

Railyard

Instructions: Unless it has been done for you, print a copy of this document for your team. As you work through it, write answers to any questions or prompts that are in **boldface**.

Team:

Manager:

Recorder:

Presenter:

Analyst:

Analyst: Pay special attention to your team's performance in the area of *teamwork*. How well does your team include everyone in discussions to build shared consensus?

Learning Objectives

Content Objectives

Parentheses below correspond to part of the knowledge units in the ACM's *Computer Science Curricula 2013*.

After completing this activity, students should be able to:

- Explain and implement stacks (SDF/Fundamental data structures).
- Explain and implement queues (SDF/Fundamental data structures).
- Explain and use interfaces (PL/Object-oriented programming).

Process Objectives

After completing this activity, students should have improved their ability to:

- Work together to solve problems. [Teamwork]

Model 1: Playing the Game

Open the Railyard project in Eclipse.

This is more of a puzzle than a game. Run Railyard.java and follow the instructions. You may each try to solve it individually or work as a team. The exact rules are deliberately terse; figuring them out is part of the puzzle.

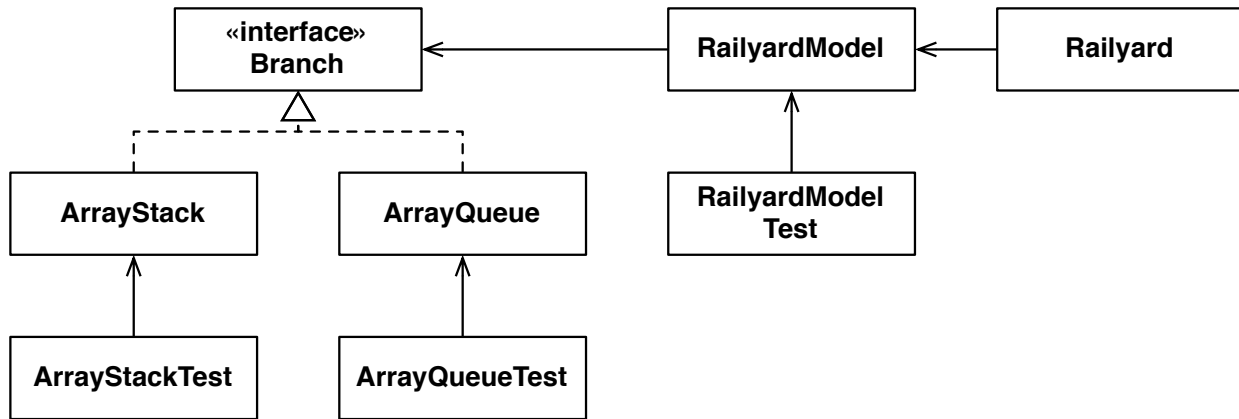
- 1. What two things happen when you click on the > or < symbol?**
- 2. If the symbol is >, what happens when you click on one of the branches on the left side?**
- 3. If the symbol is <, what happens when you click on one of the branches on the left side?**
- 4. What happens when you click on the main line on the right side?**
- 5. How do the top two branches differ from the bottom two?**
- 6. Is there a limit to how many numbered boxes (“rail cars”) can be on one branch?**

Model 2: Branch

Examine Branch.java.

7. According the the first line after the Javadoc comment, this is not a class. What is it?
8. What does Branch contain that a class also contains?
9. What does Branch *not* contain that a class contains?
10. Can you make an instance of Branch? (Try it!)

The relationships between the classes in this projects are shown in the UML class diagram below. The arrows from ArrayStack and ArrayQueue to Branch represent is-a relationships; they are dashed because Branch is an interface.



Model 3: ArrayStack

Examine ArrayStack.java. For this project we'll use "ArrayStack" and "Stack" interchangeably, but in the future you will learn about other kinds of stacks.

11. What special notation is at the top of ArrayStack.java indicating its relationship to the Branch interface?
12. What is the relationship between the methods in ArrayStack and those in Branch?
13. Looking at RailyardModel.java, which Branches are represented by ArrayStacks?

14. In the game, if you put several things into a stack, which one is removed first: the oldest or the newest? In other words, are stacks “first-in, first-out” (FIFO) or “last-in, first out” (LIFO)?
15. A stack is often described as analogous to a stack of plates, with items added and removed at the top. In the branches in the game, is the top drawn facing the left or the right? What about in the main line?
16. If an `ArrayStack` contains several things, is the one at index 0 at the top or bottom?
17. If you add something to an `ArrayStack` whose `size` field is currently 3, where in the `data` array is the new thing placed?
18. If you remove something from an `ArrayStack` whose `size` field is currently 3, from where in the `data` array does the returned item come?
19. What is the order of the running time of all methods in this class?

20. Which methods modify the ArrayStack? Which methods return values? Are there any methods that do both?

Model 4: ArrayQueue

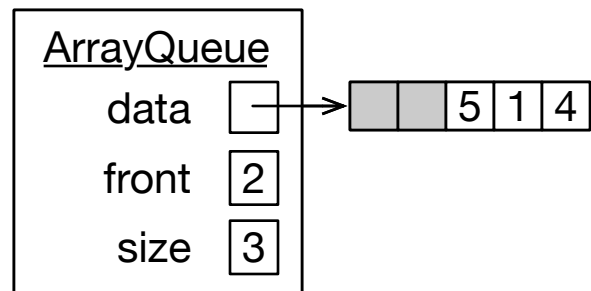
Examine ArrayQueue.java. The word “queue” is pronounced like the letter ‘Q’.

16. Looking at RailyardModel.java, which Branches are represented by ArrayQueues?

17. In the game, if you put several things into a queue, which one is removed first: the oldest or the newest? In other words, are queue “first-in, first-out” (FIFO) or “last-in, first out” (LIFO)?

18. A queue is often described as analogous to a line of people waiting for tickets, with items (people) added at the back and removed from the front. In the branches in the game, is the front drawn facing the left or the right?

19. Suppose an ArrayQueue `q` currently looks like the picture shown at right. What will it look like after `q.add(8)`?



20. What is the order of the running time of all methods in this class?

Reflection

Analyst: Reflect on your team's performance in the area of *teamwork*.

25. What were your team's strengths?

26. What were potential areas of improvement?

27. What insights did you gain?