ABSTRACTION

You will develop product management part of an E-Commerce system. In this E-Commerce system anything, a book, a smart phone, a TV etc. can be a product. You will develop an extensible Product system with the following requirements and features

1. Develop an abstract type which can refer to all products. Its name will be Product
    1. This Product type will have following common properties of all products
        1. code: A unique positive integer number. It is automatically generated for each product starting from 1.
        2. description: A String description of product at max 25 chars, if it is longer than 25 letters, first 25 letters are accepted. If no description is entered it should be set to empty
        3. cost: A positive double number that stores the cost, for negative entries and zero entries it should be set to 1.
        4. price: A double number that stores the price. It is calculated automatically by the system according to the product types as specified below.
    2. This Product type has toString() method that overrides the Object class toString() and returns the product in the following format.
        1. Product{Code:<code>, Description:<description>, Price:<price>}
           Product{Code:100, Description:Samsung Galaxy Smart Phone, Price:550}
    3. This Product type will have an abstract method with the following signature that calculates returns price of the product. Price calculation will be burdened to concrete sub-classes of the Product.
        1. double getPrice()
2. You are expected to develop only two concrete sub- classes of abstract Product
    1. Book class
        1. It should have one more field , author, to store author name
            1. author: A string that stores the author name at max 25 chars, if it is longer than 25 letters, first 25 letters are accepted. If no entries it should be set to UNKNOWN_AUTHOR
        2. Price will be calculated for all books by multiplying cost by 1.15.
    2. Camera class
        1. It will have two more fields
            1. brand: A string that stores the brand name at max 10 chars, if it is longer than 10 letters, first 10 letters are accepted. If no entry it should be set to NO_BRAND
            2. megaPixels: An integer that stores the effective megapixels in between 1 and 40. Other entries should be set to 1.
        2. Price will be calculated for all cameras by multiplying cost by 1.25.
3. All products in the system should be Relatable and Copyable.
4. You will expected to develop Relatable interface with having the relate method with the following signature Object relate(Object)
    1. If a book ,A, is Relatable, it can be compared to an other book B
        1. A.relate(B) returns if price of A is equal to or grater than B's price, otherwise return B. If B is not a Book it will return null
    2. If a Camera ,A, is Relatable, it can be compared to an other camera B
        1. A.relate(B) returns if megapixel of A is equal to or grater than B's megapixel, otherwise return B. If B is not a Camera it will return null
5. You will also expected to develop Copyable interface with having the copy method with the following signature. Object copy()
    1. If a Product is copied it should create a new product that has the name properties as the original one has except code. It will assign a new code to the product.

6. You can use helper methods, default methods static methods if you need, although it is not required.
7. All fields should be private.