IT-145 Java Module 7 Final Project

Cruise Ship Assignment

Find all 4 files completed below.

Driver.java, Ship.java, Cruise.java, Passenger.java

Diver.java

**import** java.util.ArrayList;  
**import** java.util.Scanner;  
**import static** java.lang.Integer.*parseInt*;  
**public class** Driver {  
 *// instance variables (add more as needed)* **private static** ArrayList<Ship> *shipList* = **new** ArrayList();  
 **private static** ArrayList<Cruise> *cruiseList* = **new** ArrayList();  
 **private static** ArrayList<Passenger> *passengerList* = **new** ArrayList();  
 **public static void** main(String[] args) {  
 *initializeShipList*(); *// initial ships  
 initializeCruiseList*(); *// initial cruises  
 initializePassengerList*(); *// initial passengers  
 // add loop and code here that accepts and validates user input  
 // and takes the appropriate action. include appropriate  
 // user feedback and redisplay the menu as needed* Scanner in = **new** Scanner(System.***in***);  
 String choice=**""**;  
 **while**(choice!=**"x"**){  
 *// call menu method  
 displayMenu*();  
 *// get choice* choice = in.next();  
 *// switch statement to take appropriate action* **switch**(choice){  
 **case "1"**: *addShip*(); **break**;  
 **case "2"**: *editShip*(); **break**;  
 **case "3"**: *addCruise*(); **break**;  
 **case "4"**: *editCruise*(); **break**;  
 **case "5"**: *addPassenger*(); **break**;  
 **case "6"**: *editPassenger*(); **break**;  
 **case "A"**: *printShipList*(**"name"**); **break**;  
 **case "B"**: *printShipList*(**"active"**); **break**;  
 **case "C"**: *printShipList*(**"full"**); **break**;  
 **case "D"**: *printCruiseList*(**"list"**); **break**;  
 **case "E"**: *printCruiseList*(**"details"**); **break**;  
 **case "F"**: *printPassengerList*(); **break**;  
 **case "x"**: System.*exit*(0);  
 *// for invalid selection* **default**: System.***out***.println(**"Invalid Menu Selection. Select Again..\n"**); **break**;  
 }  
 }  
  
