

5. Développez la méthode permettant de vérifier l'unicité d'une question donnée ainsi que son emplacement si elle existe.

Développez la méthode permettant l'ajout d'une question de classement QC.

6. Développez la méthode permettant de calculer le score total du questionnaire. Le score du questionnaire est la somme des scores de toutes les questions dans le questionnaire.

8. Développez la méthode permettant la suppression de toutes les questions pour une réponse (0) tout en développant une méthode de suppression de l'opérateur - dans la classe QCIS.

9. Enregistrez dans un fichier texte les attributs suivants : le texte de la question et la réponse le classement correct(e) de toutes les questions dans le questionnaire. Lavez une exception si l'ouverture du fichier a échoué.

10. Ajoutez uniquement l'entête des méthodes permettant la redéfinition du destructeur, du constructeur de copie et de l'opérateur d'affectation dans la bonne classe.

11. Le code suivant présente la redéfinition de l'opérateur d'affectation. Complétez le code suivant :

```

.....& ..... ::operator=(const ..... &Q)
{
    if(&Q!=this)
    {
        for(.....<Question*>::iterator j= ensQuestions.begin();j != ensQuestions.end();j++)
        {
            delete (*j);
        }
        Question*p;
        for(.....<.....*>::const_iterator i=Q. ensQuestions.begin(); i != Q.
ensQuestions.end();i++)
        {
            if(typeid(**i)==typeid(.....)){
                p=new .....(**i);
            }else
            {
                p=new .....(static_cast<const .....>(**i));
            }
            ensQuestions.push_back(p);
        }
    }
    return* this;
}

```





3. Complétez la classe QCU avec les attributs et les méthodes que vous jugez nécessaires.

QCU.h

class

{

public:

QCU();

virtual ~QCU();

protected:

private:

};

QCU.cpp



QC.h  
class.....

class.

public:  
QC 0;  
virtual ~ QC 0;

protected:

private:

NE RIEN ECRIRE

protected:

private:

};

Question.cpp



```
Question(){};
virtual ~Question(){};
```

**NB :**

1. Il est à noter que les attributs des classes à définir ne doivent pas être publics.
2. Utiliser les noms des attributs fournis dans l'énoncé.
3. L'utilisation du polymorphisme est obligatoire.
4. Appliquer la syntaxe du langage C++ enseignée lors des séances de cours.

**Enoncé :**

On désire développer une application pour gérer les réponses à un questionnaire.

Une question est caractérisée par un intitulé (identifiant unique), le texte et le score initialement = 0.

Une question de classement (QC) est une question caractérisée en plus par le classement envoyé et le classement correct et le nombre d'élément à classer. Si le classement envoyé et le classement correct sont identiques, le score est égal à 2. (N.B. le classement est une suite de chiffres par exemple 2314).

Une question à choix unique (QCU) est une question qui est caractérisée en plus par le numéro de la réponse envoyée et le numéro de la réponse correcte. Si la réponse envoyée et la réponse correcte sont identiques, le score est égal à 1.

**NB :** Chaque classe doit implémenter une méthode permettant de calculer le score.

La classe **Questionnaire** est caractérisée par son identifiant unique et un ensemble de questions. Les questions du questionnaire doivent être des QCU ou des QC.

Cette classe doit offrir les fonctionnalités suivantes :

- Chercher une question donnée.
- Ajouter une question de classement (QC).
- Ajouter une QCU (cette méthode ne va pas être implémentée)
- Calculer le score final du questionnaire.
- Afficher toutes les questions dans le questionnaire ainsi que le score de chacune.
- Supprimer les questions de type « QCU » dont la réponse est fausse en surchargeant l'opérateur - dans la classe QCU.
- Enregistrez dans un fichier texte les informations suivantes : l'identifiant, le texte de la question, ainsi que la réponse ou le classement correct. En cas d'échec lors de l'ouverture du fichier, la méthode doit déclencher une exception.