HW5a

**Class Airplane description**

The Airplane class keep track of key data like takeoff time, land time and arrival time,

The class can also calculate time in the landing queue and time in the takeoff queue.

**Airplane fields**

**Arrival Time** – the time when a plane is added to a queue.

**Takeoff Time**- the time when the plane is put on the runway.

**Land Time**- the time when a plane is added to the runway.

**Time In Takeoff Queue**- total time spent in takeoff queue.

**Time In Landing Queue** – total time in the landing queue.

**Airplane Methods**

**Airplane()**

**Precondition:** simulation has stated

**Postcondition:** Airplane object is created.

**Description:** creates Airplane, sets airplane arrival time to current time

**SetTakeofTime()**

**Precondition:** Airplane is in takeoff Queue

**Postcondition**: Takeoff Time is set to current time, time in takeoff Queue is set.

**Description:** sets the takeoff time and then uses the difference of arrival time and takeoff time to calculate time in takeoff Queue and sets it.

**setLandTime()**

**Precondition:** Airplane is in Landing Queue.

**Postcondition**: Land Time is set to current time, time in Landing Queue is set.

**Description:** sets the Land Time and then uses the difference of arrival time and Land Time to calculate time in Landing Queue and sets it.

**getTimeInLandingQueue()**

**Precondition:** Airplane has landed or crashed

**Postcondition**: Time in Landing Queue is returned.

**Description:** gets the amount of time the plane spent in the landing queue

**getTimeInTakeoffQueue()**

**Precondition:** Airplane has taken off

**Postcondition**: Time in Takeoff Queue is returned.

**Description:** gets the amount of time the plane spent in the takeoff queue

**getArrivalTime()**

**Precondition:** Airplane has arrived

**Postcondition**: Arrival time is returned.

**Description:** gets the Arrival time.

**Class Runway description**

The runway class manages the airplanes, takeoff, and landing queue, it also adds planes to a corresponding array list depending on if they have crashed, landed, or took off.

**Runway Fields**

**landingQueue:** holds Airplanes waiting to land.

**takeoffQueue:** holds Airplanes waiting to takeoff.

**arrayCrashed:** holds Airplanes after they crash.

**Arraylanded:** holds airplanes after they land.

**arraytookoff :** holds Airplanes after they takeoff**.**

**inAirTimeArray:** holds both crashes and landed Airplanes.

**maxLandingQueueTime:** the time an Airplane can be in the air before it crashes

**landingTime:** the time it takes an Airplane to land

**TakeoffTime:** the time it takes an Airplane to takeoff

**isClear:** Boolean true is runway is clear.

**clearTimeAt:** the time when the runway will be clear.

**Runway methods**

**Runway(int maxLandingQueueTime, int landingTime, int TakeoffTime)**

**Precondition:** maxLandingQueueTime, landingTime, TakeoffTime have been

Entered by user in airport class.

**Postcondition:** A runway object is created and maxLandingQueueTime, landingTime, TakeoffTime fields are set.

**Description:** creates the Runway object.

**addLandingQueue(Airplane plane)**

**Precondition:** Runway is created, Airplane is created

**Postcondition:** adds Airplane to LandingQueue.

**Description:** adds Airplane to LandingQueue.

**addTakeoffQueue( Airplane plane)**

**Precondition:** Runway is created, Airplane is created

**Postcondition:** adds Airplane to TakeoffQueue.

**Description:** adds Airplane to TakeoffQueue.

**isRunwayClear()**

**Precondition:** Runway is created.

**Postcondition:** none

**Description:** returns a Boolean value depending of the runway is clean or not

**simulateRunway()**

**Precondition:** Runway is created.

**Postcondition:** moved planes off and on to runway, evaluate what queue has highest landing priority, after planes are moved off runway puts plane in corresponding array list used to present data to user in airport class.

**Description:** simulates airplane queues, runway usage, and adds planes to the correct array after they are off the runway.