 }  
 *// hardcoded ship data for testing  
 // Initialize ship list* **public static void** initializeShipList() {  
 *add*(**"Candy Cane"**, 20, 40, 10, 60, **true**);  
 *add*(**"Peppermint Stick"**, 10, 20, 5, 40, **true**);  
 *add*(**"Bon Bon"**, 12, 18, 2, 24, **false**);  
 *add*(**"Candy Corn"**, 12, 18, 2, 24, **false**);  
 }  
 *// hardcoded cruise data for testing  
 // Initialize cruise list* **public static void** initializeCruiseList() {  
 Cruise newCruise = **new** Cruise(**"Southern Swirl"**, **"Candy Cane"**, **"Miami"**, **"Cuba"**, **"Miami"**);  
 *cruiseList*.add(newCruise);  
 }  
 *// hardcoded cruise data for testing  
 // Initialize passenger list* **public static void** initializePassengerList() {  
 Passenger newPassenger1 = **new** Passenger(**"Neo Anderson"**, **"Southern Swirl"**, **"STE"**);  
 *passengerList*.add(newPassenger1);  
 Passenger newPassenger2 = **new** Passenger(**"Trinity"**, **"Southern Swirl"**, **"STE"**);  
 *passengerList*.add(newPassenger2);  
 Passenger newPassenger3 = **new** Passenger(**"Morpheus"**, **"Southern Swirl"**, **"BAL"**);  
 *passengerList*.add(newPassenger3);  
 }  
 *// custom method to add ships to the shipList ArrayList* **public static void** add(String tName, **int** tBalcony, **int** tOceanView,  
 **int** tSuite, **int** tInterior, **boolean** tInService) {  
 Ship newShip = **new** Ship(tName, tBalcony, tOceanView, tSuite, tInterior, tInService);  
 *shipList*.add(newShip);  
 }  
 **public static void** printShipList(String listType) {  
 *// printShipList() method prints list of ships from the  
 // shipList ArrayList. There are three different outputs  
 // based on the listType String parameter:  
 // name - prints a list of ship names only  
 // active - prints a list of ship names that are "in service"  
 // full - prints tabbed data on all ships* **if** (*shipList*.size() < 1) {  
 System.***out***.println(**"\nThere are no ships to print."**);  
 **return**;  
 }  
 **if** (listType == **"name"**) {  
 System.***out***.println(**"\n\nSHIP LIST - Name"**);  
 **for** (**int** i = 0; i < *shipList*.size(); i++) {  
 System.***out***.println(*shipList*.get(i));  
 }  
 } **else if** (listType == **"active"**) {  
 System.***out***.println(**"\n\nSHIP LIST - Active"**);  
 *// work on code block* **for** (**int** i = 0; i < *shipList*.size(); i++) {  
 *// print ships list that are active* **if**(*shipList*.get(i).getInService())  
 System.***out***.println(*shipList*.get(i));  
 }  
 } **else if** (listType == **"full"**) {  
 System.***out***.println(**"\n\nSHIP LIST - Full"**);  
 System.***out***.println(**"-----------------------------------------------"**);  
 System.***out***.println(**" Number of Rooms In"**);  
 System.***out***.print(**"SHIP NAME Bal OV Ste Int Service"**);  
 System.***out***.println(**"\n-----------------------------------------------"**);  
 **for** (Ship eachShip: *shipList*)  
 eachShip.printShipData();  
 } **else** System.***out***.println(**"\n\nError: List type not defined."**);  
 }  
 **public static void** printCruiseList(String listType) {  
 **if** (*cruiseList*.size() < 1) {  
 System.***out***.println(**"\nThere are no cruises to print."**);  
 **return**;  
 }  
 **if** (listType == **"list"**) {  
 System.***out***.println(**"\n\nCRUISE LIST"**);  
 **for** (**int** i=0; i < *cruiseList*.size(); i++) {  
 System.***out***.println(*cruiseList*.get(i));  
 }  
 } **else if** (listType == **"details"**) {  
 System.***out***.println(**"\n\nCRUISE LIST - Details"**);  
 System.***out***.println(**"------------------------------------------------------------------------------------------"**);  
 System.***out***.println(**" |----------------------PORTS-----------------------|"**);  
 System.***out***.print(**"CRUISE NAME SHIP NAME DEPARTURE DESTINATION RETURN"**);  
 System.***out***.println(**"\n-----------------------------------------------------------------------------------------"**);  
 **for** (Cruise eachCruise: *cruiseList*)  
 eachCruise.printCruiseDetails();  
 } **else** System.***out***.println(**"\n\nError: List type not defined."**);  
 }  
 **public static void** printPassengerList() {  
 **if** (*passengerList*.size() < 1) {  
 System.***out***.println(**"\nThere are no passengers to print."**);  
 **return**;  
 }  
 System.***out***.println(**"\n\nPASSENGER LIST"**);  
 System.***out***.println(**"-----------------------------------------------------"**);  
 System.***out***.print(**"PASSENGER NAME CRUISE ROOM TYPE"**);  
 System.***out***.println(**"\n-----------------------------------------------------"**);  
 **for** (Passenger eachPassenger: *passengerList*)  
 eachPassenger.printPassenger();  
 }  
 *// display text-based menu* **public static void** displayMenu() {  
 System.***out***.println(**"\n\n"**);  
 System.***out***.println(**"\t\t\tLuxury Ocean Cruise Outings"**);  
 System.***out***.println(**"\t\t\t\t\tSystem Menu\n"**);  
 System.***out***.println(**"[1] Add Ship [A] Print Ship Names"**);  
 System.***out***.println(**"[2] Edit Ship [B] Print Ship In Service List"**);  
 System.***out***.println(**"[3] Add Cruise [C] Print Ship Full List"**);  
 System.***out***.println(**"[4] Edit Cruise [D] Print Cruise List"**);  
 System.***out***.println(**"[5] Add Passenger [E] Print Cruise Details"**);  
 System.***out***.println(**"[6] Edit Passenger [F] Print Passenger List"**);  
 System.***out***.println(**"[x] Exit System"**);  
 System.***out***.println(**"\nEnter a menu selection: "**);  
 }  
 *// Add a New Ship* **public static void** addShip() {  
 *// working on this method  
 // prompt for ship's name* Scanner newshipInput = **new** Scanner(System.***in***);  
 System.***out***.println(**"\n\nEnter the new Ship's Name:"**);  
 String newshipName = newshipInput.nextLine();  
 *// input validation* **for** (Ship eachShip : *shipList*) {  
 **if** (eachShip.getShipName().equalsIgnoreCase(newshipName)) {  
 System.***out***.println(**"That Ship is already in the system. Exiting to menu..."**);  
 **return**; *// quits addShip() method processing* }  
 }  
 *// enter other details* System.***out***.println(**"Enter the number of passenger cabins"**);  
 System.***out***.print(**"Balcony: "**);  
 String balconyString = newshipInput.nextLine();  
 **int** balcony = 0;  
 *// input validation* **if** (*isANumber*(balconyString))  
 balcony = Integer.*parseInt*(balconyString);  
 System.***out***.print(**"Ocean View:"**);  
 String oceanViewString = newshipInput.nextLine();  
 **int** oceanView = 0;  
 **if** (*isANumber*(oceanViewString))  
 oceanView = Integer.*parseInt*(oceanViewString);  
 System.***out***.print(**"Suite: "**);  
 String roomSuiteString = newshipInput.nextLine();  
 **int** roomSuite = 0;  
 **if** (*isANumber*(roomSuiteString))  
 roomSuite = Integer.*parseInt*(roomSuiteString);  
 System.***out***.print(**"Interior: "**);  
 String roomInteriorString = newshipInput.nextLine();  
 **int** roomInterior = 0;  
 **if** (*isANumber*(roomInteriorString))  
 roomInterior = Integer.*parseInt*(roomInteriorString);  
  
