Bowling Game Code Explanation

Table of content

HOW TO BUILD AND RUN APPLICATION

1. Prerequisites:
   1. Apache maven 3.5.+
   2. JDK 1.8
2. Extract zipped file into a directory named <BOWLING_HOME> located in *bowling* directory int the zipped file.
3. Execute following command to build it (If you want to run jar located inside zipped file directly skip this step):

   ```
   > cd <BOWLING_HOME>
   > mvn package -e
   ```

4. Move to directory in which you put the jar file (If you have built it with step 3 move to <BOWLING_HOME>/target) and run following command:

   ```
   > java -jar bowling-1.0-SNAPSHOT-jar-with-dependencies.jar
   ```

5. Enjoy the game. 😊

Game Instruction:

- Bowling is played by throwing a ball down a narrow alley toward ten wooden pins. The object is to knock down as many pins as possible per throwing.
- The game is played in ten frames. At the beginning of each frame, all ten pins are set up. The player then gets two tries to knock them all down.
- If the player knocks all the pins down on the first try, it is called ‘STRIKE’ and the frame ends.
- If the player fails to knock down all the pins with the first ball, but succeeds with the second ball, it is called a ‘SPARE’
- After the second ball of frame, the frame ends even if there are pins standing.
- A STRIKE frame is scored by adding ten, plus the number of pins knocked down by the next two balls, to the score of the previous frame.
- A SPARE is scored by adding ten, plus number of pins knocked down by the next ball, to the score of previous frame.
- Otherwise, a frame is scored by adding the number of pins knocked down by two balls in the frame to the score of previous frame.
- If STRIKE is thrown in the tenth frame, then the player may throw two more balls to complete the score of strike.
- Likewise, if a spare is thrown in the tenth frame, a player may throw one or more ball to complete the score of the spare.
- Thus the tenth frame may have three balls instead of two.

HOW TO ANALYZE THE GAME INSTRUCTION

By reading carefully the instruction following rules can be inferred:

- We would have a FRAME with following rules:
  - In each frame you have only two drop to knock down ten pins. If in the first try all pines knocked down, you have two bonus ball to get more score. If in second try all pines knocked down, you have one bonus ball to get more score. If in both tries we were unsuccessful to drop all pins, only number of knocked pins will be counted. So, following general formula can be driven to calculate score of a frame:

$$\underline{score} = \underline{score\ of\ previous\ frame + number\ of\ knocked\ pins + number\ of\ knocked\ pins\ by\ bonus\ balls}$$

  - For frames excluding the last frame, bonus balls are the same used for knocking pins in next frame. So maximum available of bonus for all frames excluding last frame, is ten. However, for the last frame if ‘STRIKE’ or ‘SPARE’ happens, all pins would be lined up again to be knocked by the ball, so maximum bonus for last frame is twice as much as other frames.

  - Each frame has following states:
    - WAITING: It’s not its turn to be played
    - PLAYING: There are still pins and player has a ball to be thrown
    - PLAYED: All two balls are thrown and not have all pins knocked
    - SPARE: SPARE happed at the second throw and it's waiting for all bonus needed to calculate the score
    - STRIKE: STRIKE happed at the first throw and no more throwing will be done in this frame and it’s waiting for all bonus needed to calculate it’s score.
    - SPARE_BONUS_RECEIVED: SPARE happed and all bonus are received and the score is calculable.
    - STRIKE_BONUS_RECEIVED: STRIKE happed and all bonus are received and the score is calculable.
  - Frame score is calculable if it is in one of the following states:
    - PLAYED
    - SPARE_BONUS_RECEIVED
    - STRIKE_BONUS_RECEIVED
  - Frame is playable if it is in one of the following states:
    - PLAYING
    - WAITING
- Game will finish if last frame is calculable.
- Player is able to enter number of knocked balls, based on number of available pins in each frame or available bonus dedicated.
- Scores must be calculated after each throwing from first frame to last frame respectively. Since the required condition, not enough, to calculate the frame score is calculable previous frame.

