**Lab 1**

|  |  |  |  |
| --- | --- | --- | --- |
| Student Name | | Student CSUSM ID | Contribution percentage |
| 1 | Lauren Gonzalez | gonza823 | 50 |
| 2 | Sirena Murphree | murph135 | 50 |

**Grading Rubrics (for instructor only):**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Criteria | 1. Beginning | 2. Developing | 3. Proficient | 4. Exemplary |
| Mapping from design to Java code | 0-9 | 10-14 | 15-19 | 30 |
|  |  |  |  |
| Program: quality ->  *Readability* | 0-2 | 3-5 | 6-9 | 10 |
|  |  |  |  |
| Program: quality ->  *Modularity* | 0-2 | 3-5 | 6-9 | 10 |
|  |  |  |  |
| Program: quality ->  *Simplicity* | 0-2 | 3-5 | 6-9 | 10 |
|  |  |  |  |
| Updated design:  *correctness* | 0-9 | 10-14 | 15-19 | 20 |
|  |  |  |  |
| Updated design:  *Consistency with code* | 0-9 | 10-14 | 15-19 | 20 |
|  |  |  |  |
| Total Grade (100) |  | | | |

**Problems:**

1. In the following design in UML class diagram, some classes are incomplete (lack of attributes and/or operations). You should update the class diagram by adding important attributes and/operations that are appropriate.



1. Translate your complete design into Java implementation. Remember, the goal is to make sure the implementation is consistent with the design.

**Solution:**

* First, remember to zip the src folder of your project and submit the zip file to the ungraded assignment named “Lab1CodeSubmission”. One submission from each team.
* Paste all you source code here.
* Paste your updated UML class diagram below.
* Save this report in PDF, then **each student** needs to submit the pdf report to the graded assignment named “Lab1ReportSubmission”.

Diagram

Description automatically generated

Schedule.java

**import** java.time.ZonedDateTime;

**public** **class** Schedule {

/\*\*

\* departure time

\*/

**private** ZonedDateTime start\_time;

/\*\*

\* arrival time

\*/

**private** ZonedDateTime end\_time;

/\*\*

\* rout taken

\*/

**private** Route on\_route;

/\*\*

\* trip this schedule is for

\*/

**private** Trip for\_trip;

/\*\*

\* vehicle used

\*/

**private** Vehicle via\_vehicle;

/\*\*

\* constructor

\* **@param** start\_time departure time

\* **@param** end\_time arrival time

\* **@param** on\_route rout taken

\* **@param** for\_trip trip this schedule is for

\* **@param** via\_vehicle vehicle used

\*/

**public** Schedule(ZonedDateTime start\_time, ZonedDateTime end\_time, Route on\_route, Trip for\_trip, Vehicle via\_vehicle) {

