des données sur les habitudes de vie de ses assurés pour adapter le prix de leur assurance santé en conséquence.

Pour obtenir un assurance, un client commence par en faire la demande auprès de l'assureur. Celui-ci lui installe donc une application sur son smartphone. Le client va vivre sa vie, et en particulier marcher et manger. À chaque fois qu'il marche, l'accéléromètre du téléphone compte le nombre de pas qu'il effectue. À chaque repas, il doit indiquer sur son téléphone le menu qu'il a consommé. L'assureur récupère alors les statistiques sur les habitudes de son client ; il peut, s'il trouve que l'échantillon n'est pas suffisant, demander à ce que le client continue les mesures. Sinon, il évalue le prix et établit un devis à partir des statistiques. Le client peut alors contester ce devis et demander à continuer l'évaluation de ses habitudes de vie. S'il accepte, l'assureur établit un contrat qui sera alors signé par les deux parties (dans n'importe quel ordre).

1. Construire le **diagramme d'activités** du processus de demande d'assurance.
2. Construire le **diagramme d'états** du dossier du client, de la demande jusqu'au contrat signé.

On s'intéresse maintenant à l'application mobile proprement dite. Afin de pouvoir calculer des statistiques, elle garde toujours l'ensemble des menus associés à un profil (le profil de l'utilisateur contient également son nom, prénom, et date de naissance). Chaque séance de marche est également associée au profil ; une séance de marche est un nombre de pas effectués à une certaine date (il n'est pas spécifié la manière dont une « séance » est déclarée finie). Un menu, qui a été pris à une date précise est composé de un ou plusieurs plats, pour qui on garde un nom et une valeur calorique. Cette valeur calorique aura été récupérée sur un serveur de l'Association des Nutritionnistes des Assureurs Véreux (ANAV). Pour un profil utilisateur donné, on doit pouvoir calculer le nombre de calories ingérées pour une date donnée, ainsi que ses statistiques globales.

3. Construire le **diagramme de cas d'utilisation** de cette application. (NB : on évoque également l'application au début de l'exercice.)
4. Construire le **diagramme de classes métiers** de l'application.
5. Construire le **diagramme de séquence** de l'opération `calcul_calories` de la classe `Profil`, qui prend une `Date` en argument et renvoie un nombre (entier) de calories : le nombre de calories consommées par tous les plats de tous les menus ingérés à cette date. On prendra soin ici d'ajouter au diagramme de classes de la question précédente toutes les opérations (accesseurs...) nécessaires à la construction de ce diagramme de séquence.