CODE DESCRIPTION

Based on rules described by previous section following entity are designed for application:

1. Thrower: By calling startThrowing, it will take number of available pins to be knocked.
2. Frame: Encapsulate all rules of frames and make it available for game instructor with following properties:
   - availablePins: Number of standing pins to be knocked
   - number: Frame's label.
   - score: Score of each Frame which will be calculated after the frame become calculable.
   - frameStatus: One of status mentioned in previous section.
   - playedBalls: List of balls thrown in this frame. Each ball retains number of pins which it knocked down.
   - bonusBalls: List of received bonus balls which are needed to calculate the score.
   - previousFrame: Reference to previousFrame to calculate it's score.
   - calculateScore(): Invoked to calculate the score based on whether it is calculable or not.
   - throwBall(): Invoked to knock pins based on whether the game is playable or not
   - deliverBonus(): Invoked to deliver bonus ball to the frame based on whether it is subscribed to be delivered or not
3. Ball: Entity which tells how many pins will be knocked after it is thrown
4. BowlingScoreBoardApp: The game instructor
   - Walks game to the last step which is having calculable last frame
   - Subscribes frames to receive bonus balls
   - Unsubscribes frames which are delivered suitable bonus base on whether it's on STRIKE or SPARE state
   - Calculates score of all calculable frames from the first to the last one respectively.

UNIT TESTS

The program contains following Unit Test Classes:

- TestBallClass: Create ball of specific number of knocked pins and test whether its internal state is equal to constructor input argument.
- TestFrameClass: Create frame and threw ball. In each test case it tries to examine whether the frame state is equal to the state which the frame must be in it according to the knocked pins or not. Test include all states of FrameState enum mentioned before, as well as invalid input numbers as knocked pins. *Invalid knocked pins* means that user has entered number greater available pins.
- ScoreBoardTestClass: Initiates game and read each round of game from CSV file from test resource directory. Each CSV files contains multiple lines. Each line contains three arguments and define one round throwing ball by user and describe status and score expected each frame must have after throwing.
  - First argument: Number of knocked pins
  - Second argument: List of states expected each frame must have after throwing ball.
  - Third argument: List of scores expected each frame must have after throwing ball.

