

Examen de Programmation C++

Durée : 1h

Aucun document autorisé

Mobiles interdits

Nom :

Prénom :

Dans ce DS, on souhaite représenter un jeu de 52 cartes à jouer.

Les cartes

Tout d'abord, vous allez définir la classe `carte`. On définit les types énumérés suivants :

```
enum Couleur { trèfle, carreau, coeur, pique };
enum Valeur { deux, trois, quatre, cinq, six, sept,
             huit, neuf, dix, valet, dame, roi, as };
```

- 1. Écrivez la classe `carte` avec :
- deux variables membres privées, `couleur` et `valeur` (chacune du type énuméré qui correspond);
 - un constructeur pour initialiser les deux variables membres;
 - la surcharge de l'opérateur `<<` pour écrire une carte dans un `ostream`. Vous écrirez une méthode `toString`. Rappel, les identificateurs de constantes d'un type énuméré désignent des constantes entières. Pensez à leur conversion sous forme de chaîne de caractères.

Fichier `carte.hpp` :

```
#pragma once
#include <iostream>
#include <sstream>

class carte {
private:
    static const std::string couleur2String[];
    static const std::string valeur2String[];
    enum Couleur couleur;
    enum Valeur valeur;
public:
    carte(enum Valeur v, enum Couleur c) : valeur(v), couleur(c) {}
    std::string toString() const;
    friend std::ostream& operator<<(std::ostream &f, const carte &c);
};
```

Fichier `carte.cpp` :

```
#include <string>
#include "carte.hpp"

const std::string carte::couleur2String[] =
{
    "trèfle", "carreau", "coeur", "pique"
};
const std::string carte::valeur2String[] =
{
    "deux", "trois", "quatre", "cinq",
    "six", "sept", "huit", "neuf", "dix",
    "valet", "dame", "roi", "as"
};
std::string carte::toString() const {
    return "(" + valeur2String[this->valeur] + ", "
        + couleur2String[this->couleur] + ")";
}
std::ostream& operator<<(std::ostream &f, const carte &c) {
    return f << c.toString();
}

.....
```

Le jeu

- 2. Vous allez maintenant définir la classe `jeu52`. Écrivez cette classe qui possède :
- une variable privée `lesCartes` de type `vector` dont les éléments sont des pointeurs sur `carte`;
 - un constructeur qui initialise le vecteur avec les 52 cartes différentes;
 - la surcharge de l'opérateur `<<` pour écrire toutes les cartes du jeu dans un `ostream`. Vous écrirez une méthode `toString`.

Fichier `jeu52.hpp` :

```
#pragma once

#include <vector>
#include "carte.hpp"

class jeu52 {
private:
    std::vector<carte*> lesCartes;
public:
    jeu52() {
        for (enum Valeur v=deux; v<=as; v = (enum Valeur) ((int) v+1))
            for (enum Couleur c=trèfle; c<=pique; c = (enum Couleur) ((int) c+1))
                this->lesCartes.push_back(new carte(v, c));
    }
    std::string toString() const;
    friend std::ostream& operator<<(std::ostream &f, const jeu52 &j);
};
```

Fichier `jeu52.cpp` :

```
#include <sstream>
#include "jeu52.hpp"

std::string jeu52::toString() const {
    std::string s = "[ ";
    for (carte *c : this->lesCartes)
        s+= c->toString();
    return s+" ]";
}

std::ostream& operator<<(std::ostream &f, const jeu52 &j) {
    return f << j.toString();
}
```

.....

- 3. Expliquez s'il est nécessaire de redéfinir le destructeur de la classe `jeu52`. Si oui, donnez sa programmation.

Il est nécessaire de programmer le destructeur, car les cartes sont allouées dynamiquement et, par défaut, ne sont pas désallouées à destruction d'un objet `jeu52`.

```
~jeu52() {
    for (carte *c : this->lesCartes)
        delete c;
}
```

.....

main

- 4. Écrivez la méthode `main` qui crée la carte *as de trèfle* et un jeu complet de 52 cartes, puis les affiche sur la sortie standard comme suit :

```
(as,trèfle)
-----
[ (deux,trèfle)(deux,carreau)(deux,coeur)(trois,pique)(trois,trèfle)
  (trois,carreau)(trois,coeur)(trois,pique)(quatre,trèfle)
  (quatre,carreau)(quatre,coeur)(quatre,pique)(cinq,trèfle)
  (cinq,carreau)(cinq,coeur)(cinq,pique)(six,trèfle)(six,carreau)
  (six,coeur)(six,pique)(sept,trèfle)(sept,carreau)(sept,coeur)
  (sept,pique)(huit,trèfle)(huit,carreau)(huit,coeur)(huit,pique)
  (neuf,trèfle)(neuf,carreau)(neuf,coeur)(neuf,pique)(dix,trèfle)
  (dix,carreau)(dix,coeur)(dix,pique)(valet,trèfle)(valet,carreau)
  (valet,coeur)(valet,pique)(dame,trèfle)(dame,carreau)(dame,coeur)
  (dame,pique)(roi,trèfle)(roi,carreau)(roi,coeur)(roi,pique)
  (as,trèfle)(as,carreau)(as,coeur)(as,pique) ]
```

```
int main() {
    // la carte as de trèfle
    carte c(as, trefle);
    std::cout << c << std::endl;
    std::cout << "-----" << std::endl;
    // le jeu de 52 cartes
    jeu52 jeu;
    std::cout << jeu << std::endl;
    return EXIT_SUCCESS;
}
```

.....