Test-Driven Development Exercise: Bowling Score Keeper

Personal Software Process for Engineers

# 

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

The application’s requirements are divided into a set of user stories, which serves as your to-do list. You should be able to incrementally develop a complete solution without an upfront comprehension of all the game’s rules. Don’t read ahead, and handle the requirements one at a time in the order provided. Solve the problem using TDD, starting with the first story’s requirement.

Remember to always lead with a test case, taking hints from the examples provided. Move on to the next one only when a story is done. A story is done when you are confident that your program correctly implements all the functionality stipulated by the story’s requirement. This implies that all of your test cases for that story and all of the test cases for the previous stories pass. You may need to tweak your solution as you progress toward more advanced requirements.

1. **Frame**

*Each turn of a bowling game is called a* ***frame****. 10 pins are arranged in each frame. The player’s goal is to knock down as many pins as possible in each frame. The player has two chances, or* ***throws****, to do so. The value of a throw is the number of pins knocked down in that throw.*

**Requirement:** Define a frame as composed of two throws. The first and second throws should be distinguishable.

**Example:** [2, 4] is a frame with two throws, in which two pins were knocked down in the first throw and four pins were knocked down in the second throw.

1. **Frame Score**

*An ordinary frame’s score is the sum scores from each of its throws.*

**Requirement:** Compute the score of an ordinary frame.

**Examples:** The score of the frame [2, 6] is 8. The score of the frame [0, 9] is 9.

1. **Game**

*A single game consists of 10 frames.*

**Requirement:** Define a game, which consists of 10 frames.

**Example:** The sequence of frames [1, 5] [3, 6] [7, 2] [3, 6] [4, 4] [5, 3] [3, 3] [4, 5] [8, 1] [2, 6] represents a game.

1. **Game Score**

*The score of a bowling game is the sum of the individual scores of its frames.*

**Requirement:** Compute the score of a game.

**Example:** The score of the game [1, 5] [3, 6] [7, 2] [3, 6] [4, 4] [5, 3] [3, 3] [4, 5] [8, 1] [2, 6] is 81.

1. **Strike**

*A frame is called a* ***strike*** *if all 10 pins are knocked down in the first throw. In this case, there is no second throw. A strike frame can be written as [10, 0]. The score of a strike equals 10 plus the sum of the next two throws of the subsequent frame.*

**Requirement:** Recognize a strike frame. Compute the score of a strike. Compute the score of a game containing a strike.

**Examples:** Suppose [10, 0] and [3, 6] are consecutive frames. Then the first frame is a strike, and its score equals 10 + 3 + 6 = 19. The game [10, 0] [3, 6] [7, 2] [3, 6] [4, 4] [5, 3] [3, 3] [4, 5] [8, 1] [2, 6] has a score of 94.

1. **Spare**

*A frame is called a* ***spare*** *when all 10 pins are knocked down in two throws. The score of a spare frame is 10 plus the value of the first throw from the subsequent frame.*

**Requirement:** Recognize a spare frame. Compute the score of a spare. Compute the score of a game containing a spare frame.

**Examples:** [1, 9], [4, 6], and [7, 3] are all spares. If you have two frames [1, 9] and [3, 6] in a row, the spare frame’s score is 10 + 3 = 13. The game [1, 9] [3, 6] [7, 2] [3, 6] [4, 4] [5, 3] [3, 3] [4, 5] [8, 1] [2, 6] has a score of 88.

1. **Strike and Spare**

*A strike can be followed by a spare. The strike’s score is not affected when this happens.*

**Requirement:** Compute the score of a strike when it’s followed by a spare. Compute the score of a game with a spare following a strike.

**Examples:** In the sequence [10, 0] [4, 6] [7, 2], a strike is followed by a spare. In this case, the score of the strike is 10 + 4 + 6 = 20, and the score of the spare is 4 + 6 + 7 = 17. The game [10, 0] [4, 6] [7, 2] [3, 6] [4, 4] [5, 3] [3, 3] [4, 5] [8, 1] [2, 6] has a score of 103.

