姓名： 姜现日

学号：1820191086

实验日期：2024-04-17

实验任务3：

**实验内容**：Book代表书，有”Name(书名）“、“Author(作者)”、”Price(价格)“、”IsBorrowed(是否借出)”四个属性。

类Library代表图书馆，其内部字段books用于保存图书馆中所有的书，他的FindBook()方法依据书名查找同名的书（可能有多本）。另一个Get AllBooks()方法获取馆藏所有书的详细信息。

类Reader代表作者，Name字段代表其姓名，读者可以进行BorrowBook()（借书),ReturnBook():(还书)操作。

请完成以下工作：

1. 用Java编程实现以上三个类。
2. 在main()方法内书写以下测试代码：
3. 创建一个Library的类的实例，myLittleLibrary,其中预存有以下三本书：

Java程序设计，张三著，45元

Java核心设计，李四著，50元

Java程序设计，王五著，38元

1. 显示图书馆所有图书的信息，输出样例如下：

Java程序设计，张三著，45元，可借

Java核心设计，李四著，50元，可借

Java程序设计，王五著，38元，未还

1. 创建一个Reader类的实例oneBeautifulGril，她先在myLittleLibrary中查找《Java程序设计》；
2. 当oneBeautifulGril借了张三著的那一本书时，显示当前图书馆所有图书的信息；
3. oneBeautifulGril把书还了，再次显示图书馆中图书信息；
4. **实验思路:在FindBooks函数中，用foreach句子，检查ArrayList的每个元素。如果元素中有b.name 的值是等于BookName,在新创建的ArrayList上用add输入。在ArrayList用toArray函数换成为排列。**

**在Borrow 和Return函数，每次检查book的isBorrowed值，查看书是不是可借还是未还。**

**在main函数上，要输出图书馆的所有图书的信息，创建了printBooks函数。在printBooks函数中，从Library的FindBook函数获得的包含所有图书信息的排列用foreach句子查看每个图书。以后输出图书的各个信息。**

**（3）实验源码：**

import java.util.Arrays;  
import java.util.Objects;  
  
//TIP To <b>Run</b> code, press <shortcut actionId="Run"/> or  
// click the <icon src="AllIcons.Actions.Execute"/> icon in the gutter.  
public class Main {  
  
 public static void printBooks(Book[] bks){  
  
 for(Book b : bks){  
  
 String Author = b.Author;  
 Double Price = b.Price;  
 String Name = b.Name;  
 String borrowStatus = "可借";  
 if(b.isBorrowed){  
 borrowStatus = "未还";  
 }  
  
 System.*out*.println(Name + ", " + Author + ", " + Price + "元, " + borrowStatus);  
  
 }  
  
 }  
  
  
 public static void main(String[] args) {  
  
 Library myLittleLibrary = new Library();  
  
 //(1)  
 myLittleLibrary.Books.add(new Book("张三著", 45.0, "Java 程序设计"));  
 myLittleLibrary.Books.add(new Book("李四著", 50.0, "Java 核心设计"));  
 myLittleLibrary.Books.add(new Book("王五著", 38.0, "Java 程序设计"));  
  
 //(2)  
 *printBooks*(myLittleLibrary.GetAllBooks());  
  
 //(3)  
 Reader oneBeautifulGirl = new Reader("meinv");  
  
 Book[] bks = myLittleLibrary.FindBook("Java 程序设计");  
  
 //(4)  
 for (Book b : bks){  
 if (Objects.*equals*(b.Author, "张三著")){  
 oneBeautifulGirl.Borrow(b);  
 }  
 }  
 *printBooks*(myLittleLibrary.GetAllBooks());  
  
 //(5)  
 for (Book b : bks){  
 if (Objects.*equals*(b.Author, "张三著")){  
 oneBeautifulGirl.Return(b);  
 }  
 }  
 *printBooks*(myLittleLibrary.GetAllBooks());  
  
 }  
}

public class Book {  
  
 public Book(String Author, Double Price, String Name){  
  
 this.Author = Author;  
 this.Price = Price;  
 this.Name = Name;  
 this.isBorrowed = false;  
 }  
  
 public String Author;  
  
 public Double Price;  
  
 public String Name;  
  
 public boolean isBorrowed;  
  
  
  
}

public class Reader {  
  
 public Reader(String Name) {  
 this.Name = Name;  
 }  
  
 public String Name;  
  
  
 public void Borrow(Book book){  
  
 if (!book.isBorrowed){  
 book.isBorrowed = true;  
 System.*out*.println("图书借成功");  
 }  
  
 else{  
 System.*out*.println("还未还，不能借");  
 }  
  
 }  
  
 public void Return(Book book){  
  
 if (book.isBorrowed){  
 book.isBorrowed = false;  
 System.*out*.println("图书还成功");  
 }  
 else{  
 System.*out*.println("Error");  
 }  
  
 }  
  
  
}

import java.util.ArrayList;  
import java.util.Objects;  
  
public class Library {  
  
 public Library(){}  
  
 public ArrayList<Book> Books = new ArrayList<>();  
  
  
 public Book[] FindBook(String BookName){  
  
 ArrayList<Book> bks = new ArrayList<>();  
  
 for(Book b : Books){  
 if(Objects.*equals*(b.Name, BookName)) {  
 bks.add(b);  
 }  
 }  
  
 return bks.toArray(new Book[0]);  
 }  
  
 public Book[] GetAllBooks(){  
  
 return Books.toArray(new Book[0]);  
 }  
  
  
}

**（4）实验心得:**

使用类定义对象，并通过这些对象之间的互动来解决问题的方法。创建Library和Reader类，并使用它们的属性和方法来实现功能，是理解和应用OOP核心原理的一个很好的例子。

学习了对象调用方法时如何改变其他对象状态的方法。例如，Reader调用BorrowBook()方法借阅Library的书籍，Library根据此更改书籍的借阅状态。