Here is how I did this challenge.

1. Study the UML and read the GrandparentsTradition class.

2. Write the ParentsTradition class similar to GrandparentsTradition, specifically:

+ Declare additional attributes

+ Write constructor. Inside the constructor, first call the constructor of the GrandparentsTradition (parent class) with super, and pass the attributes that are in the parent class. Then assign additional attributes

+ Add getters, setters for additional attributes

+ Override toString(): the string would be the string of parent class (accessed with super.toString()) and the additional text in the instruction file

+ Override celebrate(): change grandparents to parents

+ Override tabulateCosts(): the string would be the string of parent class (accessed with super.tabulateCosts()) and the additional text in the instruction file (costs for desserts and drinks)

3. Write the ChildrenTradition class in a similar manner (like Step 2)

4. Write the driver class

+ Complete the createTraditions() method. Do as the comments say. Create an ArrayList containing GrandparentsTradition objects (it can then hold all three types of Tradition objects). Use if and first character in the line to decide which type of tradition the input line is. Read the attributes and call the constructor accordingly

+ Complete the celebrateHolidays() method. Just iterate and print using celebrate() and tabulateCosts() methods

+ Modify the main method to call the above two methods to do the work

5. Test