The class, Program3, contains the main method. Therefore, this class controls the entire program.
It consists of:

a default contructor, accessors, mutators and methods inherited from the Object class
instance variables for the threads
instance variables for the start time and simulation time - each in milliseconds

The main method:

Input the length of time to run the simulation (in minutes)
Create an instance of Program3
Call the startSimulation method and pass the time to it
Loop while the Arrival or Departure threads are alive - wait state
Stop the Runway thread from running

The startSimulation method:

parameter - long time - represents the simulation time, convert this to milliseconds

startTime = System.currentTimeMillis();
simulationTime = time * 60000; // time in milliseconds
https://www.google.com/webhp?q=minutes+to+milliseconds#q=minutes+to+milliseconds

start the threads (Arrival, Departure, Runway)
loop for the specified length of time to run the simulation
stop each thread
interrupt each thread

The Runway class:

Instance variables:

-arrival : Arrival	A reference to the arrival thread
-departure : Departure	A reference to the departure thread
-running : boolean	Used by the main program to stop this thread

Constructor:

Runway(Arrival, Departure)	constructor - needs arrival & departure threads

Other Methods:

stopRunning() : void	Change the running state to false
toString() : String	A string representation of the Runway
run() : void	Started by the start method inherited from the thread class loop until main program calls the *stopRunning* method while there are planes in the arrival queue get a plane from the arrival queue display information about airline simulate landing - sleep for secs (arrival time - 5 secs) while there are planes in the departure queue get a plane from the departure queue display information about airline simulate takeoff - sleep for secs (departure time - 4 secs)

The Departure class:

Instance variables:

-queue : Queue<Airline>	Departure queue
-time : int	Takeoff duration (sleep time) 4000 = 4 secs
-running : boolean	Used by the main program to stop this thread

Constructor:

Departure(int)	constructor - needs the time to simulate a takeoff (4 secs)

Accessors:

getQueue() : Queue<Airline>	Accessor for the arrival queue
getTime() : int	Accessor for the takeoff duration time (sleep time)

Other methods:

stopRunning() : void	Change the running state to false
toString() : String	A string representation of the Departure
run() : void	Started by the start method inherited from the thread class loop until main program calls the *stopRunning* method calculate the time (based on time-till-next formula) sleep for random seconds (based on time-till-next formula) place a new airline into the departure queue display information about airline

The Arrival class:

Instance variables:

-queue : Queue<Airline>	Arrival queue
-time : int	Arrival duration (sleep time) 5000 = 5 secs
-running : boolean	Used by the main program to stop this thread

Constructor:

Arrival(int)	constructor - needs the time to simulate an arrival (5 secs)

Accessors:

getQueue() : Queue<Airline>	Accessor for the arrival queue
getTime() : int	Accessor for the landing duration time (sleep time)

Other methods:

stopRunning() : void	Change the running state to false
toString() : String	A string representation of the Arrival
run() : void	Started by the start method inherited from the thread class loop until main program calls the *stopRunning* method calculate the time (based on time-till-next formula) sleep for random seconds (based on time-till-next formula) place a new airline into the arrival queue display information about airline

```java
/**
 * Class with static methods - use these constants and call these methods as necessary
 * Simulation.TAKEOFF_TIME and Simulation.LANDING_TIME can be used in Program3
 * Simulation.timeTillNext method can be called from the Arrival and Departure run methods
 */
import java.util.Random;

public class Simulation {
	public static final int TAKEOFF_TIME = 4000;
	public static final int LANDING_TIME = 5000;
	public static String[] AIRLINES = {"AA","AI","AF","AZ","KL","BA","BW","DL","FL",
		"BA","AC","ET","GH","LH","JM","KE","TW","UA"};
	/*
	 * Converts milliseconds to minutes
	 * @param millisecs
	 * @return timeInMinutes
	 */
	public static long timeInMinutes(long millisecs){
		return millisecs / 60000;
	}
	/*
	 * Converts minutes to milliseconds
	 * @param timeInMinutes
	 * @return timeInMilisecs
	 */
	public static long timeInMilisecs(long timeInMinutes){
		return timeInMinutes * 60000;
	}
	/*
	 * Calculates time till next event
	 * @param meanEventTime in milliseconds
	 * @return timeTillNext - time before next event occurs
	 */
	public static int timeTillNext(int meanEventTime){
		Random random = new Random();
		double randomDouble = random.nextDouble();
		int timeTillNext = (int)Math.round (-meanEventTime * Math.log (1 - randomDouble));

		return timeTillNext;
	}
}
```