

TD2 – Les classes UML en C++

L'objectif de la séance est de pouvoir, à partir d'un diagramme de classe UML, déclarer l'interface, définir l'implémentation d'une classe en C++ et réciproquement.

1. Tagada, tagada

Soit la classe Dalton représentée par le diagramme d'analyse ci-dessous :

- Proposez un diagramme de classe de conception qui respecte les règles de nommage en vigueur et le formalisme UML (voir ex.3).
- Donnez la déclaration de la classe Dalton (voir ex.2).
- Proposez les diagrammes d'objets correspondants aux instances des 4 frères Dalton tels qu'ils apparaissent dans l'album *Hors-la-loi* de Morris & Goscinny en 1954. (voir ANNEXE)

Joe (1,40m) William (1,68m) Jack (1,93m) Averell (2,13m)



Station de métro Janson à Charleroi.

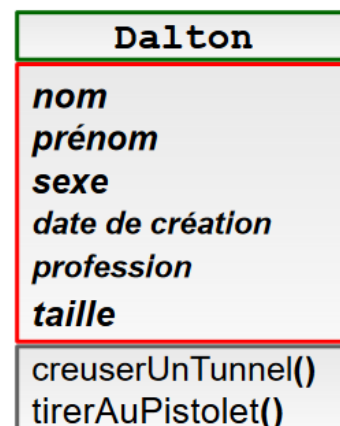


Diagramme d'analyse

2. Fil d'exécution

Soit le code suivant :

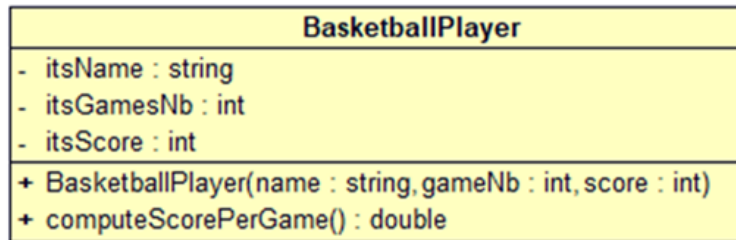
```
5  class Rectangle
6  {
7      private :
8          double itsLength;
9          double itsWidth;
10
11     public :
12         Rectangle(double length, double width);
13         double computePerimeter();
14         double computeArea();
15     };
16
17     Rectangle::Rectangle(double length, double width)
18     {
19         itsLength = length;
20         itsWidth = width;
21     }
22
23     double Rectangle::computePerimeter()
24     {
25         return 2*(itsLength+itsWidth);
26     }
27
28     double Rectangle::computeArea()
29     {
30         return itsLength*itsWidth;
31     }
32
33
34     int main()
35     {
36         Rectangle rect1(2, 3);
37         cout << rect1.computeArea() << endl ;
38
39         Rectangle* rect2 = new Rectangle(3, 4);
40         cout << rect2->computePerimeter() << endl ;
41         delete rect2;
42         return 0;
43     }
```

- Donnez la sortie écran attendue.
- Donnez le diagramme de classe d'implémentation de la classe Rectangle.
- Proposez les diagrammes des objets rect1 et rect2.



3. Tony

On souhaite fabriquer un système d'information permettant de traiter les statistiques de joueurs de basket. Soit le diagramme de classes de conception suivant :



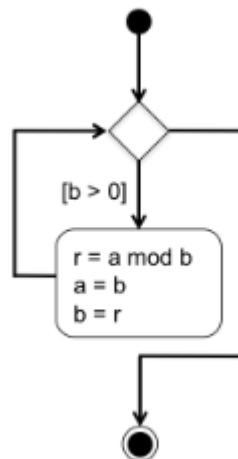
- Écrivez la déclaration de la classe BasketballPlayer.
- Écrivez la définition du constructeur paramétrique.
- Écrivez la définition de la méthode computeScorePerGame() qui calcule la moyenne des points marqués pour un joueur par match et retourne le résultat.

Lorsque Tony Parker a joué son 1100ème match NBA, face aux Houston Rockets en 2016, il avait alors marqué 18083 points.

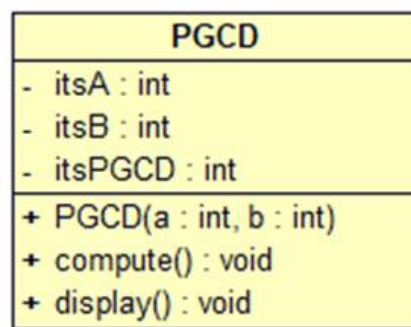
- Créez l'objet « tony : BasketballPlayer » qui représente les statistiques de Tony Parker (Lewis Hamilton) puis affichez sa moyenne.

4. PGCD

L'algorithme d'Euclide est un algorithme pour calculer le plus grand commun diviseur (PGCD) de deux entiers positifs a et b. Soit r le reste de la division euclidienne de a par b avec $a \geq b > 0$. Le PGCD est égal au dernier reste non nul. L'algorithme peut être programmé comme illustré dans le diagramme d'activité UML ci-dessous.



Soit le diagramme de classe d'implémentation de la classe PGCD suivant :



- Donnez la déclaration de la classe en C++.
- Écrivez le constructeur. Vous vérifierez les conditions de l'algorithme en supprimant le signe et en permutant les valeurs le cas échéant. Par exemple $\text{PGCD}(21, -30) = \text{PGCD}(30, 21) = 3$.
- Écrivez la méthode d'affichage qui donnera « $\text{PGCD}(30, 21) = 3$ ».
- Proposez un programme pour vérifier le bon fonctionnement de votre classe.

ANNEXE

Diagramme de classes

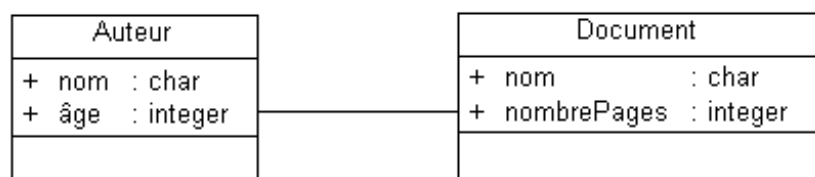


Diagramme d'objets

