Application for book ordering

1. Creational Design Patterns
2. Builder

When a user looks for a specific book and he finds the pdf form, he can choose if he also wants to buy the physical form of the book or the kindle edition. These options are not mandatory and are added only if the user wants them.

1. Prototype

A list with all the discounted books should be provided to the users. Creating the categories of the discounted books takes a lot of time because for each category we have to create an object of type pricesList. When we are creating the objects, we have to read from a file all the modified prices. We must find a solution to upload prices fast.

1. Singleton

\*implemented at III.3, together with the Observer Design Pattern

1. Structural Design Patterns
2. Composite

User can choose from a wide range of books. They are organized in a hierarchical structure of categories. Each category has subcategories with options, from which the user can choose. (Arts -> Painting ->Caricature

->Technical Drawing

->Photography ->Abstract

->Cityscape

->Macro)

2. Façade

Users can also order via phone from the book catalog, thus encountering a façade. The user calls one number and speaks with a customer service representative. He acts as a façade, providing an interface to the order fulfillment department, the billing department and the shipping department.

3. Decorator

Users can dynamically add additional objects for the books they order (bookmarks, post-it notes, covers), to give them personalized looks. The ornaments do not change the book itself.

1. Behavioral Design Patterns
2. Strategy

The application has an implementation for the payment module when a book is ordered. The way of paying is decided by the buyer in the moment of order. The payment can be made with a card, cash or PayPal.

1. State

Books have different states based on the inventory, order status. If a book is no longer available, its state will be “out of stock”. If it is on stock and someone orders it, the purchase will first appear as “ordered”, “processing”, “accepted”, “delivered”.

1. Observer & Singleton

A customer service representative wants to notify users when special offers are available. It is decided to implement a module which notifies users each time an offer appears. The application must manage this module very carefully, because it cannot be more than one administrator who sends the notifications. For solving this issue, it is used only one instance of the administrator.