Student Name: 马越
Student ID: 2022141460097

C++ 面向对象程序设计
Assignment 5

鉴于Author类与Publisher类中方法与成员的相似性，首先实现一个基类。

Listing 1: base.hpp

```cpp
#ifndef BASE_HPP
#define BASE_HPP

#include <utility>
#include <vector>
#include <string>

typedef std::vector<std::vector<std::string>> Dataframe;

class Book;

class Base {
public:
  Base(std::string _name): name(std::move(_name)) {}
  [[maybe_unused]] void setListOfBooks(std::vector<Book*> all_books) {
    list_of_books = std::move(all_books);
  }
  std::string getName() const {
    return name;
  }
private:
  std::string name;
  std::vector<Book*> list_of_books;
};

#endif //BASE_HPP
```

在基类Base的基础上派生出Author类与Publisher类。

Listing 2: author.h

```cpp
#ifndef AUTHOR_H
#define AUTHOR_H

#include "base.hpp"

class Author: public Base {
  public:
  Author(std::string _name);
};

#endif //AUTHOR_H
```

Listing 3: author.cpp

```
#include "author.h"

Author::Author(std::string _name) : Base(std::move(_name)) { }
```

Publisher类与此类似。

再实现Book类，实现赋值运算符，以便后面调用std::sort()使用自定义比较函数进行排序。

Listing 4: book.h

```
#ifndef BOOK_H
#define BOOK_H

#include "author.h"
#include "publication.h"
#include <memory>
#include <utility>

class Book {
public:
  Book(std::string _title,
       std::string _genre,
       double      _price,
       Author&     _author,
       std::shared_ptr<Publisher> _publisher);

  Book& operator=(const Book& other);

  double getPrice() const;
  std::string getAuthorName() const;
  std::string getPublisherName() const;

private:
  std::string title;
  std::string genre;
  double price;

  Author& author;
  std::shared_ptr<Publisher> publisher;
};
```

```

#endif //BOOK_H
```

Listing 5: book.cpp

```cpp
#include "book.h"

Book::Book(std::string _title,
           std::string _genre,
           double _price,
           Author &_author,
           std::shared_ptr <Publisher> _publisher):
  title(std::move(_title)),
  genre(std::move(_genre)),
  price(_price),
  author(_author),
  publisher(std::move(_publisher)) { }

Book &Book::operator=(const Book &other) {
  if (this == &other)
    return *this;
  title = other.title;
  genre = other.genre;
  price = other.price;
  author = other.author;
  publisher = other.publisher;
  return *this;
}

double Book::getPrice() const {
  return price;
}

std::string Book::getAuthorName() const {
  return author.getName();
}

std::string Book::getPublisherName() const {
  return publisher->getName();
}
```

在实现defineBooks()函数时,由于创建Book对象中有Author的引用,故需要在函数外保存Author对象以延长引用对象的生命周期。

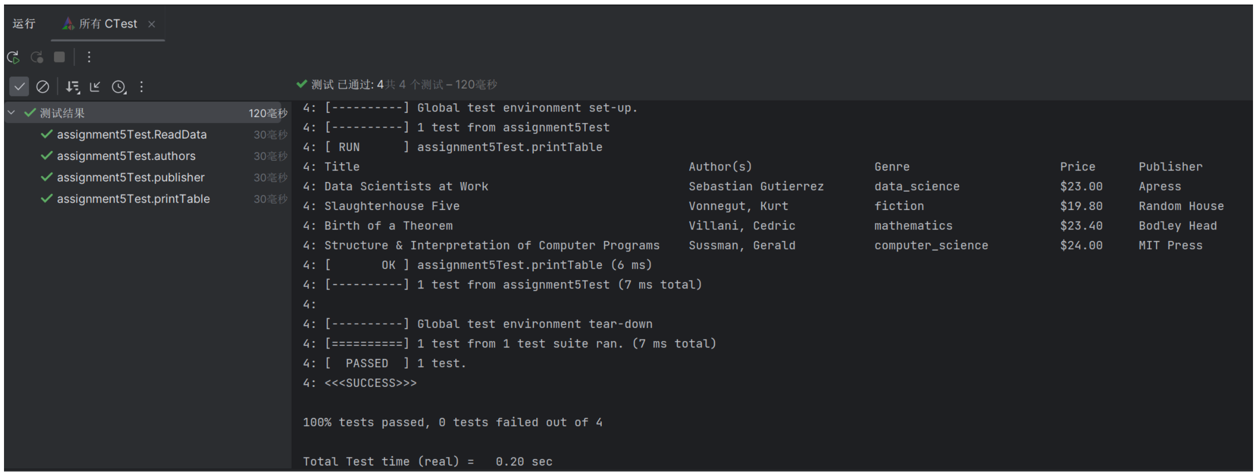
Listing 6: assignment5.cpp

```cpp
...
```

```cpp
static std::vector<Author*> authors;

std::vector<Book> defineBooks(Dataframe* Table) {
  std::vector<Book> books;
  for (auto row: *Table) {
    /// Title,Author,Genre,Price,Publisher
    std::string title = row[0];
    std::string authorName = row[1];
    std::string genre = row[2];
    int pos = row[3].find("$");
    double price = std::stod(row[3].substr(pos + 1));
    std::string publisherName = row[4];
    authors.push_back(new Author(authorName));
    books.emplace_back(
        title,
        genre,
        price,
        *authors.back(),
        std::make_shared<Publisher>(publisherName)
    );
  }
  return books;
}
...
```

在unittest.cpp中，修改文件路径为相对路径"../dataset.csv"。


运行
所有 CTest
测试 已通过: 4共 4 个测试 – 120毫秒
测试结果 120毫秒
assignment5Test.ReadData 30毫秒
assignment5Test.authors 30毫秒
assignment5Test.publisher 30毫秒
assignment5Test.printTable 30毫秒

```
4: [----------] Global test environment set-up.
4: [----------] 1 test from assignment5Test
4: [ RUN      ] assignment5Test.printTable
4: Title                                              Author(s)                 Genre                     Price      Publisher
4: Data Scientists at Work                            Sebastian Gutierrez       data_science              $23.00     Apress
4: Slaughterhouse Five                                Vonnegut, Kurt            fiction                   $19.80     Random House
4: Birth of a Theorem                                 Villani, Cedric           mathematics               $23.40     Bodley Head
4: Structure & Interpretation of Computer Programs    Sussman, Gerald           computer_science          $24.00     MIT Press
4: [       OK ] assignment5Test.printTable (6 ms)
4: [----------] 1 test from assignment5Test (7 ms total)
4:
4: [----------] Global test environment tear-down
4: [==========] 1 test from 1 test suite ran. (7 ms total)
4: [  PASSED  ] 1 test.
4: <<<SUCCESS>>>

100% tests passed, 0 tests failed out of 4

Total Test time (real) =   0.20 sec
```