Liz Brondos

Reading Interaction #5

A customer walks into a DodgySoft store, very angry, asking to speak with a TechSupport representative.

Customer: What on earth is up with your lousy TechSupport system? Whatever happened to good old-fashioned Customer Support with real people serving the customers?

TechSupport programmer: Well, I am on the team who is working on the TechSupport program. Our business isn't doing too well, so we couldn't afford to pay workers to help the customers. Don't worry; it's only temporary until our sales pick up.

Customer: Well, I don't think TechSupport is going to help sales pick up at all!

TechSupport programmer: Do you know how hard it is to write a TechSupport program? Let me tell you. First of all, we need to know how to use the Java standard class library. For our TechSupport program, we create three classes, SupportSystem, InputReader and Responder. SupportSystem is the main class, InputReader gets input from the user, and the Responder generates a response.

Customer: Well, the Responder was an idiot! I wrote down my problem, and it told me, "That sounds interesting. Tell me more…" So I did, until I told it: "That's all. I have nothing more to tell." But it still responded with "That sounds interesting. Tell me more…" I responded: "You are a dumb computer who doesn't know anything" and it still wanted to hear more about that! I typed "Bye" and it wouldn't even exit. I had to type "bye" for it to work.

TechSupport programmer: I know, our first version was faulty like that. We had to look at the interface of the class, which describes what the class does and how it can be used. We also looked at the implementation of the class, which is the complete source code that defines that class. It gave us a better sense of what we were doing. Since strings are immutable objects because their contents cannot be changed once they have been created, we had to use the equals method to fix the "bye" problem. We also had to use the toLowerCase method to fix the "Bye" vs. "bye" problem.

Customer: Big deal. The computer would still answer "That's interesting. Tell me more…" if you insulted it.

TechSupport programmer: That's why used the Random class, so that the computer would have more variety in its responses. For example, it would say "That is explained in the manual. Have you read the manual?" or "That sounds odd. Could you explain the problem in more detail?" This is how a method works to generate random responses: first, it gets the size of the response list by calling its size method. It generates a number between 0 and the size, and then it stores that random number in a local variable called index. It will then print the response that corresponds to the number called index.

Customer: That still doesn't help me fix my problem! It will generate different responses, but it won't give me any help based on my problem.

TechSupport programmer: That's why we are going to use maps. A map is a collection that stores key/value pairs as entries, and by using a HashMap we can look at specific keywords that you enter and generate a response based on that keyword. For example, if your complaint has the word "slow" in it, the computer may tell you to check

your hardware. In order for this to work properly, we also have to use a set, which is a collection that stores each individual element at most once. It's kind of like an array, but the elements don't maintain any specific order.

Customer: When are you going to change the program so that it uses maps?

TechSupport programmer: That will happen soon. I know that even so, our program will not be perfect but will be an improvement.

Customer: How do you supply information to other users who use your classes?

TechSupport programmer: We write documentation for our classes, which are detailed enough for other programmers to use the class without having to read the implementation.

Customer: I have read some java programs before. I see the words "public" and "private" a lot. What do they mean?

TechSupport programmer: Public methods are accessible from inside the class they are in and from other classes, while private elements are accessible only from within the same class. Access modifiers define the visibility of a field, constructor, or method.

Customer: Thank you for your explanation, even though it flew over my head.

TechSupport programmer: You're welcome. It's my pleasure to confuse you. Now I am going to go fix our TechSupport program so that we don't go bankrupt!