

Lors de ce TP, la réalisation en parallèle du diagramme de séquence et du diagramme de classes d'une application seront étudiées dans le cadre d'une conception simple.

TP n°4

Conception objet

Alexandre GUIDET

Le TP sera réalisé en utilisant Visual Paradigm. Un projet est fourni sur le commun, il contient le résultat de la phase d'analyse, à savoir un diagramme de cas d'utilisations et un diagramme de classes, ainsi qu'un diagramme de séquence décrivant le scénario « commander un menu ». L'application est celle étudiée au TD n°2 (restaurant de burgers).

Le but du TP est de réaliser la conception de l'application.

CONCEPTION ARCHITECTURALE

Créez un diagramme de package dans le projet. Créez deux paquets : la partie « métier » qui a déjà été analysée, et la partie « IHM » qui représente l'interface homme-machine. Indiquez la dépendance suivante : le paquet « ihm » utilise le paquet « métier » (mais pas l'inverse). Ajoutez également un paquet « banque » qui représente la partie « gestion bancaire » de l'application. Indiquez les dépendances avec ce paquet.

Associez le diagramme de classes présent dans le projet au paquet « métier ». Pour cela, il suffit de faire un clic-droit sur le paquet, de choisir « sous-diagrammes → existing diagrams » et de choisir le diagramme des classes.

DIAGRAMME DE CLASSES POUR L'IHM

Créez un diagramme de classes comme sous-diagramme du paquet « ihm ». L'analyse de l'IHM donne un découpage de l'application en trois écrans :

- Ecran d'accueil où l'on peut choisir les menus, les boissons, les burgers, le café
- Ecran de détail sur un menu (choix de la boisson, des frites, option café)
- Ecran de paiement

Chaque écran donnant lieu à une classe, complétez le diagramme de classes d'IHM. Indiquez les liens entre les écrans par des dépendances « *instanciate* ». Pour l'instant laissez ces classes vides.

CONCEPTION : DETERMINATION DES OPERATIONS

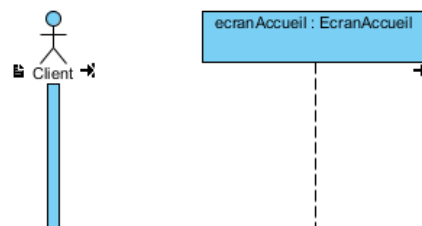
Nous allons passer à la phase de conception, à savoir déterminer les opérations nécessaires dans les classes, ainsi que les liens entre celles-ci, pour réaliser les fonctionnalités demandées.

Rouvrez le diagramme de cas d'utilisation.

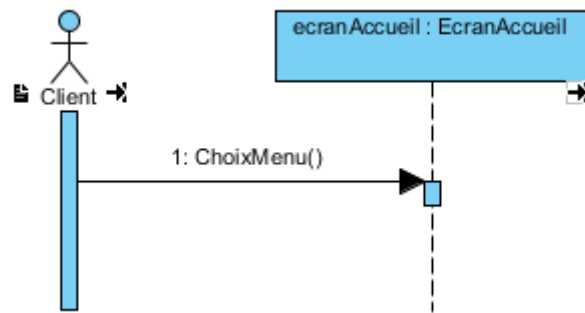
Commençons par la fonctionnalité « commander un menu » : créez un diagramme de séquence comme sous-diagramme du cas. Si ce n'est pas déjà le cas, faites afficher l'explorateur de modèles (onglet aperçu, volets, explorateur de modèles).

Glissez l'acteur « client » de l'explorateur de modèles vers le diagramme de séquence nouvellement créé.

Pour réaliser la commande, le client doit interagir avec le premier écran, et donc envoyer un message (exécuter une opération) à la classe correspondante. Placez une instance de cette classe, par glisser-déposer depuis l'explorateur de modèle. Vous devez avoir un diagramme ressemblant à celui-ci.



Sur l'écran d'accueil, il y a un bouton pour commander un menu. Il y a donc un message de l'utilisateur vers l'instance de l'écran. Créez ce message (type « *call* »). N'indiquez aucun texte pour l'instant. Ce message va correspondre à une opération dans la classe. Faites un clic-droit sur le message, puis « *type* → *call* →



create operation ». Dans la fenêtre qui s'ouvre, indiquez ChoixMenu comme nom d'opération et validez. Vous devez avoir un diagramme comme ci-contre.

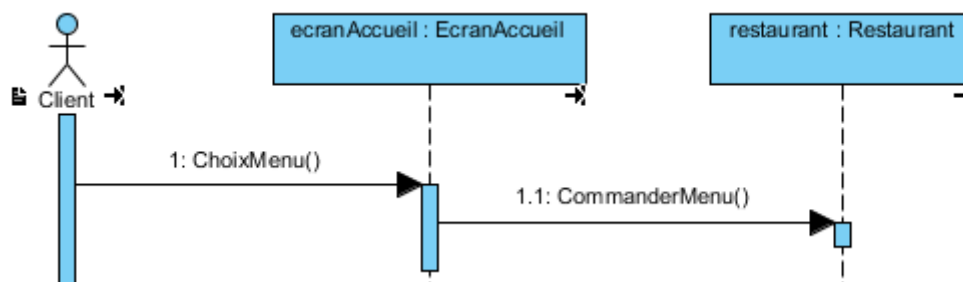
Rouvrez le diagramme de classes d'IHM : vous remarquerez qu'une opération est apparue dans la classe de l'écran d'accueil : il y a une **synchronisation** entre les deux diagrammes.