EXAMPLE:

```
2, "PLAYING, WAITING, WAITING, WAITING, WAITING, WAITING, WAITING, WAITING, WAITING, WAITING", "0, 0, 0, 0, 0, 0, 0, 0, 0, 0"
8, "SPARE, WAITING, WAITING, WAITING, WAITING, WAITING, WAITING, WAITING, WAITING, WAITING", "0, 0, 0, 0, 0, 0, 0, 0, 0, 0"
6, "SPARE_BONUS_RECEIVED, PLAYING, WAITING, WAITING, WAITING, WAITING, WAITING, WAITING, WAITING, WAITING", "16, 0, 0, 0, 0, 0, 0, 0, 0, 0"
1, "SPARE_BONUS_RECEIVED, PLAYED, WAITING, WAITING, WAITING, WAITING, WAITING, WAITING, WAITING, WAITING", "16, 23, 0, 0, 0, 0, 0, 0, 0, 0"
4, "SPARE_BONUS_RECEIVED, PLAYED, PLAYING, WAITING, WAITING, WAITING, WAITING, WAITING, WAITING, WAITING", "16, 23, 0, 0, 0, 0, 0, 0, 0, 0"
5, "SPARE_BONUS_RECEIVED, PLAYED, PLAYED, WAITING, WAITING, WAITING, WAITING, WAITING, WAITING, WAITING", "16, 23, 32, 0, 0, 0, 0, 0, 0, 0"
3, "SPARE_BONUS_RECEIVED, PLAYED, PLAYED, PLAYING, WAITING, WAITING, WAITING, WAITING, WAITING, WAITING", "16, 23, 32, 0, 0, 0, 0, 0, 0, 0"
1, "SPARE_BONUS_RECEIVED, PLAYED, PLAYED, PLAYED, WAITING, WAITING, WAITING, WAITING, WAITING, WAITING", "16, 23, 32, 36, 0, 0, 0, 0, 0, 0"
0, "SPARE_BONUS_RECEIVED, PLAYED, PLAYED, PLAYED, PLAYING, WAITING, WAITING, WAITING, WAITING, WAITING", "16, 23, 32, 36, 0, 0, 0, 0, 0, 0"
10, "SPARE_BONUS_RECEIVED, PLAYED, PLAYED, PLAYED, SPARE, WAITING, WAITING, WAITING, WAITING, WAITING", "16, 23, 32, 36, 0, 0, 0, 0, 0, 0"
0, "SPARE_BONUS_RECEIVED, PLAYED, PLAYED, PLAYED, SPARE_BONUS_RECEIVED, PLAYING, WAITING, WAITING, WAITING, WAITING", "16, 23, 32, 36, 46, 0, 0, 0, 0, 0"
0, "SPARE_BONUS_RECEIVED, PLAYED, PLAYED, PLAYED, SPARE_BONUS_RECEIVED, PLAYED, WAITING, WAITING, WAITING, WAITING", "16, 23, 32, 36, 46, 46, 0, 0, 0, 0"
7, "SPARE_BONUS_RECEIVED, PLAYED, PLAYED, PLAYED, SPARE_BONUS_RECEIVED, PLAYED, PLAYING, WAITING, WAITING, WAITING", "16, 23, 32, 36, 46, 46, 0, 0, 0, 0"
2, "SPARE_BONUS_RECEIVED, PLAYED, PLAYED, PLAYED, SPARE_BONUS_RECEIVED, PLAYED, PLAYED, WAITING, WAITING, WAITING", "16, 23, 32, 36, 46, 46, 55, 0, 0, 0"
9, "SPARE_BONUS_RECEIVED, PLAYED, PLAYED, PLAYED, SPARE_BONUS_RECEIVED, PLAYED, PLAYED, PLAYING, WAITING, WAITING", "16, 23, 32, 36, 46, 46, 55, 0, 0, 0"
0, "SPARE_BONUS_RECEIVED, PLAYED, PLAYED, PLAYED, SPARE_BONUS_RECEIVED, PLAYED, PLAYED, PLAYED, WAITING, WAITING", "16, 23, 32, 36, 46, 46, 55, 64, 0, 0"
3, "SPARE_BONUS_RECEIVED, PLAYED, PLAYED, PLAYED, SPARE_BONUS_RECEIVED, PLAYED, PLAYED, PLAYED, PLAYING, WAITING", "16, 23, 32, 36, 46, 46, 55, 64, 0, 0"
5, "SPARE_BONUS_RECEIVED, PLAYED, PLAYED, PLAYED, SPARE_BONUS_RECEIVED, PLAYED, PLAYED, PLAYED, PLAYED, WAITING", "16, 23, 32, 36, 46, 46, 55, 64, 72, 0"
10, "SPARE_BONUS_RECEIVED, PLAYED, PLAYED, PLAYED, SPARE_BONUS_RECEIVED, PLAYED, PLAYED, PLAYED, PLAYED, STRIKE", "16, 23, 32, 36, 46, 46, 55, 64, 72, 0"
2, "SPARE_BONUS_RECEIVED, PLAYED, PLAYED, PLAYED, SPARE_BONUS_RECEIVED, PLAYED, PLAYED, PLAYED, PLAYED, STRIKE", "16, 23, 32, 36, 46, 46, 55, 64, 72, 0"
10, "SPARE_BONUS_RECEIVED, PLAYED, PLAYED, PLAYED, SPARE_BONUS_RECEIVED, PLAYED, PLAYED, PLAYED, PLAYED, STRIKE_BONUS_RECEIVED", "16, 23, 32, 36, 46, 46, 55, 64, 72, 94"
```

After six times of throwing ball which knocked 1 pin each states must have following:

- States

  SPARE_BONUS_RECEIVED, PLAYED, PLAYED, PLAYED, WAITING, WAITING, WAITING, WAITING, WAITING, WAITING

- Scores

  16, 23, 32, 36, 0, 0, 0, 0, 0, 0

TESTS TO COVER ALL STATES

1. Neither strike nor spare:

2 - 5	6 - 1	4 - 5	3- 1	0 - 1	0 - 0	7 - 2	9 - 0	3 - 5	5 - 2
7	14	23	27	28	28	37	46	54	61

