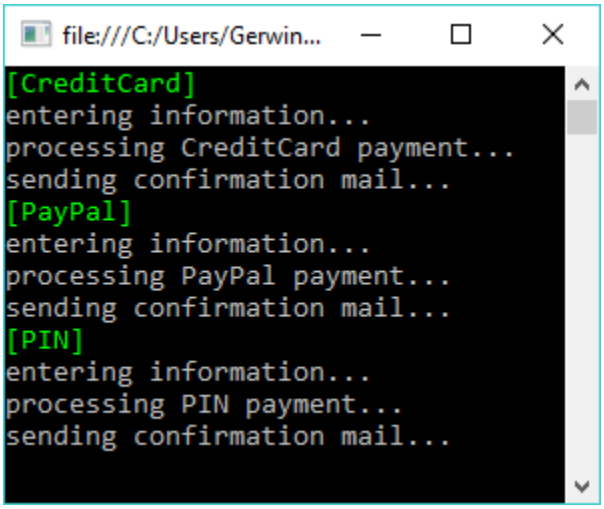
Assignment 1 (15 points)

In this assignment a (simple) Console application has to be made, in which payments can be done in several ways *(creditcard, paypal, pin)*. Doing a payment involves 3 separate steps: 1. entering the information (like amount and accountnumber), 2. the actual processing of the payment and 3. the confirmation of the payment. Entering information and the confirmation of the payment is independent of the type of payment; the actual processing of the payment does depend on the type of payment.

Use the Main-code below (you can find it on Blackboard, don't change it), implement the required classes (*using the appropriate Design Pattern*) in order to generate the output below.

```
static void Main(string[] args)
{
    PrintHeader("[CreditCard]");
    Payment ccPayment = new CreditCardPayment();
    ccPayment.Execute();

    PrintHeader("[PayPal]");
    Payment ppPayment = new PayPalPayment();
    ppPayment.Execute();

    PrintHeader("[PIN]");
    Payment pinPayment = new PINPayment();
    pinPayment.Execute();

    Console.ReadKey();
}

static void PrintHeader(string header)
{
    Console.ForegroundColor = ConsoleColor.Green;
    Console.WriteLine(header);
    Console.ResetColor();
}
```

Use the Main-code on Blackboard. *(assignment1-main.cs)*



Assignment 2

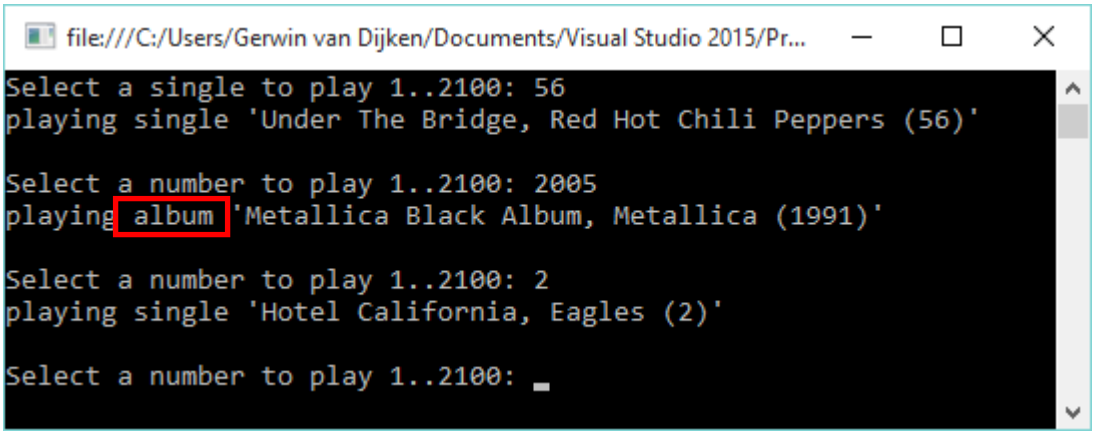
(20 points)

A music freak has a jukebox that can play vinyl singles. A song can be selected, and after pressing the Play-button, the single is picked up and played. There's also a Stop-button to stop playing the song and to store back the single.

Very nice, but when his friend comes along with his collection of vinyl albums, they noticed that the jukebox can not play these albums! Your task is to make a modification so the jukebox can not only play vinyl singles but also vinyl albums.

Create the displayed classes and interfaces (each in a separate file) and add an extra class so the jukebox can also play albums.
Use the appropriate design pattern for this problem.

Use the Main-code on Blackboard.
(assignment2-main.cs)

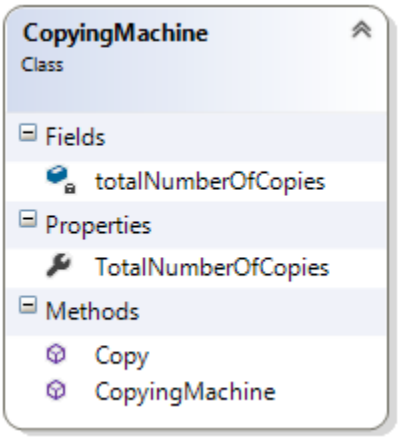


Assignment 3

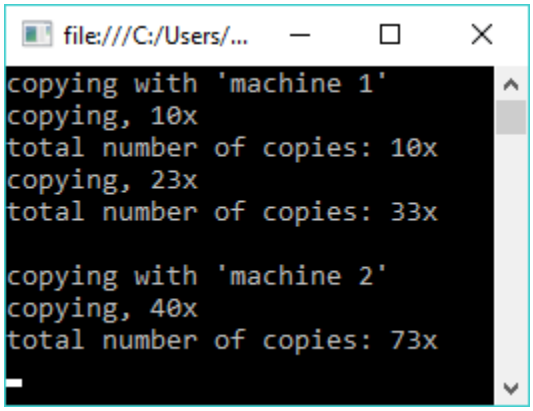
(15 points)

A company has one copying machine that is used by several employees. If we would implement such a machine in a software program, we would have to make sure that everone is using the same machine (instance). Otherwise the (e.g.) number of total copies would not be correct.

