

Faculty of Mathematics and Physics
Charles University in Prague
23rd February 2015



C# Made Easy!

Programming II

Workshop 2 – Gentle C# + OOP Introduction

Workshop 2

Outline

1. Practice Terms
2. Test
3. C#, OOP, UML
4. Homework



Practice Terms

Credits ~ Hours Donation

1 Credit	<i>45 minutes of work per week</i>
Programming II (aka NPRG031) 14bNPRG031x06	<i>2/2 points</i> <i>6 credits</i> <i>4,5 hours</i>
Lecture	<i>1,5 hours</i>
Practice	<i>1,5 hours</i>
Homework	<i>1,5 hours</i> lower-bound!

Practice Terms

Score-based Grading

Practice Number	Monday	Topic	Attendance	Test	Test Scoring	Assignment	Assignment Scoring
				<i>Somewhat demanding</i>	6	<i>Demanding, but we expect everybody to fulfill</i>	
1		16.2.2015 PRG-I Revisited	2	No		Send me an email	5
2		23.2.2015 C#, Gentle OOP introduction	2	Yes	6	Simple Number Reader, Simple Calc	10
3		2.3.2015 V pátek 9:00?	2	Yes	6		10
4		9.3.2015	2	Yes	6		10
5		16.3.2015	2	Yes	6		10
6		23.3.2015 V pátek 9:00?	2	Yes	6		10
7		30.3.2015	2	Yes	6		10
		6.4.2015					
8		13.4.2015	2	Yes	6		20
9		20.4.2015	2	Yes	6		20
10		27.4.2015	2	Yes	6		20
11		4.5.2015	2	Yes	6		20
12		11.5.2015	2	Yes	6		20
13		18.5.2015	2	Yes	6		
PRACTICE FINAL TEST		25.5.2015 Zápočtový test					
SUM			26		72		165
Total practice days		13					
Max Practice Score		263					

Check the full version [HERE!](#)

Test 02

Quick Warm up!

Find the test here (no-ads):

<http://goo.gl/jqYK6J>

Permanent link:

<https://docs.google.com/forms/d/1IUdk8voShpAafDKDVmsSDei913SkpxuIbzMGz5JjSrw/viewform>

