Cours MOOC EPFL d'introduction à la programmation orientée objet, illustré en C++

Second assignment
Constructors

J.-C. Chappelier & J. Sam

1 Exercise 1 — Green mice

The goal of this exercise is to create « mice » in different ways and to make them « evolve » over time.

1.1 Description

Download the source code available at the course webpage and complete it according to the instructions below.

WARNING : you should modify neither the beginning nor the end of the provided file. It's thus mandatory to proceed as follows :

1. save the downloaded file as `labo.cc` or `labo.cpp` ;
2. write your code between these two provided comments :
```
/*******************************************
 * Complétez le programme à partir d'ici.
 *******************************************/

/*******************************************
 * Ne rien modifier après cette ligne.
 *******************************************/
```
3. save and test your program to be sure that it works properly ; try for instance the values used in the example given below ;
4. upload the modified file (still named `labo.cc` or `labo.cpp`) in "OUTPUT submission" (not in "Additional" !).

The provided code does the following :

- « constructs » mice ;
- makes them evolve using a method `evolue` (means "evolve" in French) ;
- displays the mice before and after making them evolve.

The body of the class `Souris` (means "Mouse" in French) is missing and this is what we ask you to write.

A mouse is characterized by its weight in grams (a `double`), its color (`String`), its age in months (an `int`), its life expectancy (an `int`) and one indication about whether it is cloned or not (a boolean).

These attributes will be named respectively : `poids`, `age`, `couleur`, `esperance_vie` and `clonee`. The public methods of the class `Souris` are :

- constructors that conform to the provided `main` method, with the following order for the parameters : the weight, the color, the age and the life expectancy. These last two parameters have as default value zero and 36 respectively. The constructors display the message `Une nouvelle souris !` ; (means `A new mouse !`)

- a copy constructor which should display the message `Clonage d'une souris !` ; a cloned mouse has the same characteristics as the original mouse, *except* for its life expectancy which is less : the 4 fifths of the original mouse's life expectancy ;

- a destructor which should display the message `Fin d'une souris...` ;

- a method `afficher()` (means "display") which should display the characteristics of a `Souris` respecting ***strictly*** the following format :
  `Une souris <couleur>[, clonee,] de <age> mois et pesant <poids> grammes` (in only one line)
  where `<age>` has to be replaced by the age of the mouse and `<poids>` by its weight. The part of the sentence « `, clonee,` » won't be displayed unless the mouse has been cloned ;

- a method `vieillir` (means "aging") which will increase the age of the mouse by one unit. If the mouse is cloned, it should become green if it reaches an age superior to the half of its life expectancy ; even if it is not called explicitly in the `main()`, this method must be public ; it will be tested ;

- and a method `evolue` (means "evolves") aging a mouse from its current age up to its life expectancy.

Use the terminal for all the required displays and terminate them with an End Of Line. An execution example is provided below.

1.2 Execution example

```
Une nouvelle souris !
Une nouvelle souris !
Clonage d'une souris !
Une souris blanche de 2 mois et pesant 50 grammes
Une souris grise de 0 mois et pesant 45 grammes
Une souris grise, clonee, de 0 mois et pesant 45 grammes
Une souris blanche de 36 mois et pesant 50 grammes
Une souris grise de 36 mois et pesant 45 grammes
Une souris verte, clonee, de 28 mois et pesant 45 grammes
Fin d'une souris...
Fin d'une souris...
Fin d'une souris...
```

2 Exercise 2 — Library

The purpose of this exercise is to simulate in a very basic manner the management of a library. The library contains *exemplaires* (copies) of *œuvres* (books) written by *auteurs* (authors). It consists of modeling each one of these elements in your program.

2.1 Description

Download the source code available at the course webpage and complete it according to the instructions below.

WARNING : you should modify neither the beginning nor the end of the provided file. It's thus mandatory to proceed as follows :

1. save the downloaded file as `biblio.cc` or `biblio.cpp` ;
2. write your code between these two provided comments :

   ```
   /*******************************************
    * Complétez le programme à partir d'ici.
    *******************************************/

   /*******************************************
    * Ne rien modifier après cette ligne.
    *******************************************/
   ```
3. save and test your program to be sure that it works properly ; try for instance the values used in the example given below ;

4. upload the modified file (still named `biblio.cc` or `biblio.cpp`) in “OUTPUT submission” (not in “Additional” !).