**getCrashedArray()**

**Precondition:** Runway is created.

**Postcondition:** returns CrashedArray

**Description:** returns CrashedArray array of crashed planes

**getLandedArray()**

**Precondition:** Runway is created.

**Postcondition:** returns LandedArray

**Description:** returns LandedArray array of landed planes

**getTookoff()**

**Precondition:** Runway is created.

**Postcondition:** returns arraytookoff

**Description:** returns arraytookoff array of planes that took off

**getInAirTimeArray()**

**Precondition:** Runway is created.

**Postcondition:** returns inAirTimeArray

**Description:** returns inAirTimeArray array of planes that were waiting to land at any point.

**Airport description**

This class gets the user inputs, simulates all classes, displays output, and has a global time variable for all classes.

**Airport Fields**

**currentTime:** current time of simulation

**timeNeededLanding:** The time needed for a plane to land.

**timeNeededTakeoff:** The time needed for a plane to takeoff.

**probOfTakeoff:** probability of a plane taking off.

**probOfLanding:** probability of a plane landing.

**maxTimeLandingQueue:** max time a plane can spend in the landing queue before it crashes.

**totalSimulationTime:** the amount of time total time the simulation will run.

**Runway:** the runway object

**Airport methods**

**getCurrentTime()**

**precondition:** simulation has started.

**postcondition:** returns the current time.

**Description**: gets the current time in minutes of the simulation.

**getInputs()**

**precondition:** None

**postcondition:** sets variables for timesNeededLanding, TimeNeededTakeoff, probOfTakeoff, ProbOfLanding, MaxTimeLandingQueue, TotoalSimulationTime.

**Description**: prompts the user for all inputs necessary to run the simulation.

**simulateAll()**

**precondition:** getInput has already been called.

**postcondition:** current time has been increased by one minute, runway is simulated, planes are added to Queues, Planes are added to arrays after they have crashed, landed, or tookoff.

**Description**: simulates one minute across all classes.

**Main()**

**precondition:** none

**postcondition:** none

**Description**: creates objects of airport, runway, and Planes. The main method contains the loop for the simulation, after the simulation is done the results will be displayed to the user.

|  |
| --- |
| **Airport** |
| **+currentTime: int**  **- timeNeededLanding: int**  **- timeNeededTakeoff: int**  **- probOfTakeoff: double**  **- probOfLanding: double**  **- maxTimeLandingQueue: int**  **- totalSimulationTime: int**  **- runway: Runway** |
| **+ main(String[]): void**  **+ simulateAll(): void**  **+ getInputs(): void**  **+ getCurrentTime(): int** |

|  |
| --- |
| Runway |
| - landingQueue: Queue<Airplane>  - takeoffQueue: Queue<Airplane>  - arrayCrashed: ArrayList<Airplane>  - arraylanded: ArrayList<Airplane>  - arraytookoff: ArrayList<Airplane>  - inAirTimeArray: ArrayList<Airplane>  - maxLandingQueueTime: int  - landingTime: int  - TakeoffTime: int  - isClear: Boolean  - clearTimeAt: int |
| + Runway(int, int, int)  + addLandingQueue(Airplane): void  + addTakeoffQueue(Airplane): void  + isRunwayClear(): Boolean  + simulateRunway(): void  + getCrashedArray(): ArrayList  + getLandedArray(): ArrayList  + getTookoff(): ArrayList  + getInAirTimeArray() : ArrayList |

|  |
| --- |
| **Airplane** |
| **- arrivalTime: int**  **- takeoffTime: int**  **- landTime: int**  **- timeInLandingQueue: int**  **- timeInTakeoffQueue: int** |
| **+ Airplane()**  **+ setTakeOffTime(): void**  **+ setLandTime(): void**  **+ getTimeInLandingQueue(): int**  **+ getTimeInTakeoffQueue(): int**  **+ getArrivalTime(): int** |