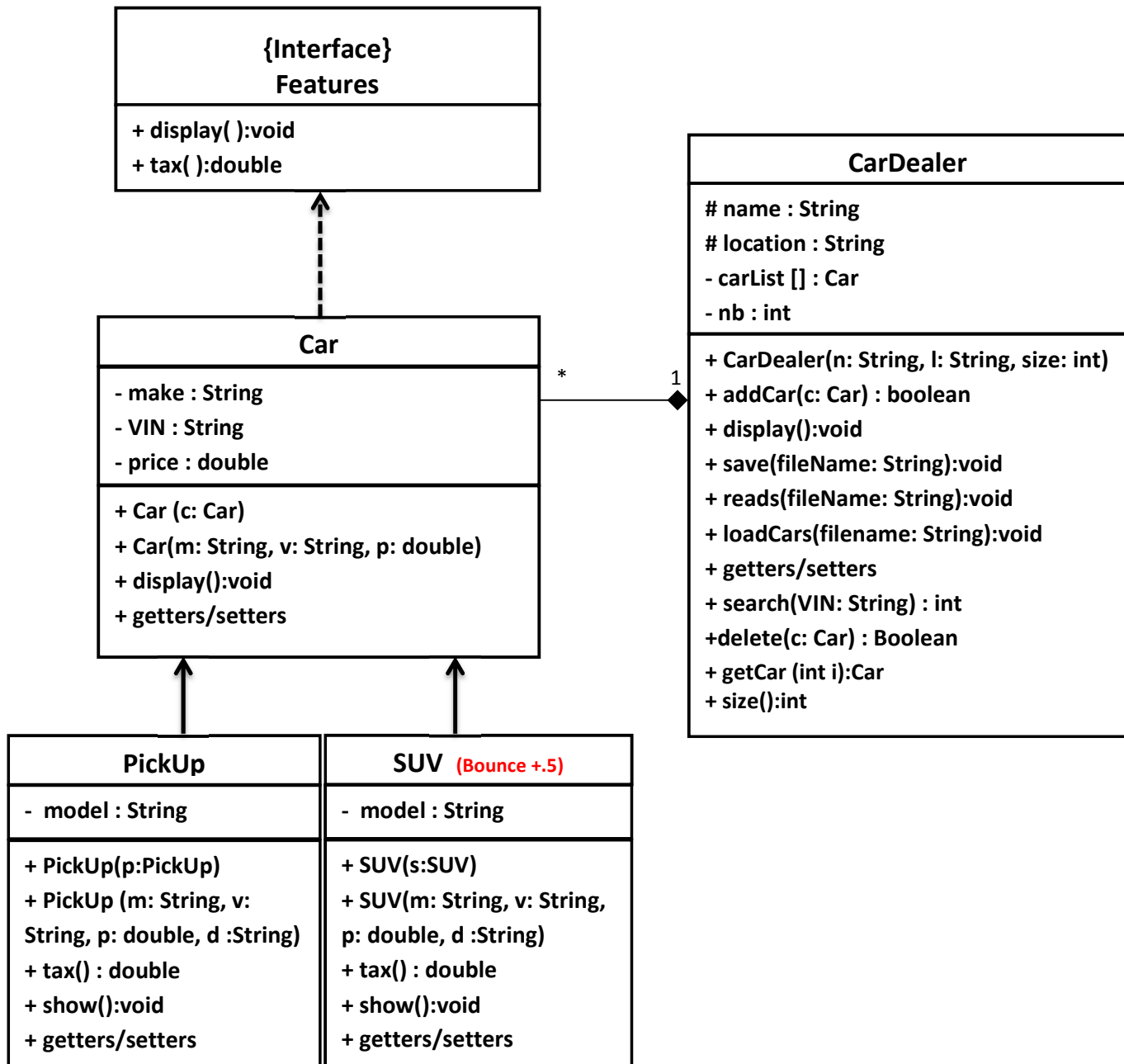


King Saud University
Collage of Computer and Information Siences
Lab Exam: Wednesday 17/12/2014



- Protect any class that is not intended to be instantiated (making it abstract).
- Protect any method that is not intended to be overridden (making it final).
- Protect any class that is not intended to be inherited (making it final).

