Assignemnt 5

Programming III in C/C++

Computer Science

Implement a small Library Management System.

The Library management system should, in principle, be able to manage various types of references to media, e.g. on books, articles from journals, articles from anthologies, CD's, etc …

- For each medium, basic data is stored: a unique abbreviation, the author, a title and the year of publication.
- In addition to a book, the publisher is saved.
- In addition to the basic data, the magazine's name, magazine number, and page number are saved in case of a journal article.

For this you should develop a base class „`Medium`“, from which you derive the different classes for different media (`book, article, CD, . . .`).

Main responsibility of class „`Medium`“ is to manage the attributes common to all media (`unique abbreviation, author, title, year of publication`). Besides that, the class provides all methods, that are required anyway (`constructors, getter, setter, ...`). In addition, the class provides a method

```
string toString() const;
```

This method provides a string representation of a medium in the usual form:

```
[abbrev] Author: Title
```

In the derived classes, this method must be overridden accordingly.

An article should be displayed in the form

> [Pa72] D. L. Parnas: On the Criteria To Be Used in Decomposing Systems into Modules. Communications of the ACM, 15, 1972, 1053.

while a book is to be displayed in the form:

> [St13] B. Stroustrup: The C++ Programming Language (4th ed.) .

Addison-Wesley, 2013.

Now, please implement a class „`bibliography`“ to manage bibliographic or media information of different types. This class provides the following interface:

- a public constructor

  ```
  Bibligraphy( int size );
  ```

  With `size` it is possible to control, how many entries may be managed.
- A public method `insert`. This method allows to add a new medium to the bibliography.

  Pay attention to meaningful error handling!
- The operator function `operator[]` should be overloaded in such a way that the associated medium can be accessed by means of the short name. A statement like

  ```
  myBibliography["Pa72"]
  ```

  shall return a pointer to the corresponding medium (in the example a pointer to the object of class article with abbreviation “Pa72”)

  If the abbreviation does not exist, a NULL-Pointer should be returned.
- The operator function `operator <<` should be overloaded in such a way that a call like

  ```
  cout << myBibliography;
  ```

  results in the output of the whole bibliography. Make sure that the output of a medium is type-specific as described in the `toString()` method.

An application example is given in the program section below.

Additionally, test your class with unit tests.

```
#include <iostream>
#include <stdlib.h>

#include "Bibliography.h"
```

```
#include "Book.h"
#include "Article.h"

using namespace std;

void testBibliography()
{
   // create bibliography
   Bibliography myBibliography(10);

   Book* b1 = new Book("Ra01",
                       "Dietmar Ratz",
                       "Grundkurs Programmieren in Java",
                       2001,
                       "Hanser-Verlag");
   Book* b2 = new Book("Kr03",
                       "Guido Krueger",
                       "Handbuch der Java-Programmierung",
                       2003,
                       "Addison-Wesley");
   Article* a1 = new Article("Pa72",
                         "D. L. Parnas",
                  "On the Criteria To Be Used in Decomposing Systems into
",
                         1972,
                         "Communications of the ACM",
                         15,
                         1053);
   try {
      myBibliography.insert(b1);
      myBibliography.insert(b2);
      myBibliography.insert(a1);

      cout << "Bibliography:" << endl << myBibliography; //display bibl.
      cout << endl << (*myBibliography["Pa72"]).toString() << endl;
   }
   catch (Error f){
        cout << endl << f.getMessage() << endl;
   }
}



int main(int argc, char *argv[])
{
   testBibliography();
   cin.sync();cin.get();
   return 0;
}
```