

CSCI 1301 - Lab 17

Clément Aubert

March 13, 2018

Part I - User Input Validation

Integer Validation

Consider the code we just studied:

```
Console.WriteLine("Please enter a positive number");
int n = int.Parse(Console.ReadLine());
while (n < 0)
{
    Console.WriteLine($"You entered {n}, I asked you for a positive number. Please try
↪ again.");
    n = int.Parse(Console.ReadLine());
}
```

- Adapt it so that the user will be asked to enter an integer between 0 and 100, and asked again as long as (s)he doesn't comply.
- Do the same, but with even numbers.

String Validation

Adapt the code above to perform string validation: ask the user to enter a string, and as long as the user does not enter "Yes" or "No", ask him/her again to enter a value.

Part II - First While Loops

- Write a while loop that display the "*" character 100 times at the screen.
- Modify your previous loop, so that a new line character is displayed on the screen every time 10 "*" has been displayed on the screen. That is, your program should display on the screen:

```
*****
*****
```

```
*****  
*****  
*****  
*****  
*****  
*****  
*****  
*****  
*****
```

Part III - TryParse Method

Getting Familiar With It

Consider the code we just studied:

```
Console.WriteLine("Please, enter an integer.");  
string message = Console.ReadLine();  
int a;  
bool res = int.TryParse(message, out a);  
if (res)  
{  
    Console.WriteLine($"The value entered was an integer: {a}.");  
}  
else  
{  
    Console.WriteLine("The value entered was not an integer, so 0 is assigned to a.");  
}  
Console.WriteLine(a);
```

What happen if:

- The user enters an integer?
- The user enters a floating-point value?
- The user enters nothing?
- The user enters a string that contains alphabetical characters?

Using It

Write a code that ask the user to enter an integer, and ask the user again as long as the user entered something that isn't an integer.

Part IV - Pushing Further

The following are two independent tasks, to widen your understanding of this class, and to prepare you for the next labs.

- a. Actually, there is a `TryParse` method in other classes as well: there is for instance a `Double.TryParse` and a `Decimal.TryParse` method. Write a small program that uses one of them.
- b. Write a static method that takes a string as an argument, and return a boolean: `true` if the string given as a parameter is an integer, `false` otherwise. You should use `TryParse` in the body of your method. Note that you can define a static method in the same file as your `Main` method:

```
class Program
{

    static bool MyMethod(string param)
    {
        // Code goes here, we just return false to avoid an error message:
        return false;
    }
    static void Main()
    {
        Console.WriteLine(MyMethod("Test"));
    }
}
```