|  |  |  |
| --- | --- | --- |
| **KSU – CCIS – CS** | **CSC 456** | **1431-32 MT2 Exam** |

**Question 1:**

1. Explain how the Berkeley algorithm works.
2. Write a pseudo code for this algorithm.
3. What are the basic rules of the synchronization based on the logical clock.
4. Define a pseudo code that preserves these rules when processes exchange messages.

**Question 2**

Let’s consider the following java class:

public class Flight {

private String name;

private double departure;

private double arrival;

public Flight(String s, double dt, double at) { …}

// all available methods are the getters and setters.

}

public class FlightManager {

private Hashtable<String, Flight> flights;

// the method **add** stores f in the flight manager

public void **add**(Flight f) { … }

// the method **get** returns the Flight having s as name

public Flight **get**(String s) { … }

We need that the objects of the class FlightManager work as servers.

Implement the following using RMI:

1. The class Flight.
2. The class FlightManager.
3. The main program that launches a flight manager.
4. The client main application:
   1. Lists the set of active flight managers.
   2. Selects one of the them.
   3. Adds the Flight (“Riyadh-Jeddah”, 14.50 , 18.20)
   4. Displays the departure and arrival times of the flight “Riyadh-Dubai” .

|  |  |  |
| --- | --- | --- |
| **KSU – CCIS – CS** | **CSC 456** | **1431-32 MT2 Exam** |

**Question 1:**

1. Explain how the Berkeley algorithm works.
2. Write a pseudo code for this algorithm.
3. What are the basic rules of the synchronization based on the logical clock.
4. Define a pseudo code that preserves these rules when processes exchange messages.

**Question 2**

Let’s consider the following java class:

public class Flight {

private String name;

private double departure;

private double arrival;

public Flight(String s, double dt, double at) { …}

// all available methods are the getters and setters.

}

public class FlightManager {

private Hashtable<String, Flight> flights;

// the method **add** stores f in the flight manager

public void **add**(Flight f) { … }

// the method **get** returns the Flight having s as name

public Flight **get**(String s) { … }

We need that the objects of the class FlightManager work as servers.

Implement the following using RMI:

1. The class Flight.
2. The class FlightManager.
3. The main program that launches a flight manager.
4. The client main application:
   1. Lists the set of active flight managers.
   2. Selects one of the them.
   3. Adds the Flight (“Riyadh-Jeddah”, 14.50 , 18.20)
   4. Displays the departure and arrival times of the flight “Riyadh-Dubai” .

|  |  |  |
| --- | --- | --- |
| **KSU – CCIS – CS** | **CSC 456** | **1431-32 MT2 Exam** |

**Question 1:**

1. Explain how the Berkeley algorithm works.
2. Write a pseudo code for this algorithm.
3. What are the basic rules of the synchronization based on the logical clock.
4. Define a pseudo code that preserves these rules when processes exchange messages.

**Question 2**

Let’s consider the following java class:

public class Flight {

private String name;

private double departure;

private double arrival;

public Flight(String s, double dt, double at) { …}

// all available methods are the getters and setters.

}

public class FlightManager {

private Hashtable<String, Flight> flights;

// the method **add** stores f in the flight manager

public void **add**(Flight f) { … }

// the method **get** returns the Flight having s as name

public Flight **get**(String s) { … }

We need that the objects of the class FlightManager work as servers.

Implement the following using RMI:

1. The class Flight.
2. The class FlightManager.
3. The main program that launches a flight manager.
4. The client main application:
   1. Lists the set of active flight managers.
   2. Selects one of the them.
   3. Adds the Flight (“Riyadh-Jeddah”, 14.50 , 18.20.
   4. Displays the departure and arrival times of the flight “Riyadh-Dubai”

|  |  |  |
| --- | --- | --- |
| **KSU – CCIS – CS** | **CSC 456** | **1431-32 MT2 Exam** |

**Question 1:**

1. Explain how the Berkeley algorithm works.
2. Write a pseudo code for this algorithm.
3. What are the basic rules of the synchronization based on the logical clock.
4. Define a pseudo code that preserves these rules when processes exchange messages.

**Question 2**

Let’s consider the following java class:

public class Flight {

private String name;

private double departure;

private double arrival;

public Flight(String s, double dt, double at) { …}

// all available methods are the getters and setters.

}

public class FlightManager {

private Hashtable<String, Flight> flights;

// the method **add** stores f in the flight manager

public void **add**(Flight f) { … }

// the method **get** returns the Flight having s as name

public Flight **get**(String s) { … }

We need that the objects of the class FlightManager work as servers.

Implement the following using RMI:

1. The class Flight.
2. The class FlightManager.
3. The main program that launches a flight manager.
4. The client main application:
   1. Lists the set of active flight managers.
   2. Selects one of the them.
   3. Adds the Flight (“Riyadh-Jeddah”, 14.50 , 18.20)

Displays the departure and arrival times of the flight “Riyadh-Dubai”.