Overview

My system consists of

- Main - this is used to run the simulator. It accepts only one argument when ran - the simulator's configuration file name. Its only job is to instantiate the simulator class.
- Simulator - this is the main driver of the RIP simulation. It is configured through a configuration file. Its job is to set up the network and control when what is happening. It has the following configurations available:
    - -numOfNodes -> Integer parameter. Specifies how many nodes are there in the network. The indexing of the nodes always starts at 0
    - - maxExchanges -> Integer parameter with default 100. Specifies maximum number of "exchanges" in range [0, inf].
    - -untilStability -> String-boolean flag with default "false". If "true" the simulation will stop once stability is achieved in the network. If "false" the simulation will stop after - maxExchanges number of exchanges.
    - -manual -> String-boolean flag with default "false". If "false" system runs the simulation without any user interference. If "true" the system gives the following commands to the user:
        - "Enter" -> tells the system to simulate the next exchange
        - "split-horizon-on" -> turns split-horizon heuristic on for the next and any subsequent exchanges
        - "split-horizon-off" -> turns split-horizon heuristic off for the next and any subsequent exchanges
    - -splitHorizon -> String flag with default "off". If set to "on" split-horizon heuristic will be on when the simulation starts, otherwise split-horizon heuristic will be off when the simulation starts
    - -infinity -> Integer with default 16. Specifies the infinity cost for the current simulation.
- NetworkNode - representation of a network node. Identified by an ID, has reference to a RoutingTable and known neighbors and implements the RIP protocol.
- RoutingTable - representation of a routing table. It holds RouteTableEntries.
- NetworkLink - representation of a network link between two NetworkNodes with cost associated to the link.
- ScheduledEvent - an abstract class which represents something that should happen at some point. Currently available events are:
    - LinkCostChangeEvent - schedules a link cost change
    - ShowBestRoute - schedules the simulator to output the best route between two nodes
    - TraceRouteTableEvent - schedules the simulator to output the route table of a node