General testing plan for the **Comment class**.

1. create five instances of the class Comment with different values for the variable “rating”. Two of the instances should have the rating on the edges of the limit. The other two should fall over the limit. And one should be within the limit range. Thus, the five instances to be created should be:

comment0 = newComment(author, text, 0)

comment1 = newComment(author, text, 1)

comment3 = newComment(author, text, 3)

comment 5 = newComment(author, text, 5)

comment 6 = newComment(author, text, 6)

1. call the method of interest on each of the instances;
2. check if the results match the predicted values;

For **method downVote**, the plan implementation would look as follows:

1. create the 5 instances of the class Comment using a fixture;
2. for each instance, call downVote method;
3. compare with the expected values:

comment0: 1 (since it is the lowest rating)

comment1: 1

comment3: 2

comment5: 4

comment6: 5 (even though the value of 6 is out of range, it is not the current method’s job to check for that, so it just makes the given value smaller by 1)

For method **getRating:**

1. create the 5 instances
2. call getRating
3. compare with expected:

comment0: 0

comment1: 1

comment3: 3

comment5: 5

comment6: 6

General testing plan for the class **SalesItems**:

1. create 4 instances of the class SalesItems, with different prices and different number of comments

mug (“Mug”, -1), 0 comments

time (“The concept of time”, 0), 2 comments

dollar (“One dollar”, 1), 1 comment

textbook (“Any university textbook you need”, 500000), 3 comments

1. call the method of interest on each instance
2. compare the results with the expected value