2. Strike happened in the first frame:

10	6 - 1	4 - 5	3- 1	0 - 1	0 - 0	7 - 2	9 - 0	3 - 5	5 - 2
17	24	33	37	38	38	47	56	64	71

3. Strike happened in the middle frame:

2 - 5	6 - 1	4 - 5	3- 1	10	0 - 0	7 - 2	9 - 0	3 - 5	5 - 2
7	14	23	27	37	37	46	55	63	70

4. Strike happened in the last frame:

2 - 5	6 - 1	4 - 5	3- 1	0 - 1	0 - 0	7 - 2	9 - 0	3 - 5	10 - 2 - 10
7	14	23	27	28	28	37	46	54	76

5. Spare happened in the first frame:

2 - 8	6 - 1	4 - 5	3- 1	0 - 1	0 - 0	7 - 2	9 - 0	3 - 5	10 - 2 - 10
16	23	32	36	37	37	46	55	63	85

6. Spare happened in the middle frame:

2 - 8	6 - 1	4 - 5	3- 1	0 - 10	0 - 0	7 - 2	9 - 0	3 - 5	10 - 2 - 10
16	23	32	36	46	46	55	64	72	94

7. Spare happened in the last frame:

2 - 8	6 - 1	4 - 5	3- 1	0 - 1	0 - 0	7 - 2	9 - 0	3 - 5	8 - 2 - 10
16	23	32	36	37	37	46	55	63	83

8. Strike happened before spare in the first frame:

10	6 - 4	4 - 5	3- 1	0 - 10	0 - 0	7 - 2	9 - 0	3 - 5	10 - 2 - 10
20	34	43	47	57	57	66	75	83	105

9. Strike happened before spare in the middle frame:

10	6 - 4	4 - 5	3- 1	10	4 - 6	7 - 2	9 - 0	3 - 5	10 - 2 - 10
20	34	43	47	67	84	93	102	110	132

10. Strike happened before spare in the last frame:

10	6 - 4	4 - 5	3- 1	10	4 - 6	7 - 2	9 - 0	10	8 - 2 - 10
20	34	43	47	67	84	93	102	122	142

11. Spare happened before strike in the first frame:

6 - 4	10	4 - 5	3- 1	10	4 - 6	7 - 2	9 - 0	10	8 - 2 - 10
20	39	48	52	72	89	98	107	127	147

12. Spare happened before strike in the middle frame:

6 - 4	10	4 - 5	3- 1	4 - 6	10	7 - 2	9 - 0	10	8 - 2 - 10
20	39	48	52	72	91	100	109	129	149

13. Spare happened before strike in the last frame:

6 - 4	10	4 - 5	3- 1	4 - 6	10	7 - 2	9 - 0	6-4	10 - 1 - 8
20	39	48	52	72	91	100	109	129	148

14. Strike happened couple of times:

10	10	4 - 5	10	10	0 - 0	7 - 2	10	3 - 5	5 - 2
24	43	52	72	82	82	91	109	117	124

15. Spare happened couple of times:

5 - 5	6 - 4	4 - 5	3- 1	9 - 1	8 - 0	7 - 3	9 - 0	5 - 5	5 - 5 - 2
16	30	39	43	61	69	88	97	112	124

16. Spares and Strikes happened in any order

10	7-2	5 - 5	3-6	10	0 - 10	7 - 3	0 - 1	3 - 5	10 - 0 - 2
19	28	41	50	70	87	97	98	106	118

17. All spares

5-5	7-3	6 - 4	0 - 10	1-9	2 - 8	4 - 6	5 - 5	3 - 7	0 - 10 - 2
17	33	43	54	66	80	95	108	118	130

18. All strikes

10	10	10	10	10	10	10	10	10	10 - 10 - 10
30	60	90	120	150	180	210	240	270	300