1. **Multiple Strikes**

*Two or more strikes in a row are possible. When this happens, the first strike’s score requires the values of throws from two subsequent frames.*

**Requirement:** Compute the score of a strike that is followed by another strike. Compute the score of a game with two strikes in a row.

**Examples:** In the sequence [10, 0] [10, 0] [7, 2], the score of the first strike is 10 + 10 + 7 = 27. The score of the second strike is 10 + 7 + 2 = 19. The game [10, 0] [10, 0] [7, 2] [3, 6] [4, 4] [5, 3] [3, 3] [4, 5] [8, 1] [2, 6] has a score of 112.

1. **Multiple Spares**

*Two spares in a row are possible. The first spare’s score is not affected when this happens.*

**Requirement:** Compute the score of a game with two spares in a row.

**Example:** The game [8, 2] [5, 5] [7, 2] [3, 6] [4, 4] [5, 3] [3, 3] [4, 5] [8, 1] [2, 6] has a score of 98.

1. **Spare as the Last Frame**

*When a game’s last frame is a spare, the player will be given a bonus throw. However, this bonus throw does not belong to a regular frame. It is only used to calculate the score of the last spare.*

**Requirement:** Compute the score of a spare when it’s the last frame of a game. Compute the score of a game when its last frame is a spare.

**Example:** The last frame in the game [1, 5] [3, 6] [7, 2] [3, 6] [4, 4] [5, 3] [3, 3] [4, 5] [8, 1] [2, 8] is a spare. If the bonus throw is [7], the last frame has a score of 2 + 8 + 7 = 17. The game has a score of 90.

1. **Strike as the Last Frame**

*When a game’s last frame is a strike, the player will be given two bonus throws. However, these two bonus throws do not belong to a regular frame. They are only used to calculate score of the last strike frame.*

**Requirement:** Compute the score of a spare when it’s the last frame of a game. Compute the score of a game when the last frame is a strike.

**Example:** The last frame in the game [1, 5] [3, 6] [7, 2] [3, 6] [4, 4] [5, 3] [3, 3] [4, 5] [8, 1] [10, 0] is a strike. If the bonus throws are [7, 2], the last frame’s score is 10 + 7 + 2 = 19. The game’s score is 92.

1. **Bonus Is a Strike**

*Further bonus throws are not granted when a game’s last frame is a spare and the bonus throw is a strike.*

**Requirement:** Compute the score of a game in which the last frame is a spare and the bonus throw is a strike.

**Example:** In the game [1, 5] [3, 6] [7, 2] [3, 6] [4, 4] [5, 3] [3, 3] [4, 5] [8, 1] [2, 8], the last frame is a spare. If the bonus throw is [10], the game’s score is 93.

1. **Best Score**

*A perfect game consists of all strikes (a total of 12 including the bonus throws) and has a score of 300.*

**Requirement:** Check that the score of a perfect game is 300.

**Example:** A perfect game looks like [10, 0] [10, 0] [10, 0] [10, 0] [10, 0] [10, 0] [10, 0] [10, 0] [10, 0] [10, 0] with bonus throws [10, 10]. It’s score is 300.

1. **Real Game**

**Requirement:** Check that the score of the game [6, 3] [7, 1] [8, 2] [7, 2] [10, 0] [6, 2] [7, 3] [10, 0] [8, 0] [7, 3] [10] is 135.

Congratulations, you are done!

Document Markings

Copyright 2020 Carnegie Mellon University. All rights reserved.  
  
This material is based upon work funded and supported by the Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.  
  
Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Department of Defense.  
  
NO WARRANTY. THIS MATERIAL IS FURNISHED ON AN “AS-IS” BASIS WITH NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, ANY WARRANTY WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT, OR THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.  
  
[Distribution Statement A] This material has been approved for public release and unlimited distribution. The United States Government has Unlimited Rights in this material as defined by DFARS 252.227-7013.

The text and illustrations in this material are licensed by Carnegie Mellon University under a [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/).

The Creative Commons license does not extend to logos, trade marks, or service marks of Carnegie Mellon University.