Kata "Bowling"

# Aufgabe

Entwickle eine Klasse zur Zählung von Bowling-Spielen. Ein Bowling-Spiel besteht aus Würfen (engl. *roll*), um 10 Pins am Ende der Bowling-Bahn abzuräumen. Mit jedem Wurf können also 0 bis 10 Pins umgeworfen werden. Pro Runde hat ein Spieler max. 2 Würfe, um die 10 Pins abzuräumen. Würfe werden zu Rahmen (engl. *frame*) zusammengefasst, die jeweils eine Punktzahl zugeordnet bekommen. Ein Spiel besteht aus 10 Runden oder Rahmen. Wieviele Würfe pro Runde von einem Spieler gemacht werden dürfen, hängt davon ab, wieviele Pins mit ihnen geworfen werden:

* Wenn der erste Wurf eines Spielers alle 10 Pins abräumt (engl. *strike*), ist er der einzige Wurf in der Runde.
* Ansonsten enthalten die Rahmen immer zwei Würfe, jeder mit 0 bis 10 Pins.
  + Ausnahme: Wenn der 10. Rahmen als ersten Wurf einen strike enthält oder beide Würfe zusammen einen spare (s.u.) ergeben, dann kann ein drittes Mal geworfen werden.

Für die Berechnung der Punktzahl eines Rahmens gibt es einige Regeln:

* Ein Rahmen, dessen beide Würfe zusammen maximal 9 Pins gerissen haben, erhält die Summe der Pins als Punktzahl.
* Ein Rahmen dessen beide Würfe zusammen 10 Pins berissen haben (engl. *spare*), erhält als Punktzahl 10 + die Zahl der Pins des nächsten Wurfes.
* Ein Rahmen mit einem strike erhält als Punktzahl 10 + die Summe der Pins der nächsten beiden Würfe.

Bei Robert C. Martin[[1]](#endnote-1) findet sich diese Darstellung für ein Bowling-Spiel:



Jeder Rahmen zeigt den ersten und zweiten und beim letzten sogar den dritten Wurf mit ihren Pins sowie die kummulierte Punktzahl. Ist das Kästchen für den zweiten Wurf halb ausgefüllt, liegt ein *spare* vor, ist es ganz ausgefüllt, war der erste Wurf ein *strike*.

Die Schnittstellen der Klassen sollen wie folgt aussehen:

public class Game {

public void addRoll(final int pins) { ... }

public int getTotalScore() { ... }

public boolean isOver() { ... }

public Map<Integer, Frame> getFrames() { ... }

public String toString() { ... }

}

public class Frame {

// Punktzahl nur dieses Frame inkl. Punkte wegen Spare oder Spike

public int getScore() { ... }

public int[] getPinsRolled() { ... }

public String toString() { ... }

}

Die Methode *addRoll()* wirft eine Exception, falls nach Spielende noch Würfe registriert werden sollen. Zusätzliche Methoden sind möglich/nötig. Die obigen Methoden müssen jedoch erhalten bleiben.

Auf das Objekt *Game* angewendete Methoden liefern folgendes Beispiel:

|  |  |  |  |
| --- | --- | --- | --- |
| addRoll(x) | toString() | getTotalScore() | isOver() |
| 1 | ([1],1) | 1 | false |
| 4 | ([1,4],5) | 5 |  |
| 4 | ([1,4],5), ([4],4) | 9 |  |
| 5 | ([1,4],5), ([4,5],9) | 14 |  |
| 6 | ..., ([6],6) | 20 |  |
| 4 | ..., ([6,4],10) | 24 |  |
| 5 | ..., ([6,4],15), ([5],5) | 34 |  |
| 5 | ..., ([5,5],10) | 39 |  |
| 10 | ..., ([5,5],10), ([10],10) | 59 |  |
| 0 | ..., ([10],10), ([0],0) | 59 |  |
| 1 | ..., ([10],11), ([0,1],1) | 61 |  |
| 7 | ..., ([7],7) | 68 |  |
| 3 | ..., ([7,3],10) | 71 |  |
| 6 | ..., ([7,3],16), ([6],6) | 83 |  |
| 4 | ..., ([6,4],10) | 87 |  |
| 10 | ..., ([6,4],20), ([10],10) | 107 |  |
| 2 | ..., ([10],12), ([2],2) | 111 |  |
| 8 | ..., ([10],20), ([2,8],10) | 127 |  |
| 6 | ..., ([2,8,6],16) | 133 | true |

# Ziele

Im Vordergrund steht nicht die möglichst schnelle Fertigstellung einer Lösung. Elegante, lesbare Lösungen sind gefragt. Die unterschiedlichen Lösungen sollen diskutiert werden.

Weitere Ziele:

* TDD mit JUnit
* Listen mit Streams (Lambda) verarbeiten

1. Rober C. Martin, Bowling Game Kata, <http://butunclebob.com/files/downloads/Bowling%20Game%20Kata.ppt> [↑](#endnote-ref-1)