

Opgave 1

(15 punten)

Maak een Console-project aan in Visual Studio met de naam '**Opgave1**' en geef als solution naam op '**DesignPatterns-tentamen**'.

(de overige opgaven (2 t/m 4) worden ook als projecten in deze solution aangemaakt, zie later)

In deze opdracht moet er een (eenvoudige) Console-applicatie gemaakt worden waarin verschillende type woningen 'gebouwd' kunnen worden. Het bouwen bestaat uit 4 vaste stappen: 1. fundering plaatsen, 2. muren plaatsen, 3. dak plaatsen en 4. ramen plaatsen. Het plaatsen van een fundering en het plaatsen van de ramen is onafhankelijk van het type woning; het plaatsen van de muren en het plaatsen van het dak is wel afhankelijk van het type woning.

Neem onderstaande Main-code over (staat op Moodle, niet wijzigen) en implementeer de benodigde classes (gebruikmakende van de juiste Design Pattern) zodat onderstaande uitvoer ontstaat.

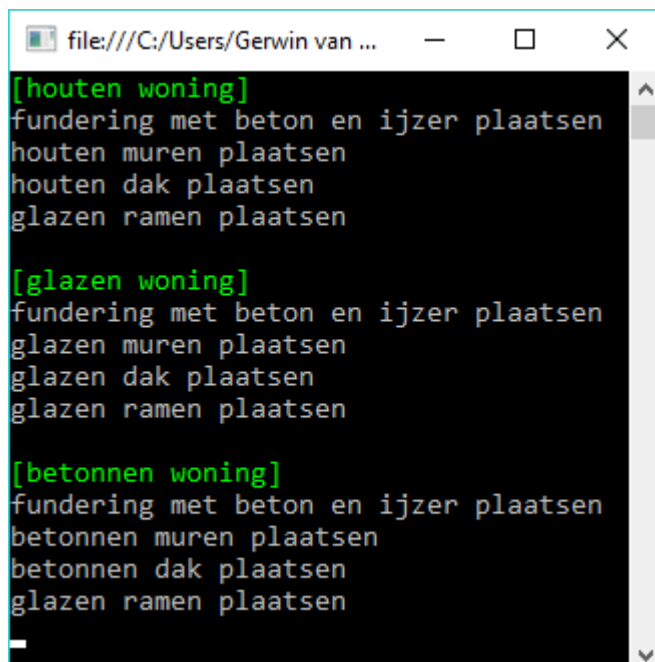
```
private void Start()
{
    PrintHeader("[houten woning]");
    BasisWoning huis1 = new HoutenWoning();
    huis1.WoningBouwen();

    Console.WriteLine();

    PrintHeader("[glazen woning]");
    BasisWoning huis2 = new GlazenWoning();
    huis2.WoningBouwen();

    Console.WriteLine();

    PrintHeader("[betonnen woning]");
    BasisWoning huis3 = new BetonnenWoning();
    huis3.WoningBouwen();
}
```



```
file:///C:/Users/Gerwin van ...
[houten woning]
fundering met beton en ijzer plaatsen
houten muren plaatsen
houten dak plaatsen
glazen ramen plaatsen

[glazen woning]
fundering met beton en ijzer plaatsen
glazen muren plaatsen
glazen dak plaatsen
glazen ramen plaatsen

[betonnen woning]
fundering met beton en ijzer plaatsen
betonnen muren plaatsen
betonnen dak plaatsen
glazen ramen plaatsen
```

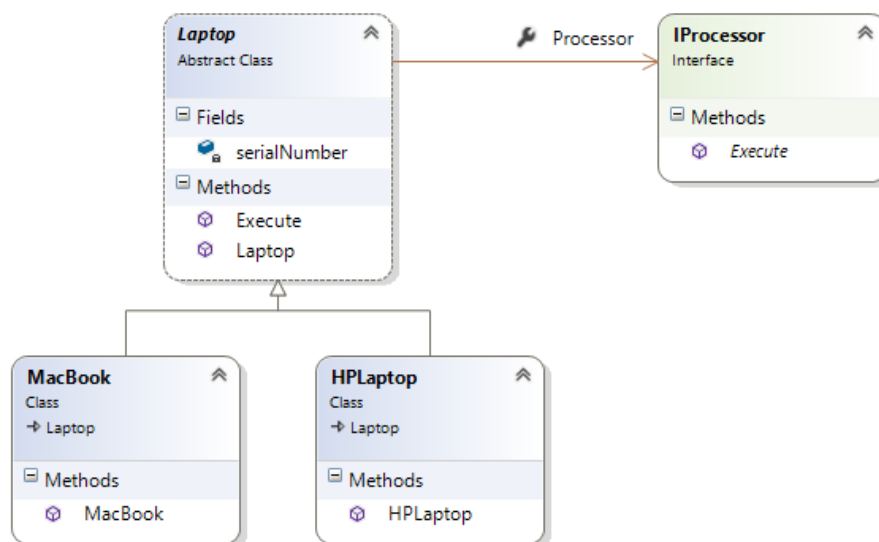
Opgave 2

(20 punten)

Voeg aan solution 'DesignPatterns-tentamen' een Console-project toe met de naam '**Opgave2**'.

