Bowling League

Introduction

This exercise gives you an opportunity to demonstrate your Java coding skills. You are to develop an application which manages bowling games. A major component is the scoring of a bowling game, other functionalities address the management of a whole game, multiple games, up to a whole league with rankings.

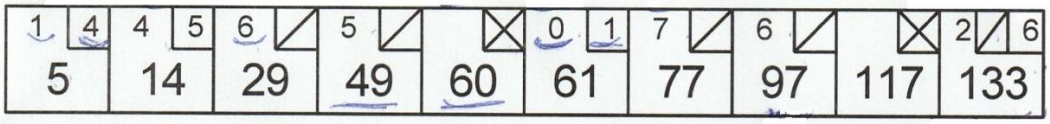
Game Rules

This section details the game rule variant to be used for this exercise.

A Bowling Game is played by two or more players (for this exercise the number of players is limited to six). A game is played in 10 turns. The turn of one player is called a *frame*, and the player throws the ball one or two times in a frame (there is an exception: the last frame can consist of three throws, see below). A player throws the first ball, and the number of pins knocked down is determined. There are 10 pins. If all 10 pins have been knocked down, it's called a *strike*, and the frame is completed – the next player takes turn. If one or more pins remain, the player throws a second time for the remaining pins. If all remaining pins could be knocked down with the second throw, it's called a *spare*.

The score of a player is determined by the number of pins knocked down. In case of a spare, the next throw result is added to the current frame; in case of a strike, the next two throw results are added. So if a player made a spare or strike, the score can't be calculated until the next one or two throws of this player. To reward a spare or strike in the last frame, one (for a spare) or two (for a strike) additional throws is/are permitted there.

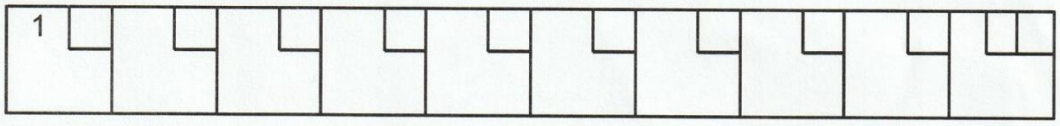
The following diagram shows the typical way to track a player's game score.



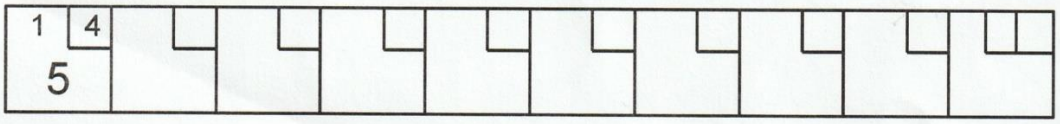
The throw results are written in the upper half of each square. A spare is symbolised by a diagonal line, a strike is marked by a cross. The lower half holds the current game score.

The diagram above shows a completed game. Let's take a look how the score developed during the game.

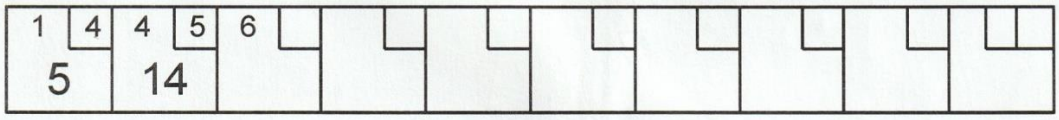
In the first throw, only one pin has been knocked down. The score of the frame remains blank – the frame is still open.



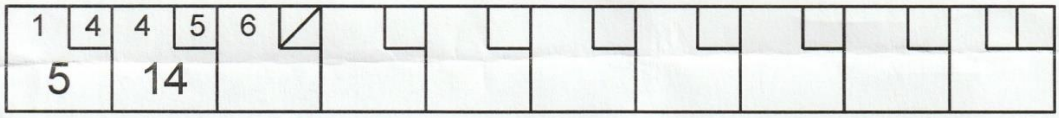
The frame score can be determined after the second throw.



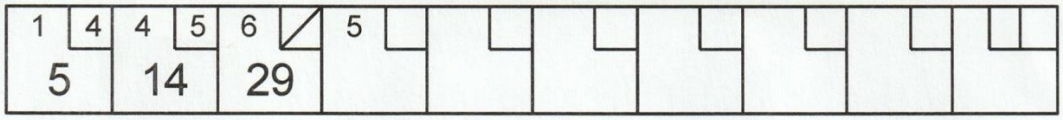
The second frame is handled the same way, just the numbers of pins knocked down differ. The first throw in the third frame is a 6.



The player succeeds to knock down the remaining 4 pins with his seconds throw – it's a spare. Because the result of the next throw (of the next frame) is still missing, the frame score can't be calculated yet.



When the player takes turn in the next round, he throws a 5. Now the preceding frame can be calculated (14 + 10 + 5 = 29).



This deferred scoring shown here for spares applies to strikes accordingly – only the next two throws have to be available.

Your Task

Develop a Java application which manages a bowling game. Two to six players can sign up for a game. They take turn for a frame and throw one or two (or three – in the last frame) balls. The application informs which player is to take turn and asks for the number of pins knocked down in one throw by user input. The application controls the game flow, i.e. if the player has to throw a second (or third) time, or if the next player takes turn. After each throw, the score is to be shown by the program. A simple text output is sufficient, a graphical output resembling the scoring diagrams above is a bonus task.

Time for the exercise: 90 minutes.

Environment: Eclipse, JDK 1.6