 System.***out***.print(**"Is the ship in Service(y/n):"**);  
 String inService = newshipInput.nextLine();  
 **boolean** ShipInService;  
 *// assign inservice status* **if**(inService==**"y"**)  
 ShipInService = **true**;  
 **else** ShipInService = **false**;  
 *// add the entered details to the list  
 shipList*.add(**new** Ship(newshipName, balcony, oceanView, roomSuite, roomInterior, ShipInService));  
 }  
 *// Edit an existing ship* **public static void** editShip() {  
 *// This method does not need to be completed* System.***out***.println(**"\nThe \"Edit Ship\" feature is not yet implemented."**);  
 }  
 *// Add a New Cruise* **public static void** addCruise() {  
 *// working on this method  
 // prompt for details* Scanner newCruiseInput = **new** Scanner(System.***in***);  
 System.***out***.print(**"\n\nEnter the new Cruise's Name: "**);  
 String newCruiseName = newCruiseInput.nextLine();  
 *// input validation* **for** (Cruise eachCruise : *cruiseList*) {  
 **if** (eachCruise.getCruiseName().equalsIgnoreCase(newCruiseName)) {  
 System.***out***.println(**"That Cruise is already in the system. Exiting to menu..."**);  
 **return**;*// quits addCruise() method processing* }  
 }  
 *// enter other details* System.***out***.print(**"Enter Cruise Ship's Name: "**);  
 String cruiseShip = newCruiseInput.nextLine();  
 System.***out***.print(**"Enter the cruise's Departure Port: "**);  
 String departurePort = newCruiseInput.nextLine();  
 System.***out***.print(**"Enter the cruise's Destination: "**);  
 String destination = newCruiseInput.nextLine();  
 System.***out***.print(**"Enter the cruise's Return Port: "**);  
 String returnPort = newCruiseInput.nextLine();  
 *// add to the list  
 cruiseList*.add(**new** Cruise(newCruiseName, cruiseShip, departurePort, destination, returnPort));  
  