Ajoutez un stéréotype « event » sur cette opération (une entrée utilisateur est un évènement pour l'IHM).

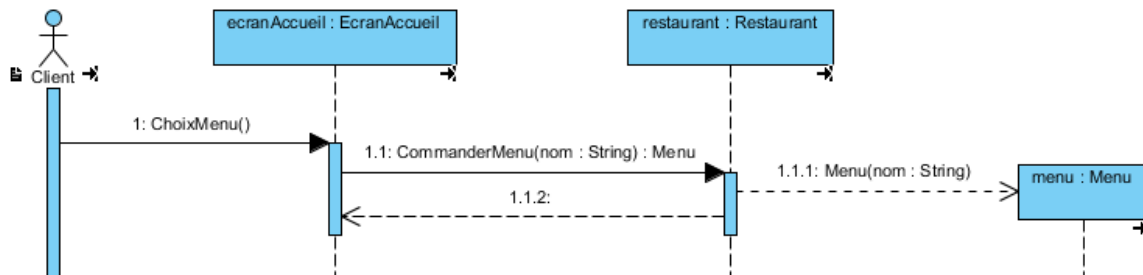
Le choix du menu (dans l'IHM) va entraîner l'envoi d'un message dans la partie « métier ». Dans notre cas, il faut envoyer un message à un objet de type « Restaurant » pour signaler le choix d'un menu. Retournez dans le diagramme de classes d'IHM. Est-ce que la classe d'écran peut envoyer un message à un objet de type Restaurant ? Non ! Il faut donc que ces deux classes soient associées ! Ajoutez la classe Restaurant (par glisser-déposer depuis l'explorateur de modèle) dans votre diagramme de classes d'IHM et associez la classe d'écran d'accueil à la classe Restaurant. Réglez les rôles et cardinalités ! Vous devez avoir un schéma comme ci-contre.



Revenez dans le diagramme de séquence et ajoutez sur le diagramme une instance de Restaurant. Nous allons à présent indiquer que, lorsque l'écran reçoit le message « ChoixMenu », il appelle l'opération « CommanderMenu » du restaurant. Le diagramme devient :



Continuons à préciser les opérations... quand on commande un menu, il faut créer une instance de Menu ! Déposez donc sur le diagramme une instance de Menu, et envoyez-lui depuis le restaurant un message de **création**. Associez le message avec le constructeur du menu (double-clic sur le message ou clic-droit, opération). Celui-ci demande une chaîne (le nom du menu) qui doit donc être fournie par le restaurant ! Modifiez l'opération CommanderMenu de la classe Restaurant pour qu'elle accepte une chaîne nom en paramètre et qu'elle retourne un Menu. Ajoutez également le message de retour pour indiquer que l'opération renvoie une valeur. Le diagramme de séquence devient:

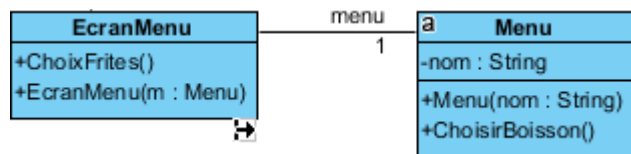


Vous avez compris le principe ? Alors on continue le travail... Si l'on suit le diagramme de séquence d'analyse, une fois le menu commandé, il faut passer à l'écran d'accueil... Modifiez le diagramme de séquence « commander un menu » en indiquant que l'écran d'accueil crée une instance de l'écran de choix de menu.

Le message « ChoixMenu » est à présent traité... Mais la séquence n'est pas terminée !

L'utilisateur doit à présent choisir sa quantité de frites... Ajoutez un message entre « Client » et l'écran de menu et ajoutez une opération « ChoixFrites ». Cette opération doit envoyer un message au menu pour lui indiquer la quantité de frites choisie... Mais l'écran de menu ne peut pas (encore) envoyer de message au menu !!! Il faut donc modifier le diagramme de classes d'IHM pour que :

- La classe d'écran de menu soit **associée** avec un menu
- Le **constructeur** de cette classe reçoive un menu en **paramètre**



Cela devrait vous donner le diagramme ci-contre.

Vérifiez que l'écran d'accueil puisse bien envoyer le menu à l'écran de menu !

A présent, lors de la réception du message « ChoixFrites », l'écran d'accueil peut transmettre au menu un message « ChoixFrites (qté) » et créer une instance de la classe Frites...

Faites de même pour la boisson.

Nous sommes bientôt à la fin de cette séquence ! Il ne reste plus qu'à payer (oublions le café pour l'instant). Pour cela, l'utilisateur valide sa commande dans l'écran de menu (et donc il faut ajouter une opération...) ce qui « passe la main » à l'écran de paiement. Cet écran doit pouvoir savoir combien il faut faire payer... il faut donc stocker un **tarif** dans un attribut de cet écran, et initialiser cet attribut via le **constructeur**. Afin d'avoir ce prix, il faut donc le demander au menu...

Ne continuez plus cette séquence : la séquence de paiement est commune à d'autres cas... Faites un nouveau diagramme de séquence, comme sous-diagramme du cas « payer » !

Déposez sur ce nouveau diagramme le client (acteur) et une instance de l'écran de paiement.

L'utilisateur doit insérer sa carte (message, donc opération...) puis saisir son code et valider. La validation (message/opération) va déclencher un message dans une autre partie de l'application (la gestion bancaire) que nous ne traiterons pas aujourd'hui...