Cours de C++, 1ère année, HE-Arc

Serie 3.1: Surcharge des opérateurs

Exercice 1: Classe `Time` Surcharge

Reprenez la classe `Time` de l'exercice 2 de la série 2.1

1. Surchargez l'opérateur d'insertion de flux `operator<<()` afin de pouvoir afficher un temps `cout << t << endl;`

2. Surchargez l'operateur d'affectation `operator=()`

3. Surchargez les opérateurs arithmétiques:

    1. L'opérateur `+` doit être implémenté par une fonction non membre de la classe et amie.
    2. L'opérateur `-` doit être implémenté par une fonction membre de la classe Est-ce que toutes les utilisations proposées dans le `main()` sont possibles ? Laquelle de ces deux opération est commutative ?

4. Surchargez les opérateurs relationnels

    1. De manière explicite, implémentez:
        1. L' `operator==()` avec des opérateurs logiques
        2. L' `operator<()` en comparant la valeur renvoyée par une méthode privée `evaluate()` qui retourne heure*60+minute.
    2. En vous basant sur ces deux opérateurs, implémentez:
        1. `operator>=()`
        2. `operator<=()`
        3. `operator!=()`

5. Surchargez les méthodes d'incrémentation préfixée et postfixée

6. Implémentez l'opérateur d'intertion de flux, capable de lire un temps sous la forme: 4:45

```
@startuml
    skinparam classAttributeIconSize 0
    class Time {
        - hour : short
        - minute : short

    + Time()
    + Time(Int, Int)
    + Time(Int)

    + operator = ( const Time& ) : Time&

    <<friend>> operator + (const Time&, const Time&) : Time
    + operator + (const Time&): Time
   - evaluate() : Short

    + operator == ( const Time& ) : Boolean {query}
    + operator <  ( const Time& ) : Boolean {query}
    + operator >= ( const Time& ) : Boolean {query}
    + operator <= ( const Time& ) : Boolean {query}
    + operator != ( const Time& ) : Boolean {query}

    + operator ++ ()    : Time&
    + operator ++ (Int) : Time

    <<friend>> operator<<(ostream&,  const Time&): ostream&
    <<friend>> operator>>(istream&,  Time&) : istream&

    }
@enduml
```

Exemple `main()`

```
#include <iostream>
#include "Time.h"

using namespace std;

int main()
{
     // INSTANTIATION DES OBJETS t1, t2, t3
     //=================================================================
     cout << "TEST DES CONSTRUCTEURS :" << endl << endl;

     cout << "Time t1; ";
     Time t1;   // Appel du constructeur par défaut
     t1.show(); //-> t1: 12H00
     cout << "Time t2(10,9); ";
     Time t2(10, 9); // Appel du constructeur standard
     t2.show();      //-> t2: 10H09
     cout << "Time t3(17.75); ";
     Time t3(17.75); // Appel du constructeur de conversion
     t3.show();      //-> t3: 17:45

     // TEST DES MODIFICATEURS
     //=================================================================
     cout << endl << "TEST DES MODIFICATEURS :\n\n";
     cout << "t2.setHour(7); ";
     t2.setHour(7);
     t2.show(); //-> t2: 07:09

     cout << "t2.setMinute(-40); ";
     t2.setMinute(-40); // -40 < 0 --> 0
     t2.show();         //-> t2: 07:00

     cout << "t2.setMinute(86); ";
```

```
    t2.setMinute(86); // 86 %60 --> 26  et h+1
    t2.show();        //-> t2: 08:26

    cout <<"operateur <<" << endl;
    cout <<"================================================================="
<<endl;
    cout << "cout << \"t1: \" << t1 << \" t2: \" << t2 << \" t3 \" << t3 <<
endl;" << endl;;
    cout << "t1: " << t1 << " t2: " << t2 << " t3 " << t3 << endl << endl;

    cout <<"operateur =" << endl;
    cout <<"================================================================="
<<endl;
    cout << "t1: " << t1 << " t2: " << t2 << endl;
    cout << "t1 =t2" << endl;
    t1=t2;
    cout << "t1: " << t1 << " t2: " << t2 << endl << endl;

    cout <<"operateur + (fonction non membre amie)" << endl;
    cout <<"================================================================="
<<endl;
    t1.setMinute(0);
    cout << "t1: " << t1 << " t3: " << t3 << endl;
    cout << "t1 = t1 + t3" << endl;
    t1 = t1 + t3;
    cout << "t1: " << t1 << " t3: " << t3 << endl << endl;

    cout << "t3: " << t3  << endl;
    cout << "t3 = t3 + 4" << endl;
    t3 = t3 + 4;
    cout << "t3: " << t3  << endl << endl;

    cout << "t3: " << t3  << endl;
    cout << "t3 = 4 + t3" << endl;
    t3 = 4 + t3;
    cout << "t3: " << t3  << endl << endl;

    cout <<"operateur - (fonction membre)" <<endl;
    cout <<"================================================================="
<<endl;

    cout << "t1: " << t1 << " t3: " << t3 << endl;
    cout << "t1 = t1 - t3" << endl;
    t1 = t1 - t3;
    cout << "t1: " << t1 << " t3: " << t3 << endl << endl;

    cout << "t3: " << t3 << endl;
    cout << "t3 = t3 - 1" << endl;
    t3 = t3 - 1;
    cout << "t3: " << t3 << endl<< endl;

    cout << "t3: " << t3 << endl;
    cout << "t3 = 1 - t3 " << endl;
    t3 = 1 - t3; not allowed
```

```
     cout << "t3: " << t3 << endl << endl;

    cout <<" == operator overloading" << endl;
    cout <<"================================================================="
<<endl;
    std::cout << t1 << " == " << t1 << boolalpha << " : " << (t1==t1) <<
std::endl;
    std::cout << t1 << " == " << t2 << boolalpha << " : " << (t1==t2) <<
std::endl;
    // < operator overloading
    std::cout << t1 << " <  " << t1 << boolalpha << " : " << (t1<t1)  <<
std::endl;
    std::cout << t1 << " <  " << t2 << boolalpha << " : " << (t1<t2)  <<
std::endl;
    // >= operator overloading
    std::cout << t1 << " >= " << t1 << boolalpha << " : " << (t1>=t1)  <<
std::endl;
    std::cout << t1 << " >= " << t2 << boolalpha << " : " << (t1>=t2)  <<
std::endl;
    // != operator overloading
    std::cout << t1 << " != " << t1 << boolalpha << " : " << (t1!=t1)  <<
std::endl;
    std::cout << t1 << " != " << t2 << boolalpha << " : " << (t1!=t2)  <<
std::endl << std::endl;


    std::cout << "\n ++ Increment operator \n";
    cout <<"================================================================="
<<endl;

    std::cout << "t2: " << t2 << " t3: " << t3 << std::endl;
    std::cout << "t2 = t3++;" << std::endl;
    t2 = t3++;
    std::cout << "t2: " << t2 << " t3: " << t3 << std::endl;
    std::cout << "t2 =  ++t3;" << std::endl;
    t2 =  ++t3;
    std::cout << "t2: " << t2 << " t3: " << t3 << std::endl;


    cout << "Entrez un temps au format hh:mm :";
    cin >> t2;
    std::cout << "t2: " << t2 << std::endl;

     cout << "\n\nPlease hit ENTER to continue... ";
     cin.get();
     return 0;
}
```