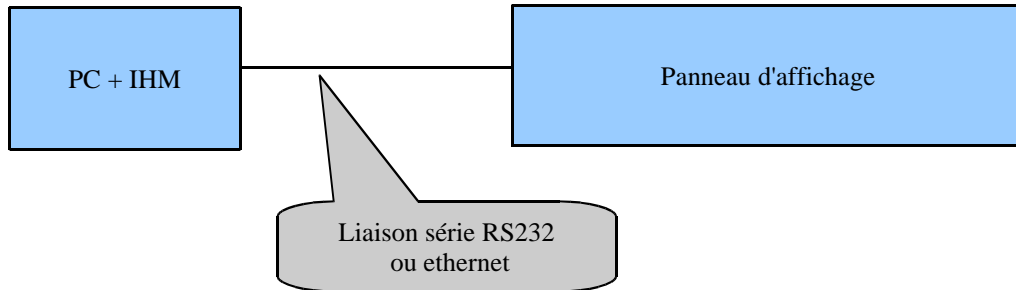


1. Présentation :

Un centre sportif désire équiper son stade d'un panneau d'affichage des scores de différentes rencontres.

Ce panneau doit être commandé à distance via un poste central à partir duquel on doit pouvoir choisir le nom des équipes et modifier le score de chacune d'elle.



L'objectif est d'écrire les classes nécessaires à l'application résidant sur le PC, de les tester puis de construire une application graphique intégrant les classes.

2. Analyse :

Lors d'une rencontre, chaque équipe possède un score qui doit pouvoir évoluer en fonction du sport. Chaque équipe doit pouvoir être nommée. On peut en déduire le diagramme des classes suivant :



L'application devra gérer deux objets de type **Score** possédant comme données membres :

- => le nom de l'équipe
- => le score cumulé de l'équipe

En déduire les services (actions) que doit offrir cette classe !:

- => augmenter le score d'une quantité x
- => retourner les membres nom et total
- => afficher le score (pour vérification)
- => un constructeur qui initialise le nom de l'équipe passé en paramètre et le total à 0.

3. Travail à réaliser :

- 3.1 Écrire la classe **Score** (déclaration et définition des méthodes (services).
- 3.2 Écrire un programme de test de la classe réalisant l'algorithme suivant :

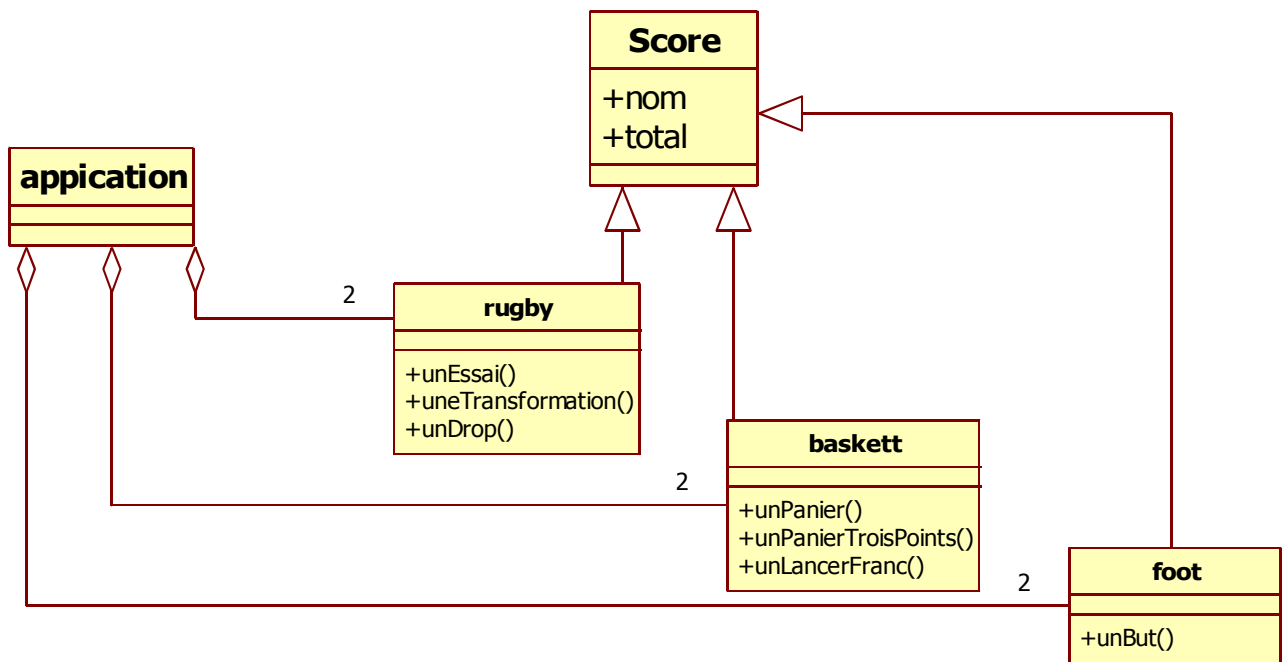
```
instancier deux objets Score avec deux noms différents
créer une variable entière choix
afficher le score de chacune d'elle
    TANT QUE choix != 0
        saisir choix
        si choix = 1
            augmenter le total de l'équipe 1
        si non si choix = 2
            augmenter le total de l'équipe 2
        si non sortir de la boucle
        afficher le score de l'équipe 1
        afficher le score de l'équipe 2
    FIN DE TANT QUE
```

4. Mise en œuvre de l'héritage.

La modification du score d'une équipe dépend du sport pratiqué et du type de marque. C'est pour cela que l'on va écrire des classes spécifiques pour les sports suivants avec les marques associées :

rugby		
	essai	total = total +5
	transformation	total = total +2
	Drop ou pénalité	total = total +3
foot		
	but	total = total +1
baskett		
	panier	total = total + 2
	Panier à trois points	total = total + 3
	Lancer franc	total = total + 1

Ces classe ayant en commun les données et services écrits dans la classe Score, elles en hériteront. On en déduit le nouveau diagramme des classes :



4.1 Travail à réaliser :

- 4.1.1 Écrire les classes **rugby**, **baskett**, **foot** (déclaration et définition des méthodes) héritant de **Score**.
- 4.1.2 Tester les classes avec trois programmes différents pour les trois sports.

5. Mise en œuvre dans un environnement graphique.

En utilisant C++Builder6, intégrer les classe écrites dans une IHM permettant de :

- => **choisir** le sport (**rugby**, **foot** ou **baskett**)
- => **nommer** chacune des deux équipes (instanciation des deux objets)
- => **afficher** le score de chacune des deux équipes
- => **modifier** le score d'une équipe conformément à la marque.

6. Analyse et critique de l'application.