King Saud University
Collage of Computer and Information Siences
Lab Exam: Wednesday 17/12/2014

Car Class:

- **Car(m: String, v: String, p: double):** Constructor with parameters for make, VIN and price of the Car.
- **Car(c: Car):** Copy constructor.
- **setters/getters:** Setters and getters for make, VIN and price.
- **display():** Show make, VIN and price.

CarDealer Class:

- **CarDealer (n: String, l: String, size: int):** Constructor with parameters for attributes name and location for the CarDealer and the size for the carList array. If the size is negative, the constructor must throw a *NegativeArraySizeException*. +1
- **addCar(c: Car):** Add a Car to the CarDealer and return true if successfully added. Otherwise return false. The method throws an *ArrayIndexOutOfBoundsException* when an illegal index term in the array has been accessed. *Hint: use InstanceOf* +1
- **save(fileName: String):** Save all the information of the CarDealer (name, location, and all the Car objects) to a file with the name fileName which is passed to this method. Also you may save the array size to assist in the read and load methods. This method will throw an *IOException*.. +1
- **reads(fileName: String):** Read and print the content of file. Just assume there is a file with strings or integers in it. +1
- **loadCars(fileName: String):** Load/read all the information of the CarDealer from the file with the specified name fileName to the object's attributes. This also requires creating appropriate size for the array before filling it with CarDealer objects. *(Bonus +0.5)*
- **setters/getters:** Setters and getters for attributes name and location.
- **search(VIN:String):** Search for Car by VIN number in the CarDealer and return its index if found. Otherwise return -1. +1
- **delete(c: Car):** Delete a Car from the CarDealer and return true if successfully deleted. Otherwise return false. *Hint use search* +1
- **getCar (i: int) :** Return the Car object from the passed index.
- **size():** Return the number of objects in the Array.

King Saud University
Collage of Computer and Information Siences
Lab Exam: Wednesday 17/12/2014

PickUp & SUV Class:

- **Constructor ():** Copy Constructor and Constructor with parameters for model, make, VIN and price. *Hint Call Super*
- **tax ():** Return Price/8 for PickUp , Return Price/10 for SUV.
- **show():** Show make, VIN and price, model and Tax

Main Class:

Write the class main to test your work and do the following:

1. Create two PickUp objects. +.25
2. Create two SUV objects.
3. Create CarDealer object. +.25
4. Add the Cars to the CarDealer. +.25
- 5 Show the cars at the CarDealer . +.25
- 6.Save the CarDealer list to a file named "CarDealerCars" +.25
7. Delete one PickUp from CarDealer. +.25
8. Show the result after delete. +.25
9. Show the Tax for the other PickUp. +.25
10. Show the Tax for the SUV.

Don't forget to catch the thrown exceptions.

(Bonus +1)

Make a menu like this:

1. **Add new Car (PickUp) or (SUV)**
(Ask the user for Name, Make, VIN, and price, then add it to CarDealer)
2. **Display all Cars**
(Display all CarDealer information)
3. **Save to file**
(Ask the user for a filename, and then save CarDealer to it)
4. **Load from file**
(Ask the user for a filename, then load CarDealer from it and print the content)
5. **Exit Program**

(Bonus +1)

In class CarDealer, write the method Sort(arr:int []) That takes an array of integers and sort it in ascending order (1,2,3,..10) then print it.

(Bonus +1)

Modify Class Car and CarDealer to be a linked list. Write methods AddFront and DeleteFront.

King Saud University
Collage of Computer and Information Siences
Lab Exam: Wednesday 17/12/2014

```
import java.io.Serializable;

public class Car implements Serializable {
    private String Make;
    private String VIN;
    private double Price;

    public String getMake() {
        return Make;
    }

    public void setMake(String make) {
        Make = make;
    }

    public String getVIN() {
        return VIN;
    }

    public void setVIN(String vIN) {
        VIN = vIN;
    }

    public double getPrice() {
        return Price;
    }

    public void setPrice(double price) {
        Price = price;
    }

    public Car(Car c) {
        this.Make = c.Make;
        this.VIN = c.VIN;
        this.Price = c.Price;
    }

    public Car(String m, String v, double p) {
        this.Make = m;
        this.VIN = v;
        this.Price = p;
    }

    public void display() {
        System.out.println("Make:" + Make + " - VIN:" + VIN + " - Price:"
            + Price);
    }
}
```