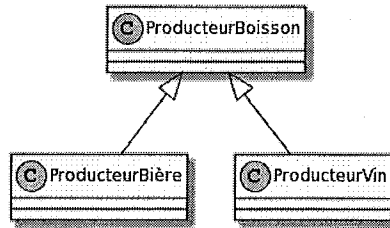


Durée : 2 h. Aucun document n'est autorisé à part une feuille A4 recto-verso. L'usage de la calculatrice ou d'un *smartphone* est interdit. Le sujet contient 2 pages **recto-verso**.

**1** (/10) Questions ouvertes QT et POO

1. Dans le cadre du développement d'une application permettant la gestion de produits utilisés par un restaurateur, il a été décidé de représenter les informations de chaque producteur par une classe, avec une hiérarchie de classe.



- (a) quelles sont les avantages et les inconvénients d'utiliser cette hiérarchie de classe pour la modélisation et développement de l'application ?
- (b) dans quel cas et pour quelles raisons, la classe « ProducteurBoisson » pourrait-elle être définie comme abstraite ?
- (c) comment la notion d'interface de POO est-elle implémentée en C++ ?
2. Quelle est la classe de base de toutes les autres classes de QT ?
3. Comment se fait l'émission d'un signal en utilisant QT ?
4. Donner et expliquer le prototype d'une connexion signal/slot de QT.
5. Le widget MyWidget1 a un slot mySlot (). La méthode MyWidget1::calcul () doit déclencher un signal qui sera intercepté par mySlot (). Écrivez le signal et la connexion correspondants, où placez-vous ce code ?
6. On dispose d'un container de type `std::list<float>`. Écrire la fonction retournant la moyenne des valeurs de ce container. Lier cette fonction à un clic sur le QPushButton nommé bt du widget MyWidget2. Où seront placés ces codes ?
7. Qu'est-ce que le MOC de QT et à quoi cela sert-il ?

## 2 (/10) Application

On veut construire une application QT qui permette de modéliser le travail de fabrication d'une boulangerie. Elle doit s'assurer que les quantités d'ingrédients (limités pour l'exercice à eau et farine) sont suffisantes, lancer la fabrication et informer l'utilisateur. La **Boulangerie** ne stocke pas ces ingrédients. Au moment où la **Boulangerie** va vouloir produire du pain, elle va donc vérifier chez les producteurs **Farine** et **Eau** que les quantités correspondent à la demande (on pourra modifier ces quantités nécessaires avant la demande). Elle dépend donc de ces 2 acteurs.

1. votre premier travail va être de construire les classes de bases et connexions de ces acteurs. Si on prend le modèle du Producteur-Consommateur de la figure 1 : donner les attributs et les méthodes de ces 2 classes, les interactions nécessaires pour assurer l'indépendance de ces 2 acteurs. Écrire le code c++ correspondant. L'Application ne sert qu'à déclencher la consommation.

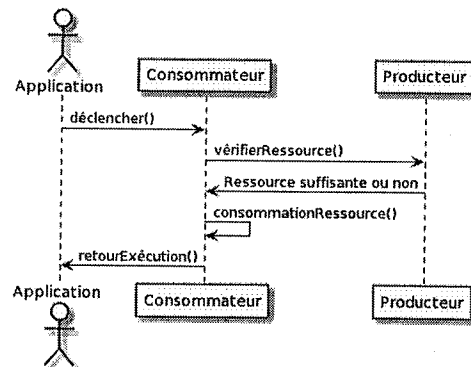


Figure 1: Producteur-consommateur

2. pour la **Boulangerie**, nous utiliserons l'IHM de la figure 2. Décrire sa mise en place avec QT, en donnant les types de widgets utilisés et leurs agencements (en donnant le code correspondant). La zone centrale est prévue pour la remontée d'informations à l'utilisateur (texte) : le résultat de la vérification des ingrédients, le lancement de la fabrication. On doit pouvoir modifier les quantités d'ingrédients avant de demander la vérification, on ne devrait également pas pouvoir lancer la fabrication sans avoir fait la vérification.

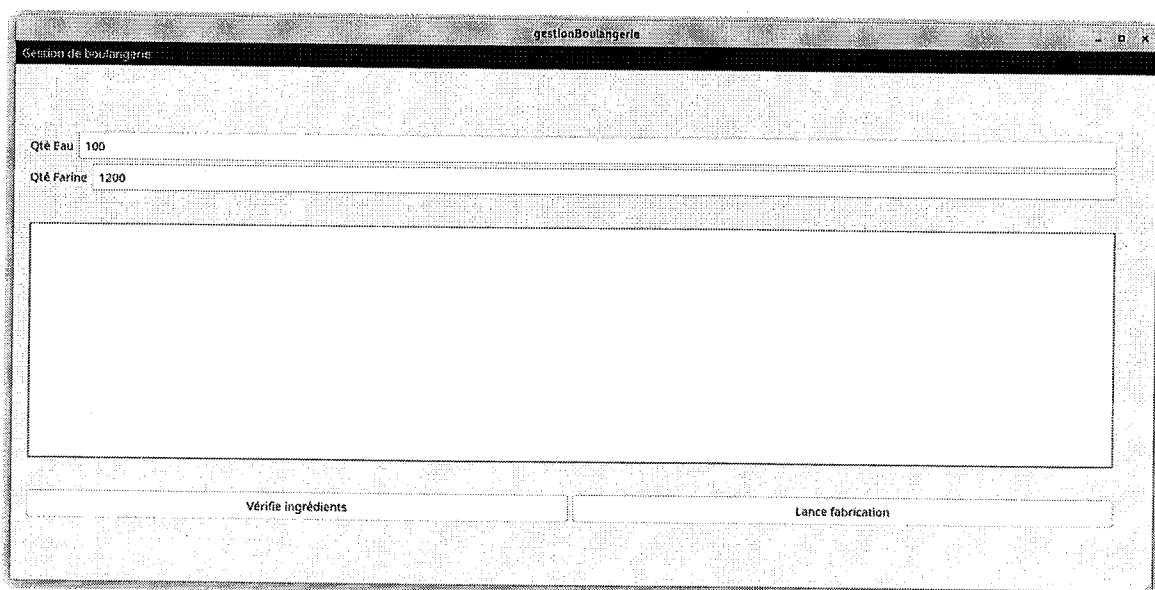


Figure 2: IHM

3. pour l'événement « clic » sur le bouton « Vérifie ingrédients » donner le déroulement des actions, les différentes classes et méthodes associées, les attributs stockés et les codes correspondants.