

Introduction to Object-Oriented Programming Lesson Final Assignment Documentation

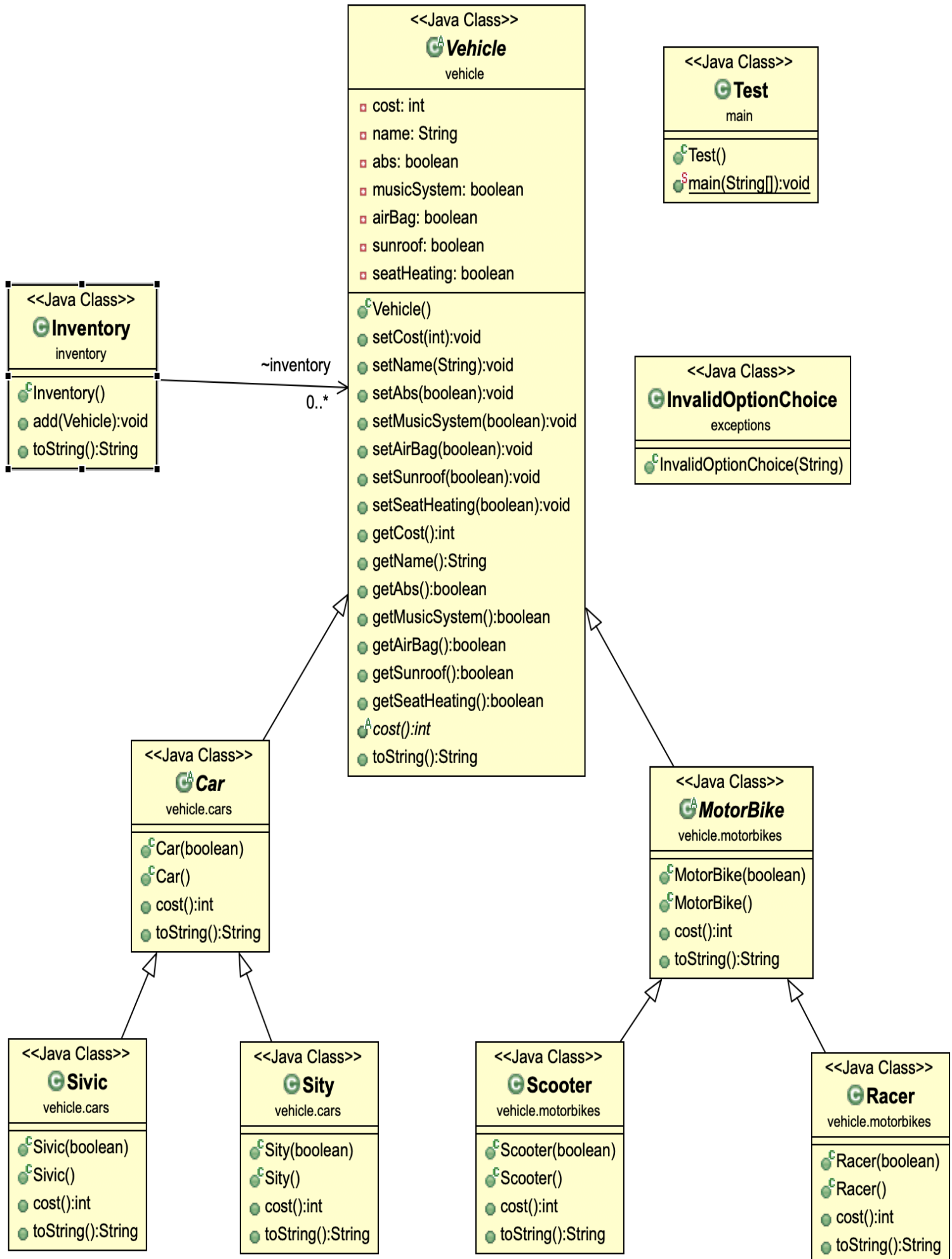
Student: Hasan Ali ÖZKAN

Number: 180709020

Instructor: Özgür KILIÇ

9 June 2021

This document is created by using LaTeX



Questions

B) What are the differences between an abstract class and a concrete class? Which class(es) can be defined as abstract classes which of them should be concrete in your code?

- Abstract class may contain abstract method but concrete class can't have.
- Abstract class can't be directly instantiated but concrete class can be directly instantiated.

Three of my classes are abstract(**Vehicle**, **Car**, **MotorBike**). These classes have to be abstract to prevent some error while adding them to **inventory** ArrayList in the **Inventory** class.(For example; If the **Vehicle** class would not be abstract then I can create an instance from it and I can add it to **inventory** ArrayList but we don't want that. It is same as for **Car** and **MotorBike** classes).

Five of them are have to be concrete(**Inventory**, **Sivic**, **Sity**, **Racer**, **Scooter**). Because I am creating instances from them in the **Test** class.I have an **InvalidOptionChoice** class that is also concrete and throwable.

C) What is Encapsulation? Have you applied encapsulation in your implementation? Explain where and how you applied.

Encapsulation is used for hiding details of classes is also called information hiding. All of my attributes are **private** to access them I used **getter** and **setter** methods this places are where I used encapsulation.

D) What are the advantages of Inheritance in Object Oriented Programming? Have you used inheritance in your implementation? Which super class(es) have you used and what did you benefit from these super classes?

I think code reusability is the most important advantages of Inheritance. We can reuse super-class's methods and attributes in our subclasses by extending them. We can add new feature to superclass's methods or we can update superclass's attributes in our subclasses. In short we can modify all codes of superclass just as we want in our subclasses.

Yes I have used three super classes these are **Vehicle**, **Car** and **MotorBike**. I put all my attributes and methods in the most superclass **Vehicle** and change them specifically for each subclasses. I got the benefit of Inheritance by reusing the codes in the **Vehicle** class into my subclasses

E) Explain the usage of final keyword in java. Can you use final keyword in any part of your implementation? Explain why or why not?

We can use final keyword in three contexts. The usage is the same for each of them.

- If we use final keyword before variables that means we cannot change its value any more.
- If we use final keyword before any method that means we cannot override it anymore.
- If we use the final keyword before the class that means we can not extend is from another class or we can say this class can't be a superclass anymore.

Yes I can use it on some of my method such as **add** method in the **Inventory** class or I can use getter and setter methods in the **Vehicle** class since I don't override them. I can also

use final keyword some of my classes that are *Inventory*, *Sivic*, *Sity*, *Racer*, *Scooter* and *InvalidOptionChoice* since I don't extend them from another class.

F)Have you benefited from polymorphism in your implementation. If yes, copy the code segment where you use polymorphism to your report and give the name of the polymorphic variable.

Yes I got the benefit of polymorphism in my Test class. I created *Sivic* and *Sity* objects from the *Car* class on the same variable which is *car* and I created *Racer* and *Scooter* objects from the *MotorBike* class on the same variable which is *motorBike*. Actually If I want I can create all cars and motorbikes on the same object which is created on the *Vehicle* class. I didn't it since I don't want to reduce the clarity.

```
Car car = new Sivic(); //I created a Sivic objects from the Car class and
                        //I assigned it to car variable

car = new Sity(); //This time I created a Sity objects from the Car class and
                //again I assigned it to car variable

MotorBike motorBike = new Racer(); //I created a Racer objects from the
                                   //MotorBike class and I assigned it to

                                   //motorBike variable

motorBike = new Scooter(); //This time I created a Scooter objects from the
                           //MotorBike class and assign it to
                           //motorBike variable
```