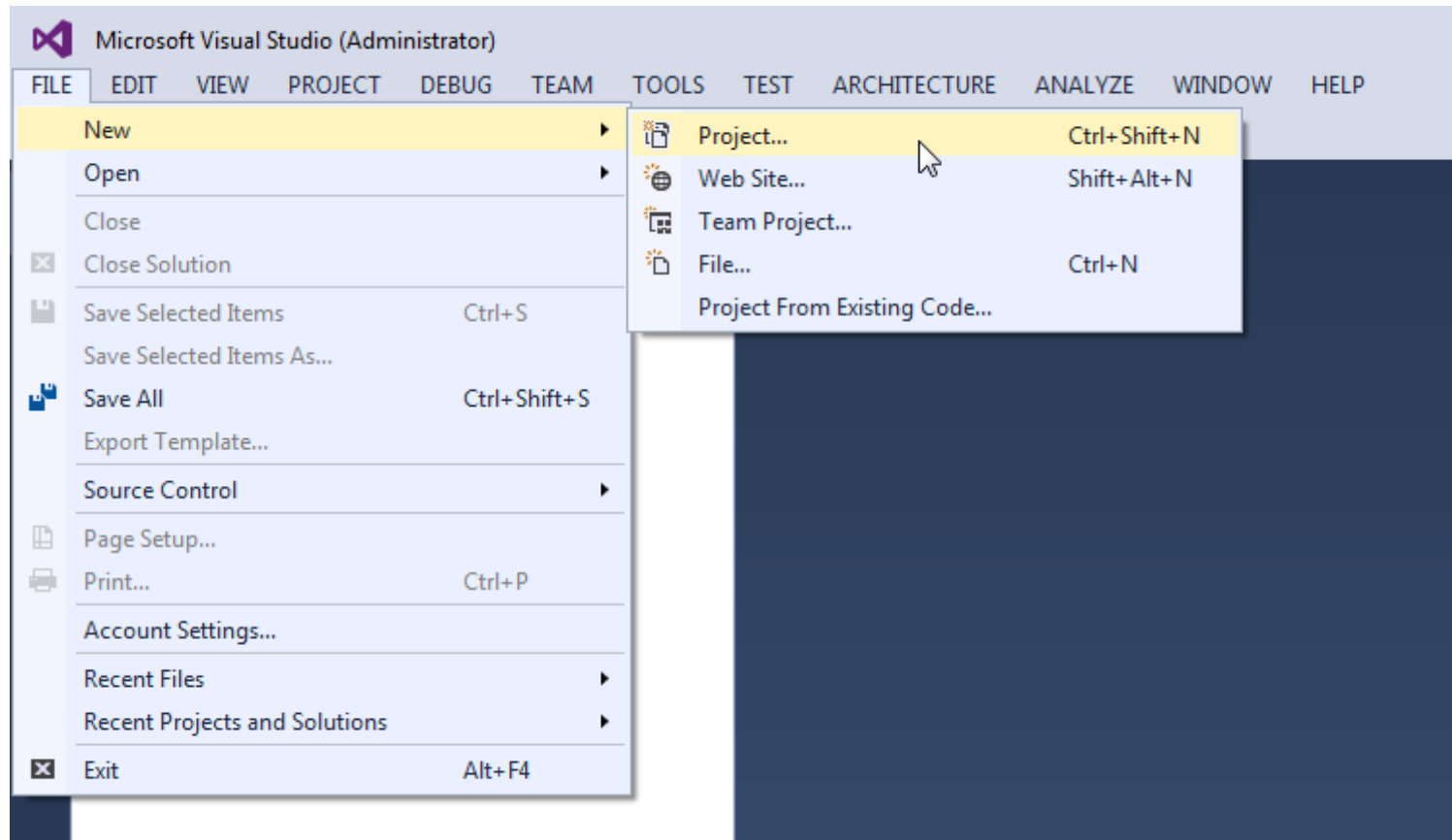
Time for the test:

