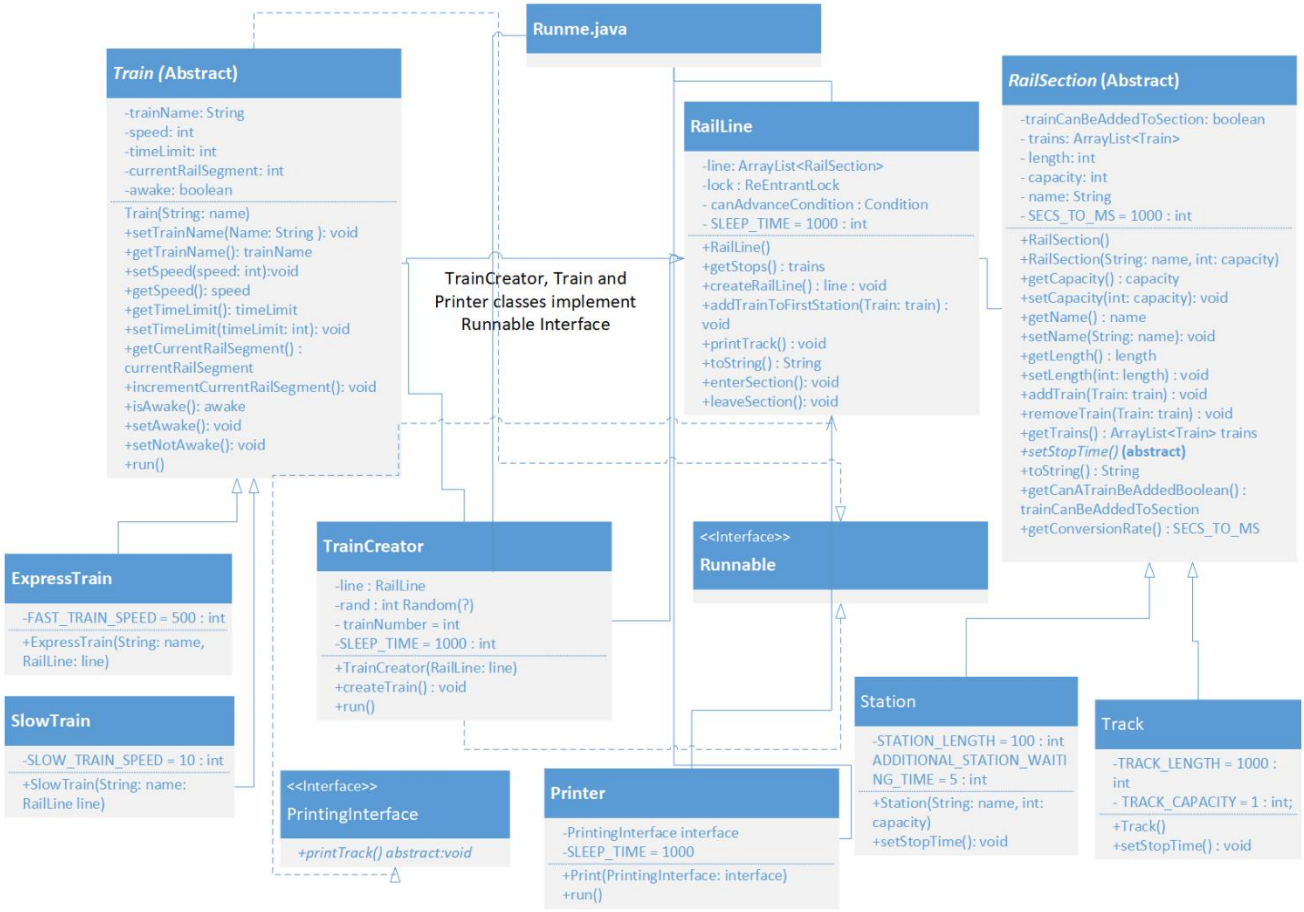
Advanced Programming: Assessed Exercise 1 **05 March 2019**

0808148w

My program consists of ten classes plus an additional "Runme" class that starts the program. It can be illustrated by the following UML Diagram.



The "ExpressTrain" and "SlowTrain" classes inherit from the abstract "Train" class. Express and slow trains differ in speed; therefore, each class has a constant to reflect this.

The train class implements the Runnable interface. Although an instance of this class cannot be instantiated as the class itself is abstract, the subclasses will inherit the run method. This method puts a train thread to sleep for an allotted time on a rail section, calculated using the setStopTime() method in the track/station subclasses. Once it awakens, it'll call enterSection() which determines whether the train can move to the next section. If not possible, it'll go into a blocked state in the enterSection() method. Afterwards, it'll call leaveSection(), which removes the train object from the current track section. A thread object will be created for that train object and it's start method will be called to repeat the cycle.

The "Station" and "Track" classes inherit from the abstract "RailSection" class. The different lengths of each rail section are reflected by different constants in each subclass. A track segment has a constant to reflect the fixed capacity. Meanwhile, a station's capacity is determined by the value that is passed to the constructor. The Rail Section class has a constant, SECS_TO_MS, that both subclasses invoke to convert stop times into milliseconds and get the correct time that a train thread occupies a rail section.

Both "TrainCreator" and "Printer" implement the Runnable interface. The thread object created from the "TrainCreator" class will create trains and place them on the first station indefinitely. If it's not

possible, it'll go into a blocked state until it is signalled (i.e. a flag that turns to true) that a train has vacated the station and finish the process. The thread object from the "Printer" class will keep printing in 1 second intervals.