Documentation – “eco-tram”

Project Description

Team based java program focused on data model and business logic implementations over a fixed timespan, utilizing git source control.

Team name	eco-tram
Project name	eco-tram
Working timespan	2020-11-02 to 2021-02-01
Due date	2021-02-02
Participants	Bünnig, Erik
	Knefelkamp, André
	Martini, Justin
	Voges, Luca

Topic Summary

The data model and general business logic for a tram management system, facilitating proven OOP-Paradigm design patterns where they were deemed appropriate. The main purpose of the data model and its implementation is to provide a framework that can be used to simulate tram-networks as they exist in the real world.

Data model

The data model was chosen to reflect the entities of a regular tram-network, such as the traversable entities that are stations and connections between them, making a core distinction between a connection from A to B and a connection form B to A as these are not always equal in length nor even existent.
Other entities obviously include the trams themselves and the lines they are assigned to. The Tram class was consciously left abstract with the actual implementation being the PassengerTram to leave users the chance to add other tram types such as a MaintenanceTram.

EventManager

The EventManager class was introduced to reduce the coupling between lower rank entities. Instead of tightly holding a reference to another entity directly or multiple for larger networks, the event manager would communicate important events through its provided implementation.
This also leaves entities the ability to opt out of reacting to certain events where other entities would, adding more choice to the end user when extending or modifying the base implementation.

Available Resources

Source control	git, git-cli	
	GitKraken	
	SourceTree	
IDE’s and editors	IntelliJ IDEA (Edu/Ultimate)	
	Visual Studio Code	
SDK	JDK	15
Build tools	Maven	3.6.3
Test frameworks	J-Unit	4.31.1

Milestones

Different milestones that just generally changed or improved the data model after evaluation of the current state of the project.

Reevaluation data model	2020-12-10
The previous data model was more a representation of what is bad planning, as it was implemented with a half-finished class diagram as template.	
Added event manager	2020-12-18
After evaluation of the data model, it was clear that communication between models must be loosely coupled to reduce the overall reference count between entities.	
Finalized data model	2021-01-14
Overhaul of the then interface ITraversable to a base class for traversable entities to ensure exposing inherited members like those of ModelBase where those entities were handled generically.	
Added repository pattern	2021-01-21
Addition of repository pattern as interface between a unknown storage system for entities that exposes simple database-like operations	

Conclusion

The project is not in the final state we would have hoped for it to be, certain features are still missing but where left as the time for evaluation and implementation would not have been enough. Those features include a MockModelFactory for easy creation of more complex entities such as Lines or a general Runner class that would simulate a tram-network, registering listeners as shown in the README.md and advancing trams regularly. But these features were intentionally set as optional, as they don’t advance the core design of the project, which had higher priority.