 }  
 *// Edit an existing cruise* **public static void** editCruise() {  
 *// This method does not need to be completed* System.***out***.println(**"\nThe \"Edit Cruise\" feature is not yet implemented."**);  
 }  
 *// Add a New Passenger* **public static void** addPassenger() {  
 *// add a flag* **int** flag = 0;  
 Scanner newPassengerInput = **new** Scanner(System.***in***);  
 System.***out***.println(**"\n\nEnter the new passenger's name: "**);  
 String newPassengerName = newPassengerInput.nextLine();  
 *// ensure new passenger name does not already exist* **for** (Passenger eachPassenger: *passengerList*) {  
 **if** (eachPassenger.getPassengerName().equalsIgnoreCase(newPassengerName)) {  
 System.***out***.println(**"That passenger is already in the system. Exiting to menu..."**);  
 **return**; *// quits addPassenger() method processing* }  
 }  
 *// get cruise name for passenger* System.***out***.println(**"Enter cruise name: "**);  
 String newCruiseName = newPassengerInput.nextLine();  
 *// ensure cruise exists* **for** (Cruise eachCruise: *cruiseList*) {  
 **if** (eachCruise.getCruiseName().equalsIgnoreCase(newCruiseName)) {  
 *// cruise does exist  
 // set flag* flag = 1;  
 }  
 }  
 *// if flag is not set return* **if** (flag==0) {  
 System.***out***.println(**"That cruise does not exist in the system. Exiting to menu..."**);  
 **return**; *// quits addPassenger() method processing* }  
 *// get room type* System.***out***.println(**"Enter Room Type (BAL, OV, STE, or INT): "**);  
 String room = newPassengerInput.nextLine();  
 *// validate room type* **if** ((room.equalsIgnoreCase(**"BAL"**)) || (room.equalsIgnoreCase(**"OV"**)) ||  
 (room.equalsIgnoreCase(**"STE"**)) || (room.equalsIgnoreCase(**"INT"**))) {  
 *// validation passed - add passenger* Passenger newPassenger = **new** Passenger(newPassengerName, newCruiseName, room.toUpperCase());  
 *passengerList*.add(newPassenger);  
 } **else** {  
 System.***out***.println(**"Invalid input. Exiting to menu..."**);  
 **return**; *// quits addPassenger() method processing* }  
 }  
 *// Edit an existing passenger* **public static void** editPassenger() {  
 *// This method does not need to be completed* System.***out***.println(**"The \"Edit Passenger\" feature is not yet implemented."**);  
 }  
 *// Method to check if input is a number* **public static boolean** isANumber(String str) {  
 **for** (**int** i = 0; i < str.length(); i++) {  
 **if** (Character.*isDigit*(str.charAt(i)) == **false**)  
 **return false**;  
 }  
 **return true**;  
 }  
}

Ship.java

**public class** Ship {  
 *// Instance Variables* **private** String **shipName**;  
 **private int roomBalcony**;  
 **private int roomOceanView**;  
 **private int roomSuite**;  
 **private int roomInterior**;  
 **private boolean inService**;  
 *// Constructor - default* Ship() {  
 }  
 *// Constructor - full* Ship(String tName, **int** tBalcony, **int** tOceanView,  
 **int** tSuite, **int** tInterior, **boolean** tInService) {  
 **shipName** = tName;  
 **roomBalcony** = tBalcony;  
 **roomOceanView** = tOceanView;  
 **roomSuite** = tSuite;  
 **roomInterior** = tInterior;  
 **inService** = tInService;  
 }  
 *// Accessors* **public** String getShipName() {  
 **return shipName**;  
 }  
 **public int** getRoomBalcony() {  
 **return roomBalcony**;  
 }  
 **public int** getRoomOceanView() {  
 **return roomOceanView**;  
 }  
 **public int** getRoomSuite() {  
 **return roomSuite**;  
 }  
 **public int** getRoomInterior() {  
 **return roomInterior**;  
 }  
 **public boolean** getInService() {  
 **return inService**;  
 }  
 *// Mutators* **public void** setShipName(String tVar) {  
 **shipName** = tVar;  
 }  
 **public void** setRoomBalcony(**int** tVar) {  
 **roomBalcony** = tVar;  
 }  
 **public void** setRoomOceanView(**int** tVar) {  
 **roomOceanView** = tVar;  
 }  
 **public void** setRoomSuite(**int** tVar) {  
 **roomSuite** = tVar;  
 }  
 **public void** setRoomInterior(**int** tVar) {  
 **roomInterior** = tVar;  
 }  
 **public void** setInService(**boolean** tVar) {  
 **inService** = tVar;  
 }  
 *// print method* **public void** printShipData() {  
 *// made some changes to it* **int** spaceCount;  
 String spaces = **""**;  
 spaceCount = 20 - **shipName**.length();  
 **for** (**int** i = 1; i <= spaceCount; i++) {  
 spaces = spaces + **" "**;  
 }  
 System.***out***.print(**shipName** + spaces + **roomBalcony** + **"\t"** + **roomOceanView**);  
 System.***out***.println(**" "**+ **roomSuite** + **" "** + **roomInterior** + **" "** + **inService**);  
 }  
 *// method added to print ship's name vice memory address* @Override  
 **public** String toString() {  
 **return shipName**;  
 }  
}  
  
