**Description:**

In this project, you will be using classes to model a Travel Agency.   
The top class in your hierarchy will be the Travel class. It will be an *abstract* superclass class. .

|  |
| --- |
| *Travel* |
| -destination:String -type:String -duration:int -activities:String [] |
| Travel( destination:String, type:String, duration:int, activities[]:String): +getDestination( ): String +getType ( ): String +getDuration( ): int +setDestination(destination:String):void +getActivities(): String [] +setType(type:String):void +setDuration(duration:int):void +addActivity(activity:String):void +removeActivity(activity:String):void +*describeTrip( ):String* (abstract) +*showAccommodations( ):String* (abstract) +*listActivities():void (abstract)* |

|  |
| --- |
| Cruise |
| -cruiseLine:String -shipName:String -roomNumber:int -accommodationType:String |
| Cruise(destination:String, duration:int, cruiseLine:String, shipName:String, roomNumber:int, accommodationType:String, activity[]:String)  Cruise(destination:String, cruiseType:String, duration:int, cruiseLine:String, shipName:String, roomNumber:int, accommodationType:String, activity[]:String) +getCruiseLine():String +getShipName():String +getRoomNumber():int +getAccommodationType():String +setCruiseLine(cruiseLine:String):void +setShipName(shipName:String):void +setRoomNumber(roomNumber:int):void +setAccommodationType(accommodation:  String):void  +describeTrip():String +showAccommodation():String +listActivities(): void |

The Travel class is an *abstract* superclass and it holds features that are common to all of its subclasses. For example, all travel types have a destination, a type of vacation and a length of vacation or duration. Note that there is a 4-arg constructor.  
  
Each of the private variables has a corresponding getter and setter method. There are three abstract methods in this class that will be implemented by each subclass of Travel. There are different descriptions for each of these methods so they will be overridden in each subclass.  
The showAccommodation( ) method is also an *abstract* method which will be implemented by each sub class of travel. There are different descriptions of each accommodation, so we’ll over-ride this method in each sub class.

The Cruise class is a sub class of *Travel* and contains two constructors:

A seven arg constructor used when creating a cruise object and an eight arg constructor which is called when you create an object of the Cruise subclass RiverCruise.

It also has getter and setter methods for all of the properties utilized by this class.

The describeTrip method will override the abstract method of the Travel class and will state the destination, the cruise line, the ship’s name, and how long the cruise is.

The showAccommodation method will override the abstract method of the Travel class and will state what room you are in and what type of room it is.  
  
The listActivities method will print the contents of the activities array.

|  |
| --- |
| RiverCruise |
| -river:String |
| RiverCruise(destination:String, duration:int, cruiseLine:String, shipName:String, roomNumber:int, accommodationType:String, activity[]:String, river:String)  +getRiver():String +setRiver(river:String):void  +describeTrip():String |

|  |
| --- |
| Resort |
| -resortName:String -unitNumber:int -unitType:String |
| Resort(destination:String, duration:int, resortName:String, unit:int, uType:String, activity[]:String)  Resort(destination:String, duration:int, resortName:String, resortType:String, unit:int, uType:String, activity[]:String) +getResortName():String +getUnitNumber():int +getUnitType():String +setResortName(resortName:String):void +setUnitNumber(unitNumber:int):void +setUnitType(unitType:String):void  +describeTrip():String +showAccommodation():String +listActivities(): void |

|  |
| --- |
| AllInclusive |
| -drinkLimit:int -diningFacilities:int |
| AllInclusive(destination:String, duration:int, resortName:String, unit:int, unitType:String, activity[]:String, limit:int, facilities:int)  +getDrinkLimit():int +getDiningFacilities():int +setDrinkLimit(drinkLimit:int):void +setDiningFacilities(diningFacilities:int):void  +describeTrip():String |

The RiverCruise class is a sub class of Cruise and represents a unique kind of cruise   
The constructor takes eight arguments and calls the eight argument constructor in its parent class.

Aside from the mutator and accessor method, you only have to be concerned with the describeTrip method here. It will call its parents describeTrip method and then add addition information about the river that this cruise will navigate along.

Resort is a direct subclass of Travel and contains two constructors. The six arg constructor is used to create a basic resort object. The seven arg constructor will be called from the AllInclusive class which extends from Resort.

Please note that you will have three accessor and three mutator methods along with both utility methods.

DescribeTrip will basically tell everyone what type of vacation it is, what the resort name is, where it is located, and how long the vacation will last.

showAccommodation will explain the type of unit you are assigned to and what the unit number is. .

listActivities will work through the array of activities and print them to the screen.

The AllInclusive class is a sub class of resort and represents the fact that food and drink are provided at no additional charge. The constructor takes eight arguments and calls the seven argument constructor in its parent class.

Aside from the mutator and accessor methods, you only have to be concerned with the describeTrip method here. It will call its parents describeTrip method and then add additional information about the number of drinks you are allowed daily and how many dining facilities exist at the