10 min

C#

Visual Studio 2013

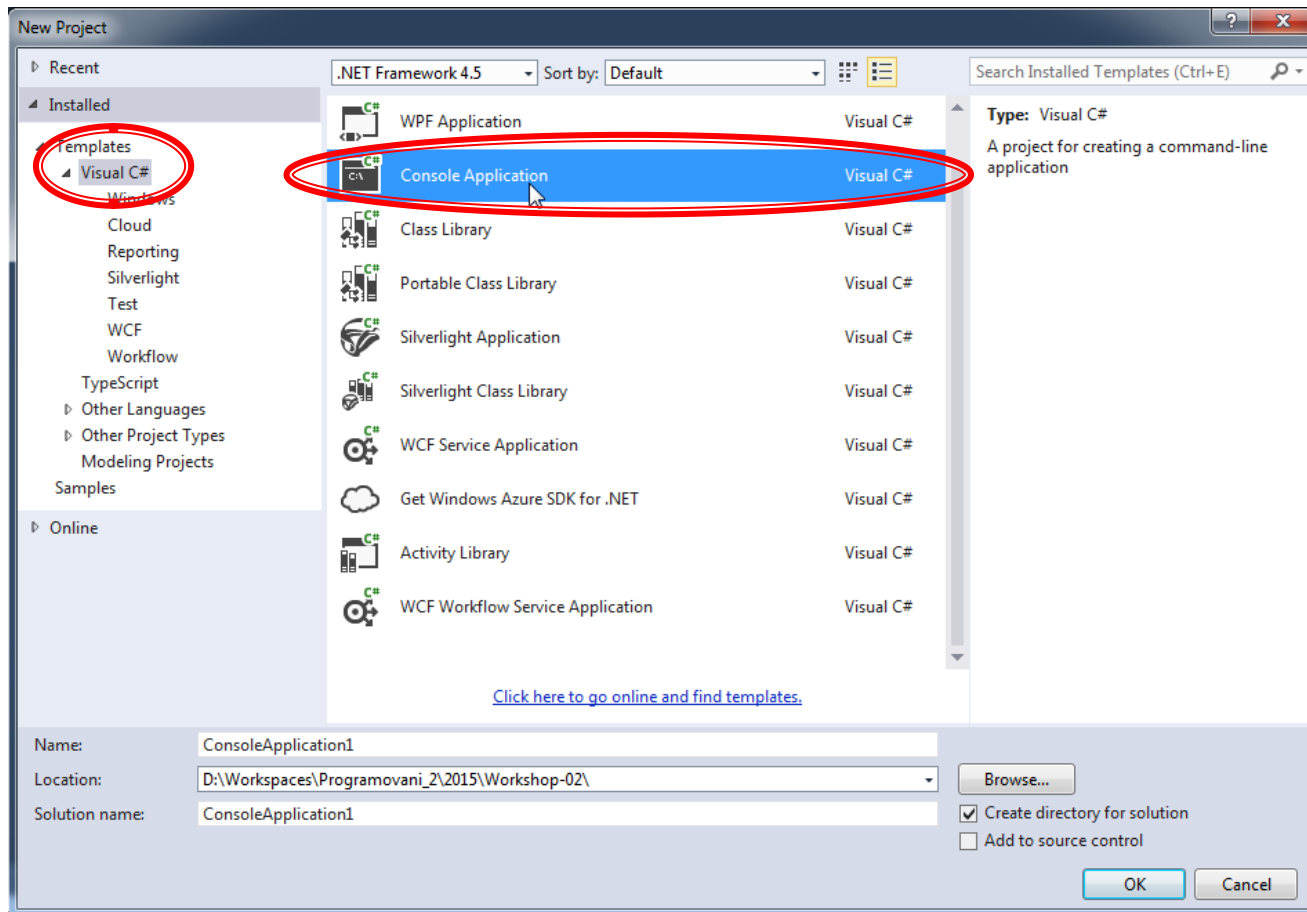
1. Creating new Solution and Project



C#

Visual Studio 2013

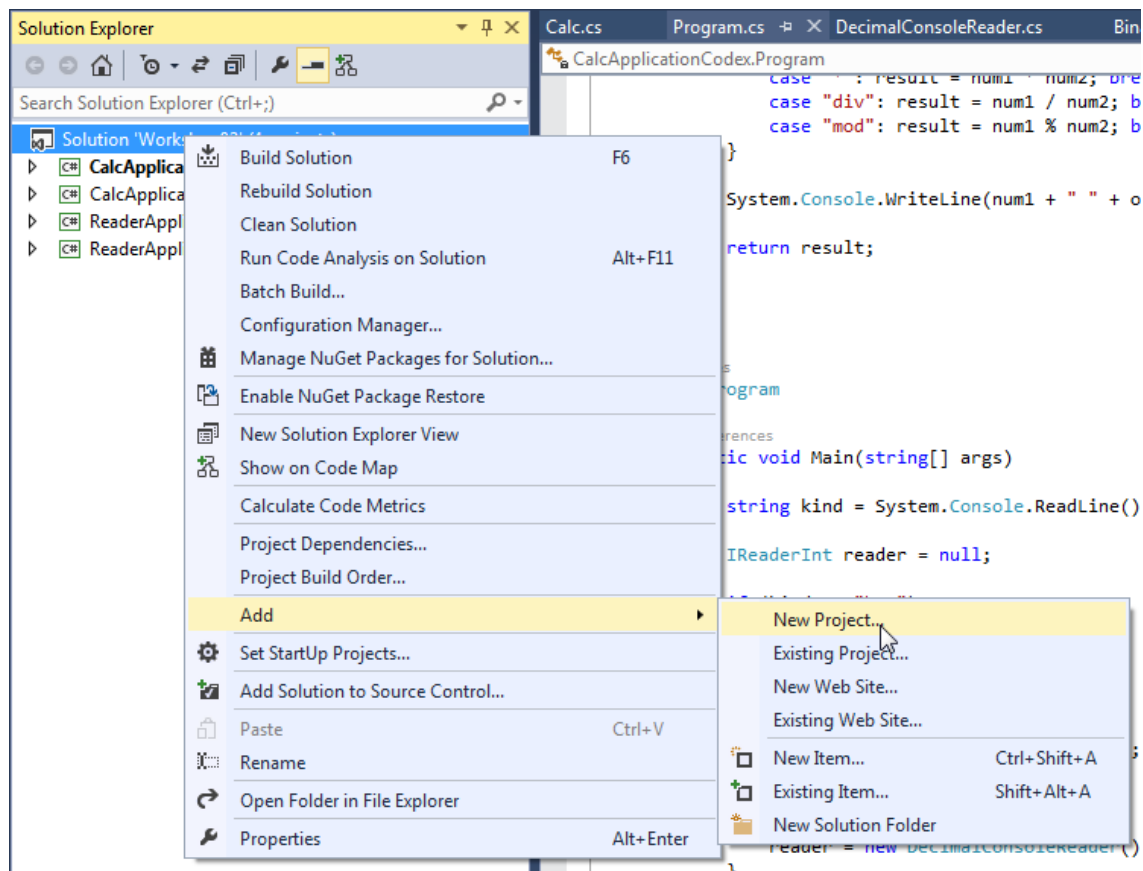