**this**.setStart\_time(start\_time);

**this**.setEnd\_time(end\_time);

**this**.setOn\_route(on\_route);

**this**.setFor\_trip(for\_trip);

**this**.setVia\_vehicle(via\_vehicle);

}

/\*\*

\* **@return** the start\_time

\*/

**public** ZonedDateTime getStart\_time() {

**return** start\_time;

}

/\*\*

\* **@param** start\_time the start\_time to set

\*/

**public** **void** setStart\_time(ZonedDateTime start\_time) {

**this**.start\_time = start\_time;

}

/\*\*

\* **@return** the end\_time

\*/

**public** ZonedDateTime getEnd\_time() {

**return** end\_time;

}

/\*\*

\* **@param** end\_time the end\_time to set

\*/

**public** **void** setEnd\_time(ZonedDateTime end\_time) {

**this**.end\_time = end\_time;

}

/\*\*

\* **@return** the on\_route

\*/

**public** Route getOn\_route() {

**return** on\_route;

}

/\*\*

\* **@param** on\_route the on\_route to set

\*/

**public** **void** setOn\_route(Route on\_route) {

**this**.on\_route = on\_route;

}

/\*\*

\* **@return** the for\_trip

\*/

**public** Trip getFor\_trip() {

**return** for\_trip;

}

/\*\*

\* **@param** for\_trip the for\_trip to set

\*/

**public** **void** setFor\_trip(Trip for\_trip) {

**this**.for\_trip = for\_trip;

}

/\*\*

\* **@return** the via\_vehicle

\*/

**public** Vehicle getVia\_vehicle() {

**return** via\_vehicle;

}

/\*\*

\* **@param** via\_vehicle the via\_vehicle to set

\*/

**public** **void** setVia\_vehicle(Vehicle via\_vehicle) {

**this**.via\_vehicle = via\_vehicle;

}

}

Trip.java

**import** java.util.ArrayList;

**public** **class** Trip {

/\*\*

\* general origin of trip

\*/

**private** String origin;

/\*\*

\* general destination of trip

\*/

**private** String destination;

/\*\*

\* list of schedules the trip will take

\*/

**private** ArrayList<Schedule> scheduled\_for;

/\*\*

\* constructor

\* **@param** origin the trip's origin

\* **@param** destination the trip's destination

\*/

**public** Trip(String origin, String destination) {

**this**.origin = origin;

**this**.destination = destination;

**this**.scheduled\_for = **new** ArrayList<Schedule>();

}

/\*\*

\* **@return** the origin

\*/

**public** String getOrigin() {

**return** origin;

}

/\*\*

\* **@return** the destination

\*/

**public** String getDestination() {

**return** destination;

}

/\*\*

\* **@param** scheduled\_event the Schedule to be added to scheduled\_for

\*/

**public** **void** pushSchedule(Schedule scheduled\_event) {

**this**.scheduled\_for.add(scheduled\_event);

}

}

Route.java

**import** java.util.ArrayList;

**public** **class** Route {

/\*\*

\* the Address of the route's start\_point

\*/

**private** Address start\_point;

/\*\*

\* the Address of the route's end\_point

\*/

**private** Address end\_point;

/\*\*

\* the list of schedules that take this route

\*/

**private** ArrayList<Schedule> scheduled\_on;

/\*\*

\* constructor

\* **@param** start the Address of the route's start\_point

\* **@param** end the Address of the route's end\_point

\*/

**public** Route(Address start, Address end) {

**this**.start\_point = start;

**this**.end\_point = end;

**this**.scheduled\_on = **new** ArrayList<Schedule>();

}

/\*\*

\* **@return** the start\_point

\*/

**public** Address getStart\_point() {

**return** start\_point;

}

/\*\*

\* **@return** the end\_point

\*/

**public** Address getEnd\_point() {

**return** end\_point;

}

/\*\*

\* **@param** scheduled\_event the Schedule to be added to scheduled\_on

\*/

**public** **void** pushSchedule(Schedule scheduled\_event) {

**this**.scheduled\_on.add(scheduled\_event);

}

}

Address.java

**public** **class** Address {

/\*\*

\* street address

\* "285 J St"

\*/

**private** String street\_address;

/\*\*

\* city

\* "San Diego"

\*/

**private** String city;

/\*\*

\* state or providence

\* "California"

\*/

**private** String state\_providence;

/\*\*

\* postal code

\* "92101"

\*/

**private** String postal\_code;

/\*\*

\* country

\* "USA"

\*/

**private** String country;

/\*\*

\* constructor

\* **@param** street\_address

\* **@param** city

\* **@param** state\_providence

\* **@param** postal\_code

\* **@param** country

\*/

**public** Address(String street\_address, String city, String state\_providence, String postal\_code, String country) {

**this**.street\_address = street\_address;

**this**.city = city;

**this**.state\_providence = state\_providence;

**this**.postal\_code = postal\_code;

**this**.country = country;

}

/\*\*

\* print formated address

\* 285 J St

\* San Diego, CA 92101

\* USA

\*/

**public** **void** print() {

System.***out***.printf("%s\n%s, %s %s\n%s", street\_address, city, state\_providence, postal\_code, country);