  
 Class Cruise:  
  
  
  
*//Cruise.java:***public class** Cruise {  
 *// Class Variables* **private** String **cruiseName**;  
 **private** String **cruiseShipName**;  
 **private** String **departurePort**;  
 **private** String **destination**;  
 **private** String **returnPort**;  
 *// Constructor - default* Cruise() {}  
 *// Constructor - full* Cruise(String tCruiseName, String tShipName, String tDeparture, String tDestination, String tReturn) {  
 **cruiseName** = tCruiseName;  
 **cruiseShipName** = tShipName;  
 **departurePort** = tDeparture;  
 **destination** = tDestination;  
 **returnPort** = tReturn;  
 }  
 *// Accessors* **public** String getCruiseName() {  
 **return cruiseName**;  
 }  
 **public** String getCruiseShipName() {  
 **return cruiseShipName**;  
 }  
 **public** String getDeparturePort() {  
 **return departurePort**;  
 }  
 **public** String getDestination() {  
 **return destination**;  
 }  
 **public** String getReturnPort() {  
 **return returnPort**;  
 }  
 *// Mutators* **public void** setCruiseName(String tVar) {  
 **cruiseName** = tVar;  
 }  
 **public void** setCruiseShipName(String tVar) {  
 **cruiseShipName** = tVar;  
 }  
 **public void** setDeparturePort(String tVar) {  
 **departurePort** = tVar;  
 }  
 **public void** setDestination(String tVar) {  
 **destination** = tVar;  
 }  
 **public void** setReturnPort(String tVar) {  
 **returnPort** = tVar;  
 }  
 *// print cruise details* **public void** printCruiseDetails() {  
 *// working on this method  
 // made some changes to it* String spaces = **" "**;  
 System.***out***.print(**cruiseName**+spaces);  
 System.***out***.print(spaces+**cruiseShipName**+spaces);  
 System.***out***.print(**"\t"**+**departurePort**+**"\t"**);  
 System.***out***.print(**"\t"**+**destination**+**"\t"**);  
 System.***out***.print(**"\t"**+**returnPort**+**"\n"**);  
 }  
 *// method added to print ship's name vice memory address* @Override  
 **public** String toString() {  
 **return cruiseName**;  
 }  
 **public static void** main(String args[]) {  
 System.***out***.println(**"Calling Constructor...."**);  
 Cruise c = **new** Cruise(**"Titanic X"**, **"Titatic"**, **"Hargeon Port"**, **"Port Blair"**, **"Hargeon port"**);  
 System.***out***.println(**"Cruise(\"Titanic X\", \"Titatic\", \"Hargeon Port\", \"Port Blair\", \"Hargeon port\")"**);  
 System.***out***.println(**"Calling printCruiseDetails() on created object...."**);  
 c.printCruiseDetails();  
 }  
}