1. Creating new Solution and Project



C#

Visual Studio 2013

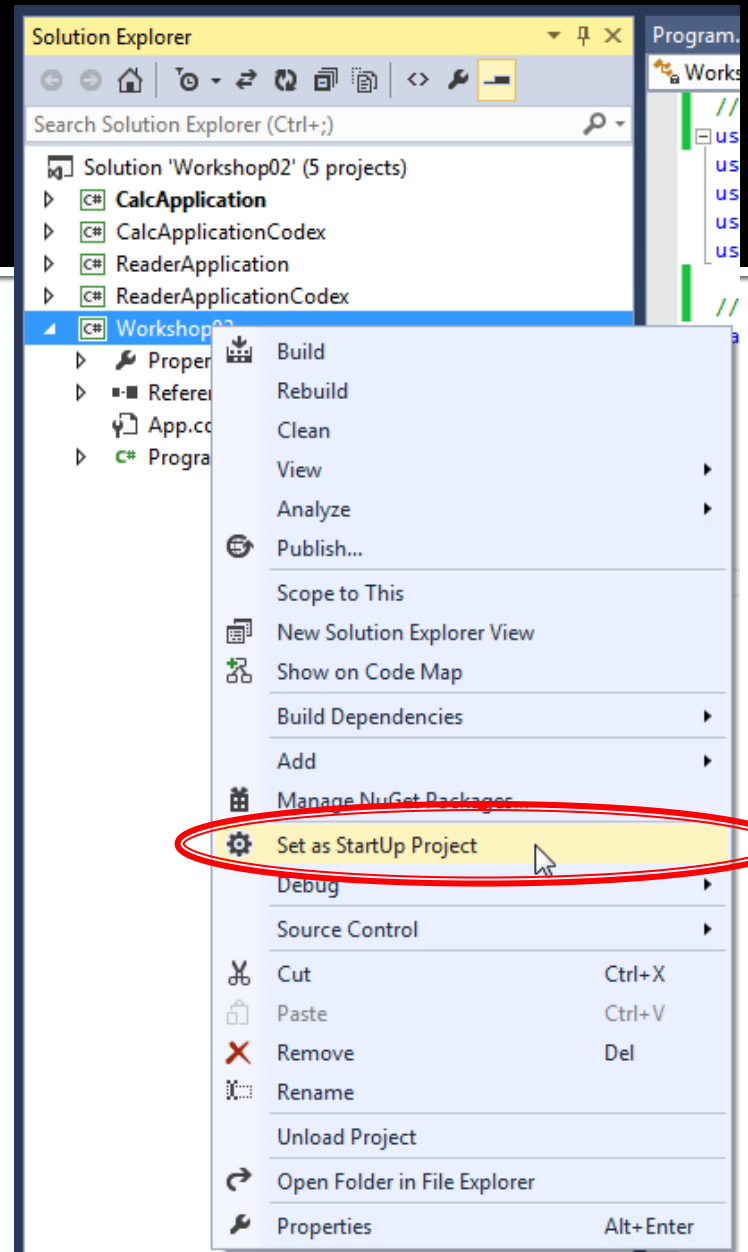
2. Adding new Project into an existing Solution