Create class 'CopyingMachine' (including the given members/fields, properties and methods) and modify this class to make sure there can only be one instance created from it. Use the appropriate design pattern.



Use a simple (Main) program to show how a CopyingMachine can be created and used. Also show that copying with '2' machines, the total number of copies correctly add up (as shown below).

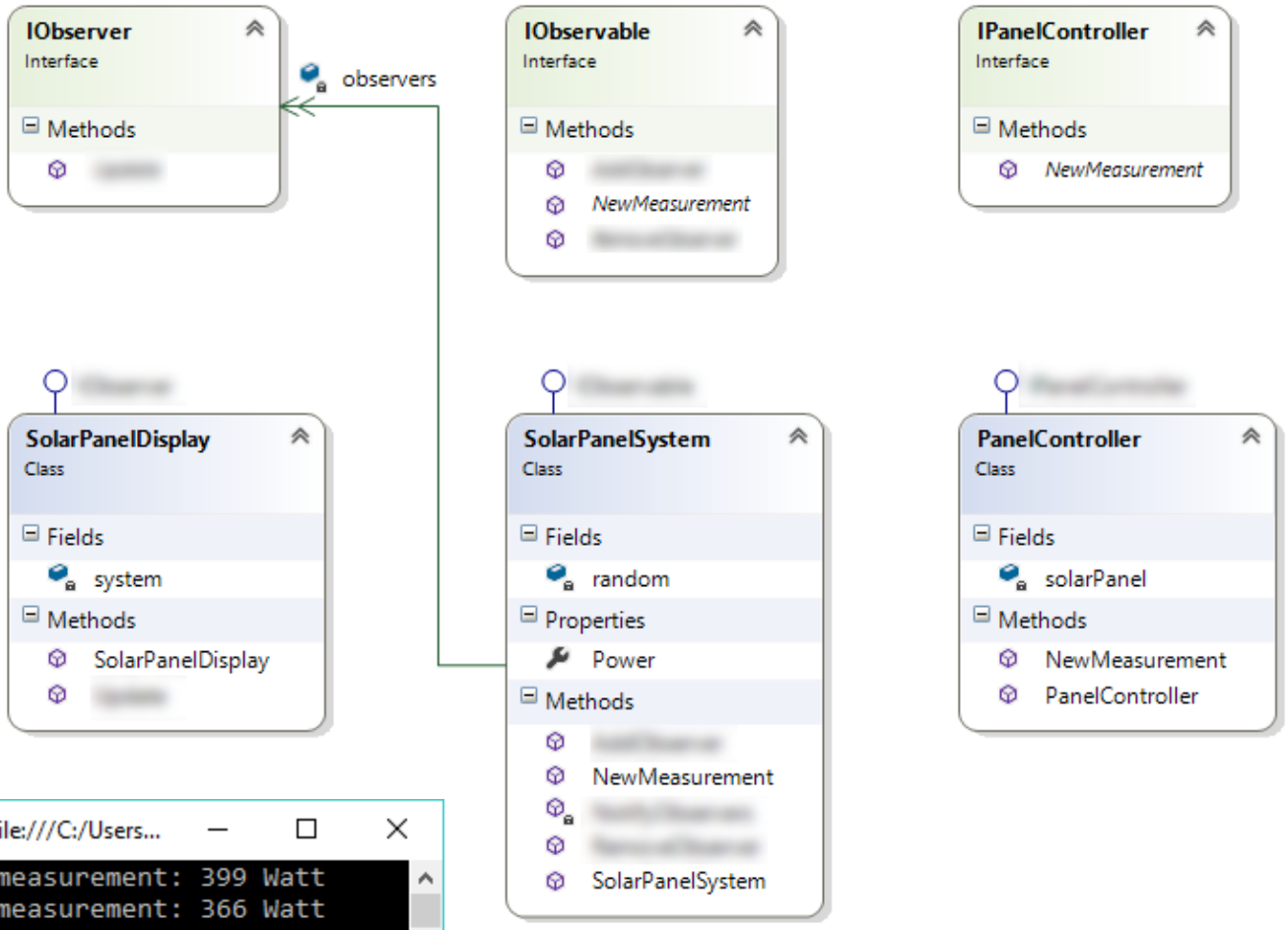


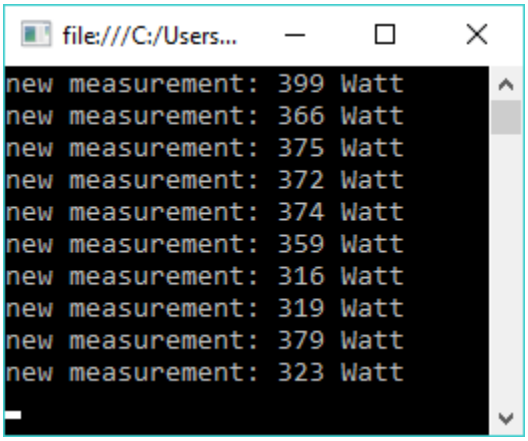
Assignment 4

(30 points)

A manufacturer of solarpanel systems delivers with each system a display that is displaying live the power of the system.
This 'actual power' is shown in Watts (a whole number).

Create a Console program that contains the classes and interfaces as shown in the class diagram below.
The solarpanel system is being controlled by a controller and observed by a solarpanel display; the display shows the measurement values.





Use the Main-code on Blackboard.
(assignment4-main.cs)