Cruise.java

**public class** Cruise {  
 *// Class Variables* **private** String **cruiseName**;  
 **private** String **cruiseShipName**;  
 **private** String **departurePort**;  
 **private** String **destination**;  
 **private** String **returnPort**;  
 *// Constructor - default* Cruise() {}  
 *// Constructor - full* Cruise(String tCruiseName, String tShipName, String tDeparture, String tDestination, String tReturn) {  
 **cruiseName** = tCruiseName;  
 **cruiseShipName** = tShipName;  
 **departurePort** = tDeparture;  
 **destination** = tDestination;  
 **returnPort** = tReturn;  
 }  
 *// Accessors* **public** String getCruiseName() {  
 **return cruiseName**;  
 }  
 **public** String getCruiseShipName() {  
 **return cruiseShipName**;  
 }  
 **public** String getDeparturePort() {  
 **return departurePort**;  
 }  
 **public** String getDestination() {  
 **return destination**;  
 }  
 **public** String getReturnPort() {  
 **return returnPort**;  
 }  
 *// Mutators* **public void** setCruiseName(String tVar) {  
 **cruiseName** = tVar;  
 }  
 **public void** setCruiseShipName(String tVar) {  
 **cruiseShipName** = tVar;  
 }  
 **public void** setDeparturePort(String tVar) {  
 **departurePort** = tVar;  
 }  
 **public void** setDestination(String tVar) {  
 **destination** = tVar;  
 }  
 **public void** setReturnPort(String tVar) {  
 **returnPort** = tVar;  
 }  
 *// print cruise details* **public void** printCruiseDetails() {  
 *// working on this method  
 // made some changes to it* String spaces = **" "**;  
 System.***out***.print(**cruiseName**+spaces);  
 System.***out***.print(spaces+**cruiseShipName**+spaces);  
 System.***out***.print(**"\t"**+**departurePort**+**"\t"**);  
 System.***out***.print(**"\t"**+**destination**+**"\t"**);  
 System.***out***.print(**"\t"**+**returnPort**+**"\n"**);  
 }  
 *// method added to print ship's name vice memory address* @Override  
 **public** String toString() {  
 **return cruiseName**;  
 }  
 **public static void** main(String args[]) {  
 System.***out***.println(**"Calling Constructor...."**);  
 Cruise c = **new** Cruise(**"Titanic X"**, **"Titatic"**, **"Hargeon Port"**, **"Port Blair"**, **"Hargeon port"**);  
 System.***out***.println(**"Cruise(\"Titanic X\", \"Titatic\", \"Hargeon Port\", \"Port Blair\", \"Hargeon port\")"**);  
 System.***out***.println(**"Calling printCruiseDetails() on created object...."**);  
 c.printCruiseDetails();  
 }  
}

Passenger.java

**public class** Passenger {  
  
 *// Instance variables* **private** String **passengerName**;  
 **private** String **passengerCruise**;  
 **private** String **passengerRoomType**;  
  
  
 *// Constructor - default* Passenger() {  
 }  
  
 *// Constructor - full* Passenger(String pName, String pCruise, String pRoomType) {  
 **passengerName** = pName;  
 **passengerCruise** = pCruise;  
 **passengerRoomType** = pRoomType; *// should be BAL, OV, STE, or INT* }  
  
 *// Accessors* **public** String getPassengerName() {  
 **return passengerName**;  
 }  
  
 **public** String getPassengerCruise() {  
 **return passengerCruise**;  
 }  
  
 **public** String getPassengerRoomType() {  
 **return passengerRoomType**;  
 }  
  
 *// Mutators* **public void** setPassengerName(String tVar) {  
 **passengerName** = tVar;  
 }  
  
 **public void** setPassengerCruise(String tVar) {  
 **passengerCruise** = tVar;  
 }  
  
 **public void** setPassengerRoomType(String tVar) {  
 **passengerRoomType** = tVar;  
 }  
  
 *// print method* **public void** printPassenger() {  
 **int** spaceCount;  
 String spaces1 = **""**;  
 String spaces2 = **""**;  
 spaceCount = 20 - **passengerName**.length();  
 **for** (**int** i = 1; i <= spaceCount; i++) {  
 spaces1 = spaces1 + **" "**;  
 }  
 spaceCount = 20 - **passengerCruise**.length();  
 **for** (**int** i = 1; i <= spaceCount; i++) {  
 spaces2 = spaces2 + **" "**;  
 }  
  
 System.***out***.println(**passengerName** + spaces1 + **passengerCruise** + spaces2 +  
 **passengerRoomType**);  
 }  
  
 *// method added to print passenger's name vice memory address* @Override  
 **public** String toString() {  
 **return passengerName**;  
 }  
  
}