Ontwikkel een Console applicatie waarin verschillende soorten laptops aanwezig zijn (afgeleiden van *abstract class Laptop*). Elke laptop gebruikt een bepaald type processor (Intel-i5, Intel-i7 of AMD-Ryzen-3): een MacBook heeft standaard een Intel-i5 processor, een HP-laptop standaard een AMD-Ryzen-3 processor. Het type processor (van een laptop) kan gewijzigd worden tijdens het runnen v/d applicatie. *Als het goed is, herken je welke design pattern hier gebruikt wordt.*

Met onderstaande (deel) class diagram en Start-code (staat op Moodle, niet wijzigen, behalve de TODO-regel) moet duidelijk zijn hoe de applicatie gemaakt moet worden.



```
private void Start()
{
    PrintHeader("MacBook");
    Laptop macBook = new MacBook("S/N A1287");
    macBook.Execute("virusscanner.exe");

    PrintHeader("HP");
    Laptop hp = new HPLaptop("S/N 893419");
    hp.Execute("virusscanner.exe");

    PrintHeader("changed MacBook");
    // wijzig macBook hier... (TODO)
    macBook.Execute("virusscanner.exe");
}
```

```
file:///C:/Users/Gerwin van Dijken/Documents/Visual Studio 2015/...
MacBook
Executing program 'virusscanner.exe' with processor Intel i5
HP
Executing program 'virusscanner.exe' with processor AMD Ryzen 3
changed MacBook
Executing program 'virusscanner.exe' with processor Intel i7
```

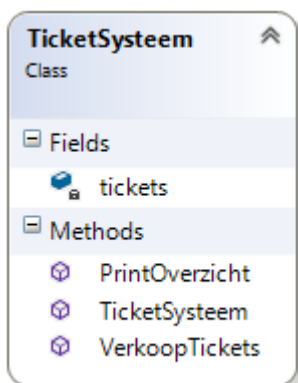
Opgave 3

(15 punten)

Voeg aan solution 'DesignPatterns-tentamen' een Console project toe met de naam '**Opgave3**'.

In een applicatie wordt een ticketsysteem op verschillende plekken gebruikt. Om te voorkomen dat er teveel tickets verkocht worden, mogen er geen verschillende instanties van dit ticketsysteem worden aangemaakt.

Implementeer onderstaande 'TicketSysteem' class (inclusief de aangegeven members/fields en methoden) en wijzig deze class waardoor er maximaal één instantie van kan bestaan. *Gebruik de juiste design pattern.*



Gebruik een Dictionary voor de tickets (`Dictionary<string, int>`), waarin per artiest (=key: string) het aantal tickets (=value: int) opgeslagen kan worden. Neem onderstaande code op in de constructor van TicketSysteem:

```
tickets = new Dictionary<string, int>();  
tickets.Add("Bruno Mars", 250);  
tickets.Add("Coldplay", 175);  
tickets.Add("Ed Sheeran", 150);
```

De definitie van de 2 methoden is als volgt:

```
public void PrintOverzicht() { ... }  
deze methode toont het aantal tickets van de artiesten
```

```
public void VerkoopTickets(string artiest, int aantal) { ... }  
deze methode verlaagt het aantal tickets van de betreffende artiest (mits aanwezig)
```

Toon met een simpel main-programma aan hoe het TicketSysteem kan worden aangemaakt en gebruikt. Laat hierbij duidelijk zien dat 2 'verschillende' ticketsystemen toch dezelfde instantie zijn.

