

Debugging

Testing is the process of running test cases to see whether errors can be uncovered. *Debugging* is the process of tracking down the cause of an error that has been uncovered and then fixing it. Debugging can be likened to the process that a detective goes through in looking for clues as to who committed a crime. Here, we discuss the process of debugging.

There are two useful ways to help track down an error. The first involves using a debugging tool that is built in to your IDE; the second involves using inserting print statements in order to get output that helps one hone in on the error. In this lecture, we'll use the print-statement route.

Bowling Primer

Each game of bowling consists of ten frames. In each frame a player has two chances to knock down all 10 pins by rolling his bowling ball at them (any pins knocked down in the first toss remain down for the second toss). If a bowler knocks down less than ten pins, they score 1 point for every pin they knock down, and the frame is recorded by putting the number of pins knocked down in the first toss next to the number of pins knocked down in the second toss (15, 33 and 06 all score 6 points). If a bowler knocks down all ten pins using both of their chances, it is called a spare, which is annotated on the score card by writing how many pins they knocked down on the first toss, then a slash (1/, 9/, 0/, etc). Whenever a bowler throws a spare they get bonus points equal to however many pins they knock down on their next toss (in addition to 10 points for knocking down 10 pins). If a bowler manages to knock down all ten pins on his first toss, they score what is a strike, annotated X, and receive bonus points equal to whatever they knock down on their next two throws. The highest number of points a bowler can score in a single frame is 30, if they throw three strikes in a row.

For the first 9 frames, a bowler scores their bonus points from strikes and spares based on how he performs in the next frame(s), but not so on the tenth and final frame. Instead, if a bowler gets a strike or spare in the tenth frame he gets to keep throwing balls until he has earned all his bonus points. This is why the tenth frame on a bowling score card has extra boxes, a bowler can throw 34-, 4/7, XX7, 5/X, depending on how they do on their bonus throws, if any.

Debugging

I have fairly exhaustive test cases, all of which are right, but I did not perform incremental testing. This means I can't be sure where any of the bugs I am going to find are located, but since this is such a small classes I hope that it won't be a real problem.

When I run my test cases I am not surprised that I failed both of them. Looking at both failed tests, I see that the second was in error in function toString, which is much simpler than function score, so I try to track that error down first.

Hmm... the number of pins knocked down on the third ball wasn't part of the output, could it be because I forgot to check if it's the tenth frame? I got lucky that time and spotted the error right away: I used less than when I meant greater than. Always be careful with conditions. I fix the error and recompile.

I test again knowing I'm going to fail, and hoping the next bug is as easy to track down. Now, since both errors are in the part where function score is called, the simpler problem to look at is the normal frame. Not seeing any errors immediately jumping out of my code, I insert a debug statement, that is a call to System.out.println("Debug!") to verify the branching code is working (that is trying to locate the source of the error). Recompiling, I get the same error, and I see in the interactions pane that "Debug!" was printed, so I know that code that looks right is executing. Since it still looks right, I output a few of the variables by changing the argument to System.out.println. Compile and fail the test case again, and I see that