

Develop the tasks described below:

I.) Payment Card

In this exercise series, a class called `PaymentCard` is created which aims to mimic a cafeteria's payment process.

- The template includes the `Program.cs` file. You have to create the `PaymentCard.cs` yourself.
- Add a new class to the project called `PaymentCard` by creating the file mentioned above.
- Create the `PaymentCard` object's constructor, which is passed the opening balance of the card and then stores that balance in the object's internal variable.
- Write the `ToString` method, which will return the card's balance in the form "The card has a balance of X euros".

Here is the template for the `PaymentCard`.

```
namespace exercise_106
{
    public class PaymentCard
    {
        private double balance;

        public PaymentCard(double openingBalance)
        {
            // write code here
        }

        public override string ToString()
        {
            // write code here
        }
    }
}
```

The following main program tests the class.

```
public static void Main(string[] args)
{
    PaymentCard card = new PaymentCard(50);
    Console.WriteLine(card);
}
```

The card has a balance of 50 euros

II.)Using Card

Expand your answer from exercise I.) by adding two methods.

- Method public void EatLunch()
- Method public void DrinkCoffee()
- The method EatLunch() should decrease the card's balance by 10.60 euros.
- The method DrinkCoffee should decrease the card's balance by 2.0 euros.

The following main program tests the class.

```
public static void Main(string[] args)
{
    PaymentCard card = new PaymentCard(50);
    Console.WriteLine(card);

    card.EatLunch();
    Console.WriteLine(card);

    card.DrinkCoffee();
    Console.WriteLine(card);
}
```

The card has a balance of 50 euros

The card has a balance of 39.4 euros

The card has a balance of 37.4 euros

III.) Checking Balance

Expand your previous answers (I,II) so that when an item is bought the balance is checked.
If there is not enough money to buy the balance does not change.

```
public static void Main(string[] args)
{
    PaymentCard card = new PaymentCard(10);
    Console.WriteLine(card);

    card.EatLunch();
    Console.WriteLine(card);

    card.DrinkCoffee();
    Console.WriteLine(card);
}
```

The card has a balance of 10 euros
The card has a balance of 10 euros
The card has a balance of 8 euros

NOTICE! See how the EatLunch() method did not change the balance when there was not enough money. The DrinkCoffee() method still worked as it should.

IV.)Charging Card

Expand your previous answers so that you can charge money on your card.

```
public void AddMoney(double amount) {  
    // write code here  
}
```

The purpose of the method is to increase the card's balance by the amount of money given as a parameter.

However, the card's balance may not exceed 150 euros.

As such, if the amount to be topped up exceeds this limit the balance should become exactly 150 euros.

The following main program tests the class.

```
public static void Main(string[] args)  
{  
    PaymentCard card = new PaymentCard(100);  
    Console.WriteLine(card);  
  
    card.AddMoney(49.99);  
    Console.WriteLine(card);  
  
    card.AddMoney(10000.0);  
    Console.WriteLine(card);  
  
    card.AddMoney(-10);  
    Console.WriteLine(card);  
}
```

The card has a balance of 100 euros
The card has a balance of 149.99 euros
The card has a balance of 150 euros
The card has a balance of 150 euros

V.)Book Class

- Write a program that first reads book information from the user.
- The details to be asked for each book include the title, the number of pages and the publication year.
- Entering an empty string as the name of the book ends the reading process.
- After this the user is asked for what is to be printed.
- If the user inputs “everything” all the details are printed: the book titles, the number of pages, and the publication years.
- However if the user enters the string “title” only the book titles are printed.
- If something other than “everything” or “title” is given the program should not print anything.
- Implement the class Book.
- Implement the functionality in the Main method.

Example of how the program in Main should work.

Name: To Kill a Mockingbird
 Pages: 281
 Publication year: 1960
 Name: A Brief History of Time
 Pages: 256
 Publication year: 1988
 Name: Beautiful Code
 Pages: 593
 Publication year: 2007
 Name: The Name of the Wind
 Pages: 662
 Publication year: 2007
 Name:

What information will be printed? everything

To Kill a Mockingbird, 281 pages, 1960
 A Brief History of Time, 256 pages, 1988
 Beautiful Code, 593 pages, 2007
 The Name of the Wind, 662 pages, 2007

What information will be printed? title

To Kill a Mockingbird
 A Brief History of Time
 Beautiful Code
 The Name of the Wind

Add a logic to store the information from the user in a file (.csv) and then read it from the file.

