**Kind:** Assignment 2

**Team Size:** 1

**Hand-in:** Week 41 (October 13, 2016), end of session

**Task Code: NSTL**

# Preamble

In this assignment, you work individually. The main topics of this assignment are:

* Software Development
* Software Test, in particular: Iterative Test-Last Development

In this task, you will have to develop a software solution and have to provide proper test environment. The task is time-boxed, and it is not expected that you provide a 100% solution. Do as much as you can. **BUT:** Regardless of the amount of work you manage in the given time, you have to submit a solution that compiles!

**Expected Deliverables:**

Your submission to Blackboard consists of a single zip file containing your solution:

* Naming: [no]-[login]-NSTL.zip

[no] means the number you have drawn, [login] means your BB login/e-mail name, such that we can find you on BB (just in case), for example: “33-mylogin-NSTL.zip”. It is essential that you follow this naming schema.

**Tip:**

In the Visual Studio environment, go to the “View” menu, and select “Task List”. The respective dialog helps you navigating the code in the solution scaffold.

**Task Code: NSTL**

**In order to pass this assignment, the delivery must be complete in terms of the code must compile! Further detailed instructions are in the task description.**

**After the submission of your file, don’t leave the room. There will be another questionnaire to answer. You can leave when the solution was shipped and the questionnaire is done.**

# Implement the Bowling Score Keeper

The objective is to develop an application that can calculate the score of a single bowling game. There is no graphical user interface. You work only with objects and NUnit test cases in this assignment. You won’t need a main method.

# Detailed Setup and Project Description

## Project Template

You are provided with a project template that contains three classes: Frame, BowlingGame and BowlingException. The template contains some fields and methods. You may add additional fields and methods. However, **DO NOT CHANGE** the names of the existing fields and methods. Also, the parameters and the return types of the methods in the template **SHOULD NOT BE CHANGED**.

Your program should throw BowlingException in all error situations.

You should follow *Iterative Test-last Development* method as shown in the following diagram:



Figure 1 Iterative Test-Last Development

## Game Description

The game consists of 10 frames as shown below. In each frame, the player has two opportunities to knock down 10 pins. The score for the frame is the total number of pins knocked down, plus bonuses for strikes and spares.

A spare is when the player knocks down all 10 pins in two throws. The bonus for that frame is the number of pins knocked down by the next throw. So in frame 3 of the example game below, the score is 10 (the total number knocked down) plus a bonus of 5 (the number of pins knocked down on the next throw.)

A strike is when the player knocks down all 10 pins on his first try. The bonus for that frame is the value of the next two throws.

In the tenth frame, a player who rolls a spare or strike is allowed to have bonus throws to complete the frame. However, no more than three balls can be rolled in tenth frame.



The preceding scorecard shows an example game. Each box shows a frame where the first row contains the number of pins knocked down in the first and second throws and the second row shows the game score at that point in time. (See further explanations below):

* In the first frame, the player knocked down 1 pin with the first throw and 4 more with the second. [1,4] denotes the pins knocked down in the first and second throw of the frame. The score of the frame is 1+4=5.
* The score of the second frame ([4,5]) is 4+5=9. That makes the game score 14, the sum of the first two frame scores.
* The third frame [6,4] is a spare. Frame score including the bonus points cannot be calculated for this frame until the next ball is rolled. In the fourth frame [5, 5], 5 balls are knocked down in the first throw. So, the score of third frame is calculated as 10 + 5 = 15.
* Frame 5 [10, 0] is a strike. It’s score including the bonus points, after the next two throws can be calculated as 10 + 0 + 1 = 11.
* Ninth frame is a strike. It’s score, after the next two throws can be calculated as 10 + 2 + 8 = 20.
* The last frame [2,8] is a spare. Therefore, the game has one bonus throw. 6 pins are knocked down in bonus throw. So, the score of the last frame is 10 + 6 = 16.