The provided code creates authors and books of these authors, stores in the library copies of these books, then :

- lists all the copies of this library ;
- lists all the copies written in english ;
- displays the names of all the awarded authors having written a book of which the library has a copy ;
- and displays the number of copies of a given book ;

A possible execution example is provided further below.

The definitions of the classes `Auteur` (means "Author"), `Oeuvre` (stands fo "Book"), `Exemplaire` (stands for "Copy of a book") and `Bibliotheque` (means "Library"), described here above, are missing and you are asked to provide them.

The `Auteur` class An `Auteur` is characterized by his name (a `String`) as well as an indication allowing to know whether he has been awarded.

The methods that are specific to this class and are part of its user interface are :

- constructors that conform to the provided `main` program, with the following order for the parameters : the name and the indication allowing to know whether the author has been awarded ;
- a method `getNom` returning the name of the author ;
- a method `getPrix` returning `true` if the author has been awarded.

The `Oeuvre` class An `Oeuvre` (means "literary work"/"book" in French) is characterized by its title (of type `string`), (a constant reference to the) author who wrote it and the language in which it was written (of type `string`).

The methods that are specific to this class and are part of its user interface are :

- constructors that conform to the provided `main`, with the following order for the parameters : the title, the author and the language ;
- a method `getTitre` returning the title of the book ;
- a method `getAuteur` returning *a constant reference to the* author (this type must used) ;
- a method `getLangue` returning the language of the book ;
- and a method `affiche` displaying the characteristics of the book respecting ***strictly*** the following format :
  `<titre>, <nom de l'auteur>, en <langue>`

where `<titre>` is to be replaced by the tile of the book, `<nom de l'auteur>`, by the name of the author and `<langue>` by its language ;

— a destructor displaying a message following ***strictly*** the following format :
`L'oeuvre "<titre>, <nom de l'auteur>, en <langue>" n'est plus disponible.`

We consider here that an `Oeuvre` cannot be copied.

See the execution example provided further below for displaying examples.

The `Exemplaire` class The class `Exemplaire` models a copy of a book. An instance of this class is characterized by a (reference to a) book of which it is a copy.

The methods that are specific to the class `Exemplaire` and that should be part of its user interface are :

— a constructor taking as argument a reference to an `Oeuvre` and displaying a message respecting ***strictly*** the following format :
`Nouvel exemplaire de : <titre>, <nom de l'auteur>, en <langue>`
followed by an End Of Line ;

— a copy constructor displaying a message respecting ***strictly*** the following format :
`Copie d'un exemplaire de : <titre>, <nom de l'auteur>, en <langue>` in one line, followed by an End Of Line ;

— a destructor displaying a message respecting ***strictly*** the following format :
`Un exemplaire de "<titre>, <nom de l'auteur>, en <langue>" a été jeté !` in one line, followed by an End Of Line ;

— a method `getOeuvre` returning a *constant reference* to the book (`Oeuvre`) ;

— and a method `affiche` displaying a description of the copy respecting ***strictly*** the following format :
`Exemplaire de : <titre>, <nom de l'auteur>, en <langue>`
without any End Of Line.

The `Bibliotheque` class A `Bibliotheque` (means "library" in French) is characterized by its name and contains a set of pointers to (`Exemplaire`). The set will be modeled using a `vector`.

The methods that are specific to the class `Bibliotheque` and are part of its user interface are :

— a constructor that conforms to the provided `main` method and displays the message :

`La bibliothèque <nom> est ouverte !` followed by an End Of Line, where `<nom>` is to be replaced by the name of the library ;

