# Report

The objective of this internship was to implement a system capable of keeping track of the stock and sales of a bookshop. The classes and methods to be implemented are not included in this document as they were given to us in the Lab. This document will not include the classes and methods to be implemented, as they were given to us in the Lab's presentation document. Firstly, the Book class consists of 5 attributes and 5 methods (with the exception of the constructor) where each method is simply a getter of each of the attributes. Next, the Catalog class extends, i.e. it is also a BookCollection. It will have no methods other than the constructor, will not have any method apart from the constructor, which calls the constructor of the superclass. The ShoppingCart class is dedicated to controlling which copies of each book are taken out of the shop and placed in the user's cart to proceed with the purchase. It shall have an attribute catalog, of type Catalog and, in addition, it shall have the collection of the superclass of ShoppingCart(BookCollection). It will be in this one where the books will be stored with the corresponding copies in the cart while in the catalog attribute, we will have a copy of the stock of the shop. The constructor first calls the constructor of superclass and initialises itself with the shop's catalogue. In addition, we have redefined a method getStock(which, in addition to booktitle, takes as an argument the collection in which it should search) to make it easier to find copies of certain books in the addcopies and removecopies methods. addcopies and removecopies methods. As for these two methods, in the first of them, we first call the method firstly we call the getStock method to find the existence of the book in the catalogue of the shop. Finally, we will add the copies specified in the collection and remove them from the collection in the catalog. The implementation of remove is the same as add but in reverse. For the totalprice method, we simply call totalprice for each Stock in the collection (cart). Finally, in the payment method, apart from calling the doPayment method in the return, what is done is to remove the copies that are in the collection. The Stock class, apart from having the information of each book by having a book attribute, also has information about the number of copies, also has information about the number of copies of each book, the price, and the currency. It has getter methods to obtain these attributes as well as addcopies methods, removecopies and totalprice methods, which do the relevant things.

Finally, to make the TestStore class, we initialise 2 catalogues (one for the bookstore's stock and another one for the bookstore's and one for the shopping cart). What we will do first of all is to read the file with the existing books and pass each type of data to the corresponding one.

Next, we create a variable of type Book that stores the data of the book read and 2 Stock variables of which one of them will have the number of copies initialised to 0. This last variable with the copies at 0, will be added to the cart catalogue (cart) while the other will be added to the shop catalogue (books). Finally, we create a shoppingcart initialised with the books of the store and set the books of the shop and set the collection attribute of this shoppingcart with the catalogue of the cart (with the copies set to 0). Finally, we call the constructor of the Bookstore with the corresponding parameters to run the shop.