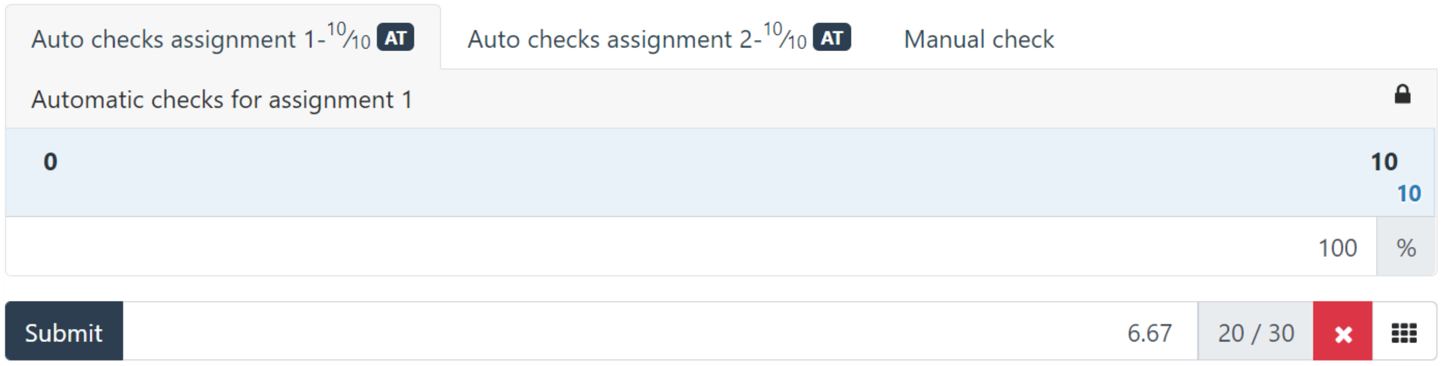
# Week 6 assignments

When creating the program code, you must apply the following basic principles:

- create a separate project for each assignment;
- use name ‘assignment1’, ‘assignment2’, etcetera for the projects;
- create one solution for each week containing the projects for that week;
- make sure the output of your programs are the same as the given screenshots;

## CodeGrade auto checks

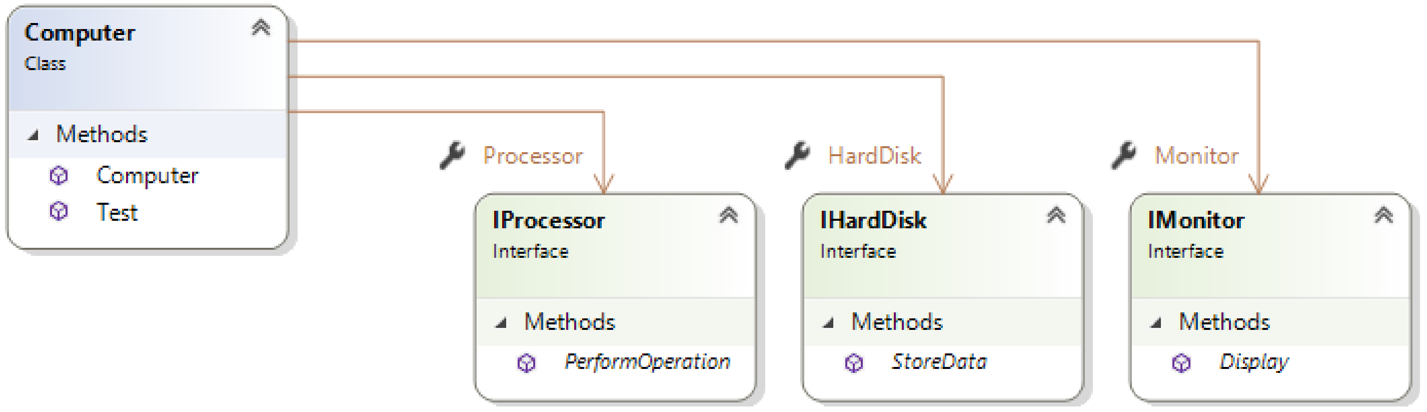
Make sure all CodeGrade auto checks pass (10/10) for your assignments. The manual check will be done by the practical teacher.


Auto checks assignment 1- 10/10 AT
Auto checks assignment 2- 10/10 AT
Manual check
Automatic checks for assignment 1
0
10
10
100 %
Submit
6.67
20 / 30

## Assignment 1 ('Factory Method')

In this assignment computers are created in computershops; your task is to use Design Pattern 'Factory Method' for creating these computers. This means that computer parts (processor, monitor and hard disk) are being created through **virtual or abstract** methods (in the computershop). There are two different shops, a LowBudgetShop creating cheap computer parts, and a HighBudgetShop creating expensive computer parts.

Use the following Computer class and interfaces for the computer parts:



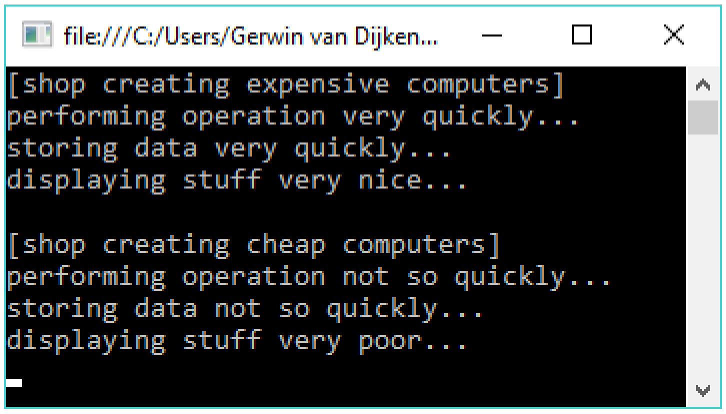
The computer parts are created in (factory) method "AssembleComputer" of the ComputerShop class, and this method returns a new computer containing these created computer parts. After assembling a new computer, it's being tested: for each part, the corresponding method is being called (e.g. "PerformOperation" of the processor).

