

A dark blue vertical bar runs down the left side of the page. A blue arrow points to the right from the bar, containing the date.

26-05-2023

Java1 Project_4

Several thin, curved lines in dark blue and light grey originate from the bottom left corner and sweep upwards and to the right.

Heðin Tórstún & Helena Hentze

The task of this project is to construct a train simulation prototype with the given initial data for stations, trains, vehicles, and timetable.

The simulation should consist of the assembly of the trains at the departure stations, running of the trains and their disassembly at the destination stations. The outcome is the information on which trains arrived on time, which trains were delayed and which trains were incomplete.

The following platforms and development tools were used:

Windows 10, arch Linux, vscode, neovim, Emacs and make.

Entry point

The entry point to this program (project) is the Main.java file.

Cart class

The Cart class contains the cart ID and type, these values are final integers meaning that the values cannot be changed.

This class is the parent class of other subclasses such as Coach, SleepingCar, OpenFreight, CoveredFreight, ElectricalEngine and DieselEngine.

The getID method returns the ID value for a cart. The ID's range from 1 to 741. The getType method returns the type of cart. The type's range from 0 to 5, for example type 0 represents the coach car.

Child classes

The other child classes extend the Cart class, these classes inherit the ID and type as super keyword. The super keyword refers to the parent class. It is used to call superclass methods, and to access the superclass value. All the values in the child classes are final values.

Coach

Coach has the type 0. This class contains numberOfChairs and internetAccesses.

The `getNumberOfChairs` method returns the number of chairs in a cart and the `getInternetAccess` returns a value that is either 0 meaning no access or 1 meaning access.

SleepingCar

Sleeping car has the type 1. This class contains `numberOfBeds`. The `getNumberOfBeds` method returns the number of beds that are in a cart.

OpenFreight

Open freight has the type 2. This class contains `cargoCapacity` and `floorArea`. The `getCargoCapacity` method returns the cargo capacity measured in tonnes and the `getFloorArea` method returns the floor area measured in square meters.

CoveredFreight

Covered freight has the type 3. This class contains `cargoCapacity`. The `getCargoCapacity` method returns the cargo capacity measured in cubic meters.

ElectricalEngine

Electrical engine has the type 4. This class contains `maxSpeed` and `power`. The `getMaxSpeed` method returns the max speed measured in km/h and the `getPower` method returns the power in measured kW.

DieselEngine

Diesel engine has the type 5. This class contains `maxSpeed` and `fuelConsumption`. The `getMaxSpeed` method returns the max speed measured in km/h and the `getFuelConsumption` method returns the fuel consumption measured in liters/h.

File reader class

The file reader class reads in the text (.txt) files one line at a time, and then adds the line to an array-list of strings, then it returns the array list when no new lines occur.

For each text file there is an associated handler that calls the file reader class with a path to correct file.

The file reader class also catches errors such as NoSuchElementException, IllegalStateException and IOException.

NoSuchElementException

This exception occurs if the data structure is empty or if its next element is requested after reaching the end of the structure.

IllegalStateException

This exception occurs when a method has been invoked at an illegal or inappropriate time for example once a thread has started it cannot be restarted again. This exception may usually occur when working with lists and queues.

IOException

This exception occurs when there is an issue with the Input and Output operations. Examples of such error can be the following: file or directory not found, incorrect file permissions, disk full or write-protected, network or I/O device failure or invalid user input.

These exceptions are caught, and the system prints out ERROR and the exception type.

Vehicle depo class

This class handles the adding and removing carts from a station. This is done with a priority queue that sorts each type of cart queue by their cart ID.

User input and User screen

The user input class initializes userScreen and simulation.

When the user runs the program the following line will appear “Press enter to move the time forwards, or help for help” when the user presses enter the current time 00:10 is shown when the

user presses enter again current time is 00:20, this will continue until all the trains that are in the ready running or arrived state are finished.

When the user types “Help” the list of commands will be displayed:

To Exit the whole program, type “-1”

To Clear the terminal, type “clear”

To quickly finish the simulation, type “rush”

To get to the train Menu, type “train” and then the user can type in a specific train number.

To see the Top 10 Trains, type “top”

To see the timetable, type “time table”

To pick a station, type “station” this will allow the user to type in a specific station where the output will show the trains at that station and the available vehicles.

To see the help display screen, type “help”

The user screen class initializes and updates the train simulation. The trainsLeft method uses stream to take input from the array and prints the color red for incomplete trains, orange for late trains and green for finished trains.

Simulator

The simulator class has two constructors one that sets the start time at 00:00

And one where a parameter sets the start time.

This class simulates all the trains and their respective stations and time passage. It also controls the state of all trains.