C# - Assignment 5
Inheritance

In this exercise, we are proposing a design modeling a mailbox in object-oriented.

A mailbox collects letters, parcels, and advertisements.

A letter is characterized by:

- its weight (in grams)
- the shipping method (express or normal) its destination address
- its format ("A3" or "A4")

A parcel is characterized by:

- its weight (in grams)
- the shipping method (express or normal) its destination address
- its volume (in liters)

An advertisement is characterized by:

- its weight (in grams)
- the shipping method (express or normal) its destination address

Here are the rules used to stamp the mail:

1. in normal shipping method:
    - the amount required to stamp a letter depends on its size and weight:
      Formula: amount = base fare + 1.0 * weight (kg), where the base fare for a letter "A4" is 2.50, and 3.50 for a letter "A3"
    - the amount needed to stamp an advertisement depends on its weight:
      Formula: amount = 5.0 * weight (kilos)
    - the amount required to stamp a parcel depends on its weight and volume:
      Formula: amount = 0.25 * volume (liters) + weight (kilos) * 1.0;

2. In express shipping mode: the previous amounts are doubled, regardless of the type of mail;
3. Valid mails are stamped;
4. A mail is not valid if the destination address is empty;
5. A parcel is invalid if its destination address is empty or exceeds a volume of 50 liters.

The three main methods related to the mailbox are:

1. a `stamp()` method allowing to associate to each mail of the box, the amount necessary to stamp it. This method will return the total postage amount of the mail.
2. a method `invalidMails()` calculating and returning the number of invalid mails present in the mailbox.
3. a `display()` method displaying the contents of the mailbox (we will then indicate which mails are invalid).

On paper, start by drawing a hierarchy of classes to implement the suggested design considering the constraints mentioned. You will specify in your diagram the classes, attributes, and headers of the methods (without the bodies).

The following constraints must be respected:

1. Your design must be done in such a way that none of the required methods need to test the nature of the object to which it applies.
2. Classes must provide all the methods they need.
3. A class will only include the methods / attributes that are specific to it.
4. Access modifiers shall be clearly specified.
5. Your classes should avoid unnecessarily duplicating methods or attributes and will be compatible with the main program provided in the provided file Post.cs

Then implement the program resulting from your design in the Post.cs file. With the main program provided, you should have a run such that (the price indicates the cost of postage):

```
The total amount of postage is 47.4
Letter
        Weight: 200.0 grams
        Express: yes
        Destination: Chemin des Acacias 28, 1009 Pully Price: $ 7.4
        Format: A3
Letter
        (Invalid courier)
        Weight: 800.0 grams Express: no Destination:
        Price: 0.0 CHF Format: A4
Advertisement
        Weight: 1500.0 grams
        Express: yes
        Destination: Les Moilles 13A, 1913 Saillon Price: $ 15.0
Advertisement
        (Invalid courier)
        Weight: 3000.0 grams Express: no
        Destination: Price: 0.0
Parcel
        Weight: 5000.0 grams
        Express: yes
        Destination: Grand rue 18, 1950 Sion Price: $25.0
        Volume: 30.0 liters
Parcel
        (Invalid courier)
        Weight: 3000.0 grams
        Express: yes
        Destination: Chemin des fleurs 48, 2800 Delemont Price: 0.0
        Volume: 70.0 liters
The box contains 3 invalid mails
```