Use the code below:

```
void Start()
{
  // create a shop where they assemble expensive computers
  Console.WriteLine("[shop creating expensive computers]");
  // ... create shop
  // ... assemble (one) computer
  // ... test the new computer

  // create a shop where they assemble cheap computers
  Console.WriteLine("[shop creating cheap computers]");
  // ... create shop
  // ... assemble (one) computer
  // ... test the new computer
}
```

to generate the output to the right:



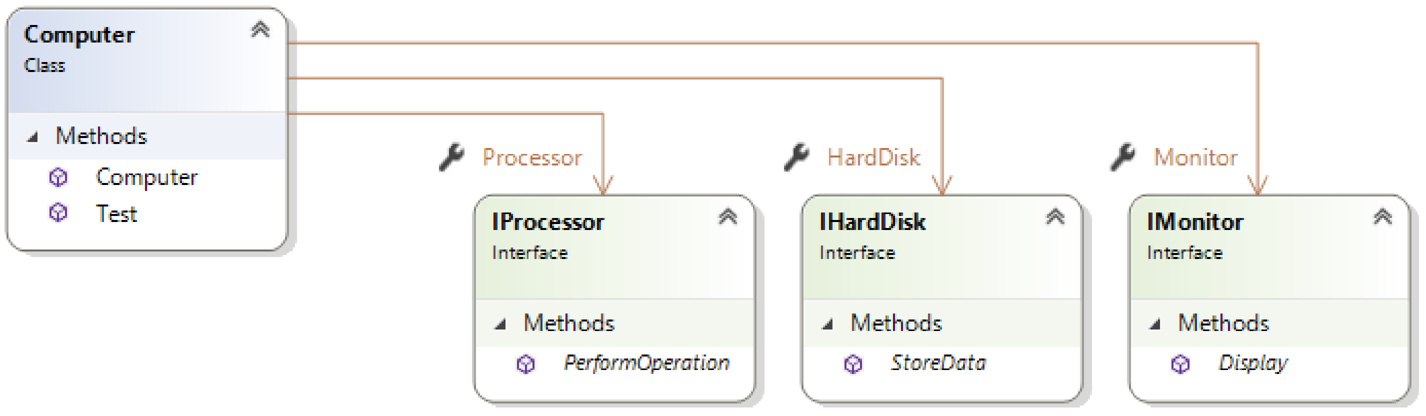
```
[shop creating expensive computers]
performing operation very quickly...
storing data very quickly...
displaying stuff very nice...

[shop creating cheap computers]
performing operation not so quickly...
storing data not so quickly...
displaying stuff very poor...
```

## Assignment 2 (‘Abstract Factory’)

In this assignment computers are created in computershops (again); your task to use Design Pattern ‘Abstract Factory’ for creating these computers. This means that computer parts (processor, monitor and hard disk) are created through **a factory**. There are two different factories, a LowBudgetFactory creating cheap computer parts, and a HighBudgetFactory creating expensive computer parts.

Use the following Computer class and interfaces for the computer parts:



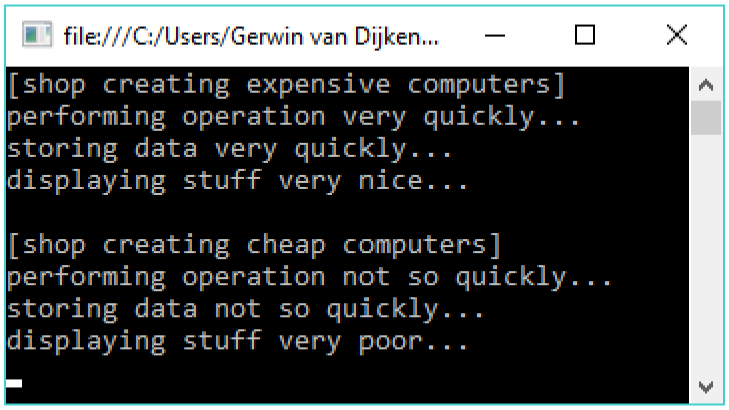
The computer parts are created in method “AssembleComputer” of the ComputerShop class (using a factory), and this method returns a new computer containing these created computer parts. After assembling a new computer, it’s being tested: for each part, the corresponding method is being called (e.g. “PerformOperation” of the processor).

Use the code below:

```
void Start()
{
  // create a shop where they assemble expensive computers
  Console.WriteLine("[shop creating expensive computers]");
  // ... create factory
  // ... create shop
  // ... assemble (one) computer
  // ... test the new computer

  // create a shop where they assemble cheap computers
  Console.WriteLine("[shop creating cheap computers]");
  // ... create factory
  // ... create shop
  // ... assemble (one) computer
  // ... test the new computer
}
```

to generate the output to the right:



```
[shop creating expensive computers]
performing operation very quickly...
storing data very quickly...
displaying stuff very nice...

[shop creating cheap computers]
performing operation not so quickly...
storing data not so quickly...
displaying stuff very poor...
```