C#

Visual Studio 2013

3. Changing "main" project



Hotkeys MSVC# 2010 Express / 2010 Ultimate / 2012 Ultimate

4. Hot-keys overview

- **Ctrl+Shift+N**: nový projekt
- **F5**, **Ctrl+F5**: spustit (debug, bez-ladění)
- **Shift+F5**: ukončit
- **F9**: breakpoint zapnout/vypnout
- **F10**, **F11**: krokování

- **F12**: volání funkce -> definice
- **F7**, **Shift+F7**: zdrojový kód, designer / **F7**
- **Ctrl+K,R**: nalézt všechny výskyty / **Shift+F12**
 - **Ctrl+W,S**: Solution Explorer / **Ctrl+Alt+R**
 - **Ctrl+W,C**: Class View / **Ctrl+Shift+C**
 - **Ctrl+W,X**: Toolbox / **Ctrl+Alt+X**
 - **Ctrl+W,T**: ToDo-list / **Ctrl+\, T**
- **Ctrl+D**:
 - **Ctrl+D,C**: Call Stack / **Ctrl+Alt+C** / **Alt+F7**
 - **Ctrl+D,W**: Watches / **Ctrl+Alt+W, 1..4**

.....

C#

Basics

Code wrapped within classes, which are wrapped within namespaces.

Specific method as application “entry point”, which is invoked as the first by .NET

```
// Which namespace are available
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

// Which namespace we're creating classes/interfaces/structs for
namespace Workshop02
{
    // Class definition
    0 references
    class Program
    {
        // Application entry point
        0 references
        static void Main(String[] args)
        {
            // Pascal: writeln(...);
            System.Console.WriteLine("You!");
            // Pascal: sort-of readln(...);
            System.Console.ReadLine();
        }
    }
}
```

C#

Declaring variables

“Standard” types and their nullable counterparts.

Why is there no “string?” ... ?

string is already a class!

```
int    wholeNumber = 20;  
float  floatNumber = 0.5f;  
char   character   = 'A';  
string stringVar   = "Ahoj";  
bool   trueFalse   = false;
```

```
int?   wholeNumberN = null;  
float? floatNumberN = null;  
char?  characterN   = null;  
bool?  trueFalseN   = null;
```

C#

Reading/Writing from/to console

Sort of writeln() but
different
readln()...

```
// Pascal: writeln(...);  
System.Console.WriteLine("You!");  
// Pascal: sort-of readln(...);  
string line = System.Console.ReadLine();  
  
// Converting String to int/float, accessing respective chars  
int    wholeNumber = int.Parse(line);  
float  floatNumber = float.Parse(line);  
char   character   = line.Length > 0 ? line[0] : '?';
```

C#

Functions

```
<return type> <function name>(<parameters>) {  
    <function body>  
    return <value of return type>;  
}
```

0 references

```
int readNumber_1()  
{  
    string line = System.Console.ReadLine();  
    int number = int.Parse(line);  
    return number;  
}
```

0 references

```
int readNumber_2()  
{  
    return int.Parse(System.Console.ReadLine());  
}
```

C#

Task 2.1.1

- Create a simple number reader
- Use functions only

This simple reader should be able to read numbers in binary, decimal and hexadecimal numeral systems. Hexadecimal system may mix lower and upper case symbols for respective digits (e.g. 'aAbB' is a valid number).