- a method `getNom` returning the name of the library ;
- a method `stocker` (means "store") allowing to store one or multiple copies (`Exemplaire`) of a book in the library ; it should conform to the provided `main`, with the following order for the parameters : the reference to a book and the number n of copies to add ; this method will add to the library's list of `Exemplaire`, n `Exemplaire` of the provided `Oeuvre`, which is to be constructed. If the number of copies is not provided, this means that it is 1 by default. **Attention, the copies should imperatively be added in the end of the vector** ;
- a method `lister_exemplaires` displaying all the copies of a book written in a given language ; if no language is given (empty string), all the copies of the library will be displayed ; the `Examplaire` must be displayed with their own display method ; they will be separated using End of Lines (see the execution example provided further below) ;
- a method `compter_exemplaires` returning the number of copies of a given book passed as a parameter ;
- a method `afficher_Auteur` with a parameter of type boolean (with `false` as default value) that displays the names of the author having a copy of one of their books stored in the library. If the boolean is equal to `true`, only the names of authors with a prize (awarded) will be displayed ; The name of the author will be displayed as many times as there are copies written by the author.
- a destructor displaying the following message :
  ```
  La bibliothèque <nom> ferme ses portes,
  et détruit ses exemplaires :
  ```
  where `<nom>` is the name of the library, and freeing the memory associated to the stores copies (`Exemplaire`).

2.2 Execution example

```
La bibliothèque municipale est ouverte !
Nouvel exemplaire de : Les Misérables, Victor Hugo, en français
Nouvel exemplaire de : Les Misérables, Victor Hugo, en français
Nouvel exemplaire de : L'Homme qui rit, Victor Hugo, en français
Nouvel exemplaire de : Le Comte de Monte-Cristo, Alexandre Dumas, en français
Nouvel exemplaire de : Le Comte de Monte-Cristo, Alexandre Dumas, en français
Nouvel exemplaire de : Le Comte de Monte-Cristo, Alexandre Dumas, en français
Nouvel exemplaire de : Zazie dans le métro, Raymond Queneau, en français
Nouvel exemplaire de : The Count of Monte-Cristo, Alexandre Dumas, en anglais
La bibliothèque municipale offre les exemplaires suivants :
Exemplaire de : Les Misérables, Victor Hugo, en français
Exemplaire de : Les Misérables, Victor Hugo, en français
```

```
Exemplaire de : L’Homme qui rit, Victor Hugo, en français
Exemplaire de : Le Comte de Monte-Cristo, Alexandre Dumas, en français
Exemplaire de : Le Comte de Monte-Cristo, Alexandre Dumas, en français
Exemplaire de : Le Comte de Monte-Cristo, Alexandre Dumas, en français
Exemplaire de : Zazie dans le métro, Raymond Queneau, en français
Exemplaire de : The Count of Monte-Cristo, Alexandre Dumas, en anglais
 Les exemplaires en anglais sont :
Exemplaire de : The Count of Monte-Cristo, Alexandre Dumas, en anglais
 Les auteurs à succès sont :
Raymond Queneau
 Il y a 3 exemplaires de Le Comte de Monte-Cristo
La bibliothèque municipale ferme ses portes,
et détruit ses exemplaires :
Un exemplaire de "Les Misérables, Victor Hugo, en français" a été jeté !
Un exemplaire de "Les Misérables, Victor Hugo, en français" a été jeté !
Un exemplaire de "L’Homme qui rit, Victor Hugo, en français" a été jeté !
Un exemplaire de "Le Comte de Monte-Cristo, Alexandre Dumas, en français" a été jeté !
Un exemplaire de "Le Comte de Monte-Cristo, Alexandre Dumas, en français" a été jeté !
Un exemplaire de "Le Comte de Monte-Cristo, Alexandre Dumas, en français" a été jeté !
Un exemplaire de "Zazie dans le métro, Raymond Queneau, en français" a été jeté !
Un exemplaire de "The Count of Monte-Cristo, Alexandre Dumas, en anglais" a été jeté !
L’oeuvre "The Count of Monte-Cristo, Alexandre Dumas, en anglais" n’est plus disponible.
L’oeuvre "Zazie dans le métro, Raymond Queneau, en français" n’est plus disponible.
L’oeuvre "Le Comte de Monte-Cristo, Alexandre Dumas, en français" n’est plus disponible.
L’oeuvre "L’Homme qui rit, Victor Hugo, en français" n’est plus disponible.
L’oeuvre "Les Misérables, Victor Hugo, en français" n’est plus disponible.
```