

Il s'agit d'implémenter en équipe de trinôme la solution « Simulation Bank » en langage C++ orienté objet. Choisissez vos trinômes et envoyez un mail (en mettant les deux autres en CC) à [Tianxiao.Liu@cyu.fr](mailto:Tianxiao.Liu@cyu.fr) pour l'informer. Pour faciliter la communication avec l'enseignant en différentes UE cette année, rejoignez le serveur Discord : <https://discord.gg/7ZbS9SRc>

### A faire

- 1) Implémenter en langage C++ la simulation bancaire correspondant au diagramme de classes fourni.
- 2) Extensions possibles (choisissez un seul parmi les aspects suivants) :
  - a. Enrichir la simulation : pour les aspects clients, opérations, caissiers, file(s) d'attente : proche du réel. Vous pouvez proposer librement les extensions logiques et intéressantes à ce sujet.
  - b. Enregistrer les données statistiques dans une bases de données (pas besoin de relecture de données) et afficher graphiquement les résultats statiques.
  - c. Transformer la simulation en temps réel et afficher avec une IHM graphique simple, le déroulement de la simulation (animations).
- 3) Envoyez votre rendu en code source compressé (fichiers .cpp + fichiers header) au plus tard par email à l'enseignant le 14 septembre à 12H00. Ecrivez brièvement dans votre email comment vous avez travaillé à trois pour obtenir ce résultat. Il n'y a pas besoin de rapport.

### Rappel de la simulation bancaire (synthèse de la version 1 et 2 en COO)

Vous avez besoin de concevoir un simulateur qui vous permet de simuler une période de déroulement d'une agence bancaire. La simulation se déroule pendant un certain nombre de temps unitaires. L'horloge de la simulation se présente en unité de temps. Au début, unité 0, et puis 1, 2, 3, 4, 5... On suppose que pendant la simulation, les clients arrivent aux moments « uniformément répartis ». C'est-à-dire qu'un client arrive à tous les N (paramétrable) unités de temps. Par exemple, si N = 5, un nouveau client arrive à 0, 5, 10, 15, 20 ... unité de temps.

Chaque caissier met un temps aléatoire (entre deux valeurs min et max paramétrables) pour traiter un client. Quand un client arrive, si aucun caissier n'est libre, on le met dans la file d'attente, sinon, le premier caissier libre trouvé va le servir. Il n'y a qu'une seule file d'attente partagée par tous les caissiers. Le nombre de caissiers doit être certainement paramétrable afin d'analyser des résultats de différentes simulations avec différentes entrées.

Il s'agit donc, pour chaque itération de la simulation, de mettre à jour l'état des caissiers et de la file d'attente, et de traiter un client arrivant le cas échéant. Pour simplifier, la fin de la simulation s'arrête quand la durée prédéfinie est atteinte, même si certains caissiers ne finissent pas encore le service du client courant.

On distinguera trois types de demandes de client : consultation, demande de virement, retrait important d'espèce. On suppose qu'un nouveau client arrivant aura une seule demande d'un type quelconque, avec une probabilité identique (1 / 3) pour chaque type. Pour les clients ayant une demande de consultation (considérée comme non urgente), ils ne sont pas patients : ils partent après une attente de plus N unités de temps. En revanche, les deux autres opérations sont considérées urgentes.

On a un type de client spécial « VIP » dans la simulation. Le taux des clients VIP pour la simulation est paramétrable. Ces clients VIP sont prioritaires. A chaque fois qu'un caissier finit de servir un client (normal ou VIP), il vérifie s'il y a des clients VIP dans la file d'attente, si oui, il servira le client VIP le plus proche de la tête de la file d'attente, sinon, il prendra le premier client de la file d'attente.