Input:

('bin' | 'dec' | 'hex')

<number in given system>

Output:

<number in decimal system>

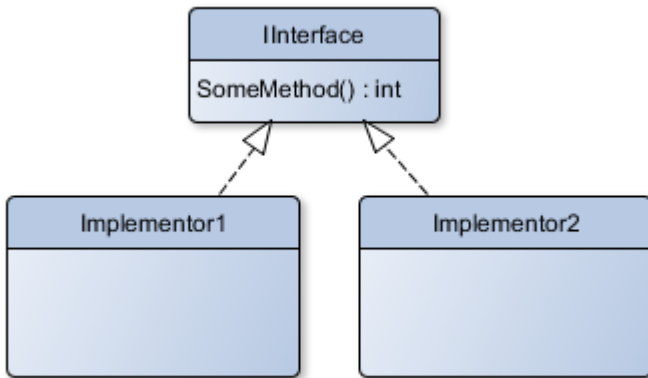
Examples:

No.	Input	Output
1	bin 111	7
2	dec 111	111
3	hex 111	273
4	hex Ff	255

C#

Interfaces, Classes

Hiding concrete implementation behind common interface.



UML

```
2 references
interface IInterface
{
    0 references
    int SomeMethod();
}

0 references
class Implementor1 : IInterface
{
    0 references
    int SomeMethod()
    {
        return 0;
    }
}

0 references
class Implementor2 : IInterface
{
    0 references
    int SomeMethod()
    {
        return 1;
    }
}
```

```
static void Main(String[] args)
{
    IInterface iface;

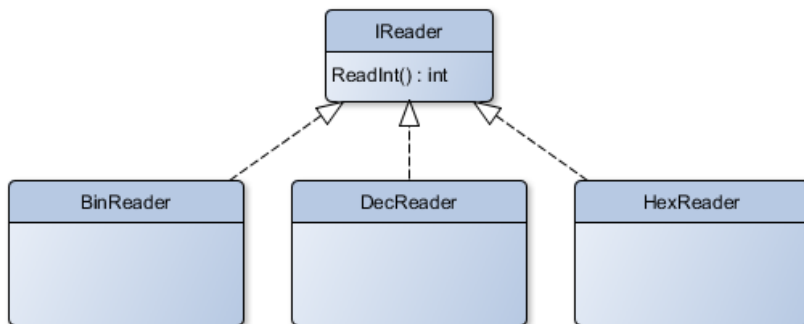
    iface = new Implementor1();
    System.Console.WriteLine(
        iface.SomeMethod()
    );

    iface = new Implementor2();
    System.Console.WriteLine(
        iface.SomeMethod()
    );
}
```


C#

Task 2.1.2

- Create common `IReader` interface
- Provide different implementations



This simple reader should be able to read numbers in binary, decimal and hexadecimal numeral systems. Hexadecimal system may mix lower and upper case symbols for respective digits (e.g. 'aAbB' is a valid number).

Input:

('bin' | 'dec' | 'hex')

<number in given system>

Output:

<number in decimal system>

Examples:

No.	Input	Output
1	bin 111	7
2	dec 111	111
3	hex 111	273
4	hex Ff	255

C#

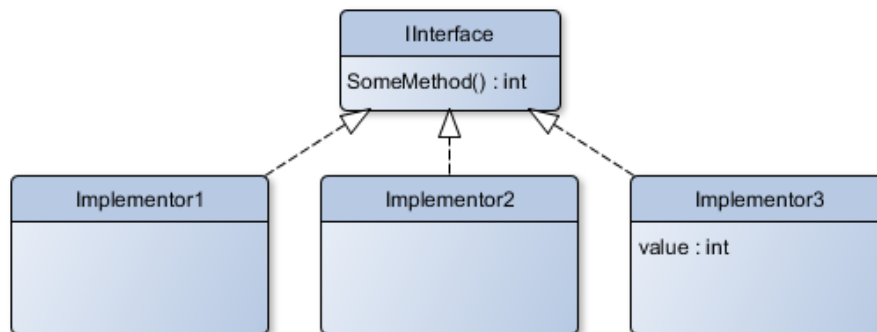
Constructor

Enforcing correct initialization of the object.

```
class Implementor3 : IInterface
{
    private int value;

    2 references
    public Implementor3(int value)
    {
        this.value = value;
    }

    0 references
    int SomeMethod()
    {
        return value;
    }
}
```