}

}

Trail.java

**public** **class** Trail **extends** Route {

/\*\*

\* enumeration type of different terrains a trail can be

\*/

**public** **enum** Terrain {

***FOOT\_TRAIL***,

***BIKEWAY***,

***BOARDWALK***,

***NATURE***,

***MULTI\_USE***

}

/\*\*

\* enumeration different levels of experience required for hiking a trail

\*/

**public** **enum** Level{

***BEGINNER***,

***INTERMEDIATE***,

***SKILLED***,

***EXPERT***

}

/\*\*

\* name of the park the trail goes through

\*/

**private** String park\_name;

/\*\*

\* terrain of the trail

\*/

**private** Terrain primary\_terrain;

**private** Level primary\_level;

/\*\*

\* constructor

\* **@param** start the Address of the route's start\_point

\* **@param** end the Address of the route's end\_point

\* **@param** park\_name name of the park the trail goes through

\* **@param** primary\_terrain terrain of the trail

\* **@param** primary\_level level of the trail

\*/

**public** Trail(Address start, Address end, String park\_name, Terrain primary\_terrain, Level primary\_level) {

**super**(start, end);

**this**.setPark\_name(park\_name);

**this**.setPrimary\_terrain(primary\_terrain);

**this**.setPrimary\_level(primary\_level);

}

/\*\*

\* **@return** the park\_name

\*/

**public** String getPark\_name() {

**return** park\_name;

}

/\*\*

\* **@param** park\_name the park\_name to set

\*/

**public** **void** setPark\_name(String park\_name) {

**this**.park\_name = park\_name;

}

/\*\*

\* **@return** the primary\_terrain

\*/

**public** Terrain getPrimary\_terrain() {

**return** primary\_terrain;

}

/\*\*

\* **@param** primary\_terrain the primary\_terrain to set

\*/

**public** **void** setPrimary\_terrain(Terrain primary\_terrain) {

**this**.primary\_terrain = primary\_terrain;

}

/\*\*

\* **@return** the primary\_level

\*/

**public** Level getPrimary\_Level() {

**return** primary\_level;

}

/\*\*

\* **@param** primary\_level the primary\_level to set

\*/

**public** **void** setPrimary\_level(Level primary\_level) {

**this**.primary\_level = primary\_level;

}

}

Highway.java

**public** **class** Highway **extends** Route {

/\*\*

\* enumeration of the Cardinal Directions

\*/

**enum** CardinalDirection{

***NORTH***,

***SOUTH***,

***EAST***,

***WEST***,

***NORTHWEST***,

***NORTHEAST***,

***SOUTHWEST***,

***SOUTHEAST***

}

/\*\*

\* general direction to travel down the highway

\*/

**private** CardinalDirection direction;

/\*\*

\* the speed limit of the highway

\*/

**private** **int** speed\_limit;

/\*\*

\* constructor

\* **@param** start the Address of the route's start\_point

\* **@param** end the Address of the route's end\_point

\* **@param** direction general direction to travel down the highway

\* **@param** speed\_limit the speed limit of the highway

\*/

**public** Highway(Address start, Address end, CardinalDirection direction, **int** speed\_limit) {

**super**(start, end);

**this**.direction = direction;

**this**.speed\_limit = speed\_limit;

}

/\*\*

\* **@return** the direction

\*/

**public** CardinalDirection getDirection() {

**return** direction;

}

/\*\*

\* **@return** the speed\_limit

\*/

**public** **int** getSpeed\_limit() {

**return** speed\_limit;

}

}

FlightRoute.java

**public** **class** FlightRoute **extends** Route {

/\*\*

\* flight number assigned to flight route

\*/

**private** **int** flight\_num;

/\*\*

\* number of layovers

\* 0 if direct flight

\*/

**private** **int** num\_of\_layovers;

/\*\*

\* constructor

\* **@param** start the Address of the route's start\_point

\* **@param** end the Address of the route's end\_point

\* **@param** flight\_num flight number assigned to flight route

\* **@param** num\_of\_layovers number of layovers

\*/

**public** FlightRoute(Address start, Address end, **int** flight\_num, **int** num\_of\_layovers) {

**super**(start, end);

**this**.flight\_num = flight\_num;

**this**.num\_of\_layovers = num\_of\_layovers;

}

/\*\*

\* constructor

\* **@param** start the Address of the route's start\_point

\* **@param** end the Address of the route's end\_point

\* **@param** flight\_num flight number assigned to route

\*/

**public** FlightRoute(Address start, Address end, **int** flight\_num) {

**this**(start, end, flight\_num, 0);

}

/\*\*

\* **@return** the flight\_number

\*/

**public** **int** getFlight\_number() {

**return** flight\_num;

}

/\*\*

\* **@return** the num\_of\_layovers

\*/

**public** **int** getNum\_of\_layovers() {

**return** num\_of\_layovers;

}

}

Vehicle.java

**import** java.util.ArrayList;

**public** **class** Vehicle {

/\*\*

\* vehicle maker

\*/

**private** String maker;

/\*\*

\* vehicle model

\*/

**private** String model;

/\*\*

\* vehicle manufactured year

\*/

**private** **int** year;

/\*\*

\* list of schedules that vehicle is scheduled for

\*/

**private** ArrayList<Schedule> scheduled\_via;

/\*\*

\* constructor

\* **@param** maker vehicle maker

\* **@param** model vehicle model

