CS 350 Task 3: Project Proof of Concept Component

This task serves as a proof of concept for the key component (i.e., the five classes collectively) in the larger architecture to come that we presumably identified after Tasks 1 and 2. The specification resides in the provided Javadoc. This document is an overview of the component and its usage.

Overview

This component provides basic creational, structural, and behavioral capabilities on a single simplistic railroad track. A track consists of at least two straight segments joined at both ends.

Creational and Structural Elements

Class `Coordinates` defines a single (*x*,*y*) coordinate pair with arbitrary units.

Class `A_Segment` is an abstract class defining a family of segment types. A segment consists of an arbitrary identifier and coordinates for its two ends called tips *C* and *D*.

Class `SegmentStraight` is the only member of the `A_Segment` family for now. `SegmentCurve` would also be appropriate, but its math is more complex.

Class `Join` connects two segments at either their *C* or *D* tips.

Class `TrackManager` serves two purposes. First, in a creational role, it maintains a collection of independent segments as they are added. Second, in a structural role, it compiles them into a closed track of segments connected by joins. It also validates the structure to ensure the track is correct.

Behavioral Element

Class `A_Segment` contains method `calculatePosition()`. It plays the behavioral role of a notional train car that is located somewhere on the compiled track as a function of distance from a segment tip. Although this solution is entirely static, incrementing the distance iteratively would have the effect of moving the car around the track.

Part I: Questions

Read and attempt to understand this document and the Javadoc with respect to Lecture 22. Submit to the Task 3 Pre Due link in a plain text document any questions you have. There is no format to the document, except separate the questions by a blank line. If you have no questions, submit this statement.

Part II: Implementation

Implement the solution as specified. The line counts from my solution are in square brackets. Submit your five Java source files either individually or zipped.