UML

```
static void Main(String[] args)
{
    IInterface iface;

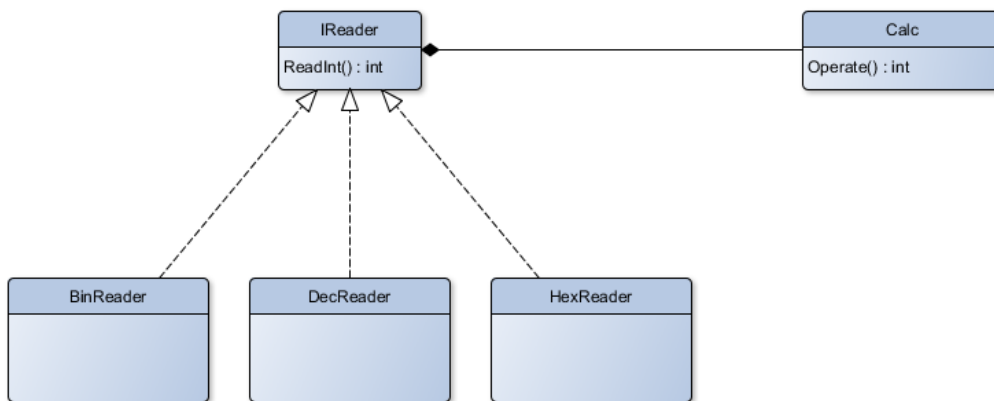
    iface = new Implementor3(10);
    System.Console.WriteLine(
        iface.SomeMethod()
    );

    iface = new Implementor3(20);
    System.Console.WriteLine(
        iface.SomeMethod()
    );
}
```

C#

Task 2.2

- Create Calc class that will internally use some IReader



UML - Composition

This Calc is simple extension to the previous Reader. You should create another class, that should be initialized with a correct reader (according to the input) and implement `+`, `-`, `*`, `div`, `mod` operations. Again you should be able to read numbers in binary, decimal and hexadecimal numeral systems. Hexadecimal system may mix lower and upper case symbols for respective digits (e.g. 'aAbB' is a valid number).

Input:
(`'bin' | 'dec' | 'hex'`)
<number in given system>
(`'+' | '-' | '*' | 'div' | 'mod'`)
<number in given system>

Output:
<first number in decimal system> <given operator> <second number in decimal system> = <result>

WARNING: The output **MUST HAVE** single whitespace around operator and around '='. E.g.: "10 + 10 = 20" is correct, "10+10=20" is incorrect.

Examples:

No.	Input	Output
1	bin 111 - 101	7 - 5 = 2
2	dec 111 + 1	111 + 1 = 112
3	hex 111 * 11	273 * 17 = 4641
4	hex eE - Ff	eE - fF = -17

Assignment 2

Send me an email if you haven't already

- Email: jakub.gemrot@gmail.com
- Subject: **Programming II – 2015 – Assignment 02**
- Content:
 - Your name
 - Your CUNI number
 - Your CoDex nick
 - I will add you into a CodEx group
- Create implementations for **tasks 2.1, 2.2** (if you haven't already) and submit them to **CodEx**
- Deadline: **1.3.2015 23:59**

Questions?

I sense a soul in search of answers...

- Sadly, I do not own the patent for perfection (and will never do)
- In case of doubts about the assignment or some other problems don't hesitate to contact me!
 - Jakub Gemrot
 - jakub.gemrot@gmail.com