\* **@param** year vehicle manufactured year

\*/

**public** Vehicle(String maker, String model, **int** year) {

**this**.maker = maker;

**this**.model = model;

**this**.year = year;

**this**.scheduled\_via = **new** ArrayList<Schedule>();

}

/\*\*

\* **@return** the maker

\*/

**public** String getMaker() {

**return** maker;

}

/\*\*

\* **@return** the model

\*/

**public** String getModel() {

**return** model;

}

/\*\*

\* **@return** the year

\*/

**public** **int** getYear() {

**return** year;

}

/\*\*

\* **@param** scheduled\_event the Schedule to be added to scheduled\_via

\*/

**public** **void** pushSchedule(Schedule scheduled\_event) {

**this**.scheduled\_via.add(scheduled\_event);

}

}

Airplane.java

**public** **class** Airplane **extends** Vehicle{

/\*\*

\* type of airplane

\*/

**private** String type;

/\*\*

\* constructor

\* **@param** maker vehicle maker

\* **@param** model vehicle model

\* **@param** year vehicle model year

\* **@param** type type of airplane

\*/

**public** Airplane(String maker, String model, **int** year, String type) {

**super**(maker, model, year);

**this**.type = type;

}

/\*\*

\* **@return** the type

\*/

**public** String getType() {

**return** type;

}

}

Bus.java

**public** **class** Bus **extends** Vehicle{

/\*\*

\* vehicle identification number

\*/

**private** String VIN;

/\*\*

\* registered license plate number

\*/

**private** String PlateNO;

/\*\*

\* constructor

\* **@param** maker vehicle maker

\* **@param** model vehicle model

\* **@param** year vehicle model year

\* **@param** VIN vehicle identification number

\* **@param** PlateNO registered license plate number

\*/

**public** Bus(String maker, String model, **int** year, String VIN, String PlateNO) {

**super**(maker, model, year);

**this**.PlateNO = PlateNO;

**this**.VIN = VIN;

}

/\*\*

\* **@return** the VIN

\*/

**public** String getVIN() {

**return** VIN;

}

/\*\*

\* **@return** the plateNO

\*/

**public** String getPlateNO() {

**return** PlateNO;

}

}

CharterBus.java

**public** **class** CharterBus **extends** Bus {

/\*\*

\* number of passenger seats

\*/

**private** **int** seat\_count;

/\*\*

\* constructor

\* **@param** maker vehicle maker

\* **@param** model vehicle model

\* **@param** year vehicle model year

\* **@param** VIN vehicle identification number

\* **@param** PlateNO registered license plate number

\* **@param** seat\_count number of passenger seats

\*/

**public** CharterBus(String maker, String model, **int** year, String VIN, String PlateNO, **int** seat\_count) {

**super**(maker, model, year, VIN, PlateNO);

**this**.seat\_count = seat\_count;

}

/\*\*

\* **@return** the seat\_count

\*/

**public** **int** getSeat\_count() {

**return** seat\_count;

}

}

TourBus.java

**public** **class** TourBus **extends** Bus {

/\*\*

\* number of beds

\*/

**private** **int** bed\_count;

/\*\*

\* under-cab storage capacity

\*/

**private** **float** storage\_capacity;

/\*\*

\* constructor

\* **@param** maker vehicle maker

\* **@param** model vehicle model

\* **@param** year vehicle model year

\* **@param** VIN vehicle identification number

\* **@param** PlateNO registered license plate number

\* **@param** bed\_count number of beds

\* **@param** storage\_capacity under-cab storage capacity

\*/

**public** TourBus(String maker, String model, **int** year, String VIN, String PlateNO, **int** bed\_count, **float** storage\_capacity) {

**super**(maker, model, year, VIN, PlateNO);

**this**.bed\_count = bed\_count;

**this**.storage\_capacity = storage\_capacity;

}

/\*\*

\* **@return** the storage\_capacity

\*/

**public** **float** getStorage\_capacity() {

**return** storage\_capacity;

}

/\*\*

\* **@return** the bed\_count

\*/

**public** **int** getBed\_count() {

**return** bed\_count;

}

}

ExpressBus.java

**public** **class** ExpressBus **extends** Bus {

/\*\*

\* safe passenger occupancy limit

\*/

**private** **int** passenger\_limit;

/\*\*

\* Mechanism exist that facilitates passenger to request a stop

\*/

**private** **boolean** stop\_request\_bell;

/\*\*

\* constructor

\* **@param** maker vehicle maker

\* **@param** model vehicle model

\* **@param** year vehicle model year

\* **@param** VIN vehicle identification number

\* **@param** PlateNO registered license plate number

\* **@param** passenger\_limit safe passenger occupancy limit

\*/

**public** ExpressBus(String maker, String model, **int** year, String VIN, String PlateNO, **int** passenger\_limit, **boolean** stop\_request\_bell) {

**super**(maker, model, year, VIN, PlateNO);

**this**.passenger\_limit = passenger\_limit;

**this**.stop\_request\_bell = stop\_request\_bell;

}

/\*\*

\* **@return** the passenger\_limit

\*/

**public** **int** getPassenger\_limit() {

**return** passenger\_limit;

}

/\*\*

\* **@return** the stop\_request\_bell

\*/

**public** **boolean** isStop\_request\_bell() {

**return** stop\_request\_bell;

}

}