1 Activity 1 (10 points)

This activity requires you to use many of the concepts learned so far on the module. You will develop a simple eCommerce system. The program has to allow the user to:

- add products;
- remove products;
- show a summary of their shopping session;
- export the cart in *JSON* format.

The activity is divided in three parts that add up to 10 points. You need to submit one file called `shopping.py`.

1.1 The domain classes (3 points)

First create a class **Product** with the following attributes:

- `name`;
- `price`;
- `quantity`;
- `EAN identifier` (this is the 13-digit number below the bar code of a product, see also https://en.wikipedia.org/wiki/International_Article_Number);
- `brand`.

The class **Product** also offers a `to_json()` method that returns the *JSON* representation of the product.

Then, create three subclasses of the class **Product**:

- **Clothing**, which has the following attributes: `size`, and `material`;
- **Food**, which has the following attributes: `expiry_date`, `gluten_free`, and `suitable_for_vegans`; and
- one additional subclass of your choice. Think about the domain of your system and pick a product you are familiar with. Define at least 2 relevant attributes of that product.

Each subclass of **Product** must override the `to_json()` method of the superclass.

1.2 The shopping system (3 points)

Next, create the class **ShoppingCart**, which is a container of products in a shopping session. The **ShoppingCart** class offers the following methods:

- `addProduct(p)`, to add a product p to the cart;
- `removeProduct(p)`, to remove the product p from the cart;
- `showSummary()`, to display the content of the cart;

- changeProductQuantity(p, q), to change the quantity of product p to the quantity q .

1.3 Doing some shopping (4 points)

Now that the classes are ready, you need some code to let the user do some shopping. The script is prompting the user to type in commands in a while loop; see Listing 1.

```
print('The program has started.')
print('Insert your next command (H for help):')
terminated = False
while not terminated:
    c = input.read("Type your next command:")
    ....
    ....
print( 'Goodbye.' )
```

Listing 1: Structure of the main while loop. The loop can terminate only when the users types the command **T**

The script should support the following commands:

- **A** - allows the user to add a product to the cart (see the example in Listing 2);
- **R** - allows the user to remove an existing product from the shopping chart (see Listing 3);
- **S** - the script prints out a summary of the cart, along the lines of the example in Listing 4;
- **Q** - the user can change the quantity of a product already present in the cart;
- **E** - the script generates a summary of the cart as *JSON*, printed to the console. The *JSON* output is an array of *JSON* products (see the example in Listing 5);
- **C** - at any moment, allows the user to interrupt the shopping session. The program will print out the following message: *"The current operation is cancelled"*. Then, the user should be able to continue with a new command.
- **T** - the script terminates (exiting the while loop);
- **H** - a request for help from the user. The commands that the program recognises are printed out to the console (see in Listing 6);

Any other command should print out the following message: *"Command not recognised. Please try again"*.

```
>>> Insert the next command: A
>>> Adding a new product:
>>> Insert its type: Clothing
>>> Insert its name: Gloves
>>> Insert its price (£): 23.5
>>> Insert its quantity: 1
>>> Insert its brand: Moret and Spark
>>> Insert its EAN code: 1234567891234
>>> Insert its size: XL
>>> Insert its material: Sink
>>> The product Gloves has been added to the cart.
>>> The cart contains 4 products.
```

Listing 2: Adding a new product to the cart

```
>>> Insert the next command: R
>>> What is the name of the product you want to remove? Shoes
>>> 'Shoes' is not in the cart.
>>> Insert the next command: R
>>> What is the name of the product you want to remove? Slippers
>>> 'Slippers' successfully removed from the cart.
>>> The cart contains 3 products.
```

Listing 3: Removing a product from the cart

```
>>> Insert the next command: S
>>> This is the total of the expenses:
>>>   1 - Hat = £15
>>>   2 - 6 * Eggs  = £2.4
>>>   3 - 2 * Orange juice = £5
>>>   4 - "Gloves" = £23.5
>>>   Total = £45.9
```

Listing 4: The summary of an ongoing shopping session

```
>>> Insert the next command: E
>>> [{ "type": \clothing", \name": \Hat", . . .
```

Listing 5: *JSON* export of the cart content

```
>>> Insert the next command: H
>>> The program supports the following commands:
>>>    [A] - Add a new product to the cart
>>>    [R] - Remove a product from the cart
>>>    [S] - Print a summary of the cart
>>>    [Q] - Change the quantity of a product
>>>    [E] - Export a JSON version of the cart
>>>    [C] - Cancel the current operation
>>>    [T] - Terminate the program
>>>    [H] - List the supported commands
```

Listing 6: User help request