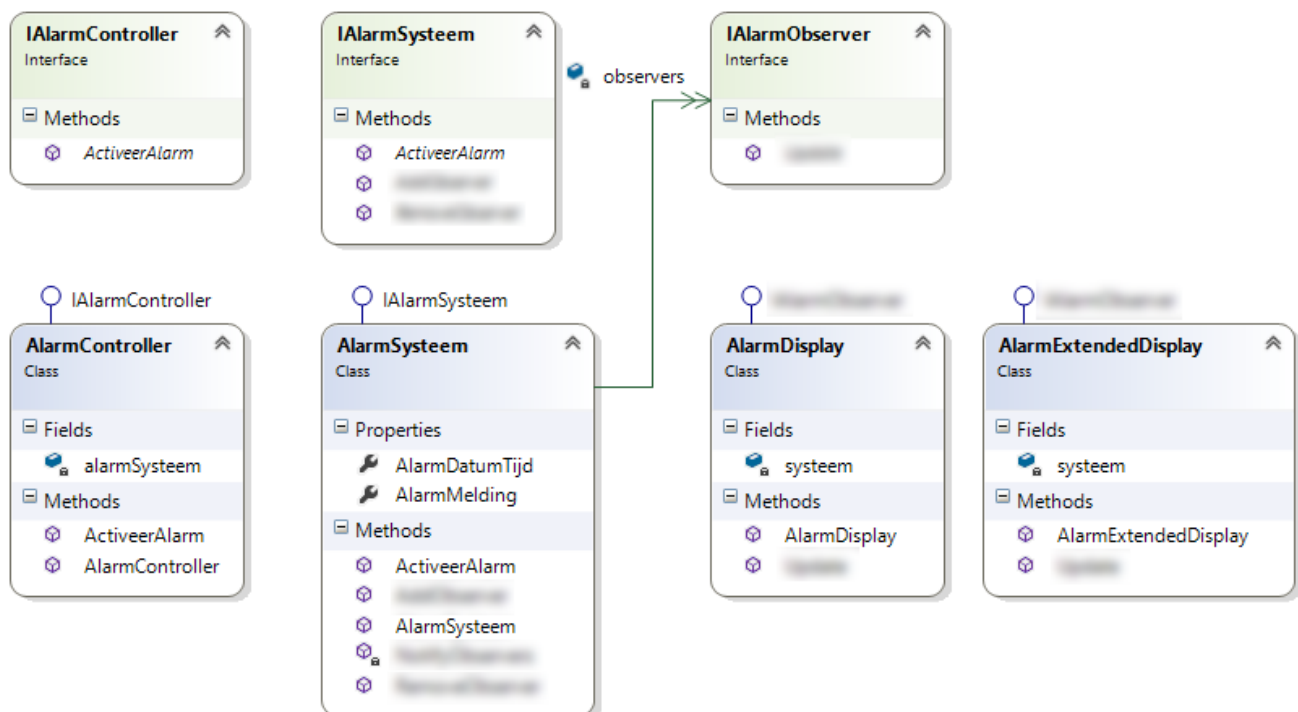
Opgave 4

(30 punten)

Voeg aan solution 'DesignPatterns-tentamen' een Console project toe met de naam '**Opgave4**'.

Maak een Console applicatie op basis van onderstaand class-diagram, waarin een alarmsysteem alarm-meldingen kan verwerken. Er kunnen meerdere schermen 'luisteren' naar dit alarmsysteem. Steeds als er een alarm-melding komt, moeten alle (aangemelde) schermen ge-update worden.

Er zijn 2 soorten schermen: een standaard scherm dat alleen de datum/tijd van een alarm weergeeft, en een 'extended' scherm, dat zowel de datum/tijd als de precieze melding van een alarm weergeeft.
(*formaat datum/tijd: {0: dd/MM/yyyy HH:mm:ss}*)



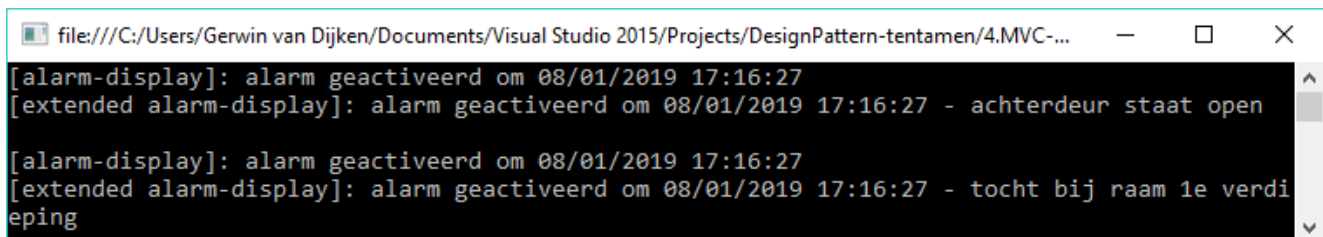
Het geheel kun je testen via het programma op de volgende pagina (deze code staat op Moodle; niet wijzigen, behalve het aanmaken van een controller en de schermen).

```
private void Start()
{
    // maak alarmsysteem aan
    IAlarmSysteem alarmSysteem = new AlarmSysteem();

    // maak een controller aan
    ... (TODO)

    // maak displays aan
    ... (TODO)

    // activeer het alarmsysteem een paar keer (om te testen)
    controller.ActiveerAlarm("achterdeur staat open");
    Console.WriteLine();
    controller.ActiveerAlarm("tocht bij raam 1e verdieping");
    Console.WriteLine();
}
```



The screenshot shows a console window with the following output:

```
file:///C:/Users/Gerwin van Dijken/Documents/Visual Studio 2015/Projects/DesignPattern-tentamen/4.MVC-...
[alarm-display]: alarm geactiveerd om 08/01/2019 17:16:27
[extended alarm-display]: alarm geactiveerd om 08/01/2019 17:16:27 - achterdeur staat open

[alarm-display]: alarm geactiveerd om 08/01/2019 17:16:27
[extended alarm-display]: alarm geactiveerd om 08/01/2019 17:16:27 - tocht bij raam 1e verdi
eping
```