Exercise 6

In this exercise you have to create a simple library of several classes. This task will help you to learn about class declaration, object creation and inferitance.

The structure of the library is shown in the figure 1:

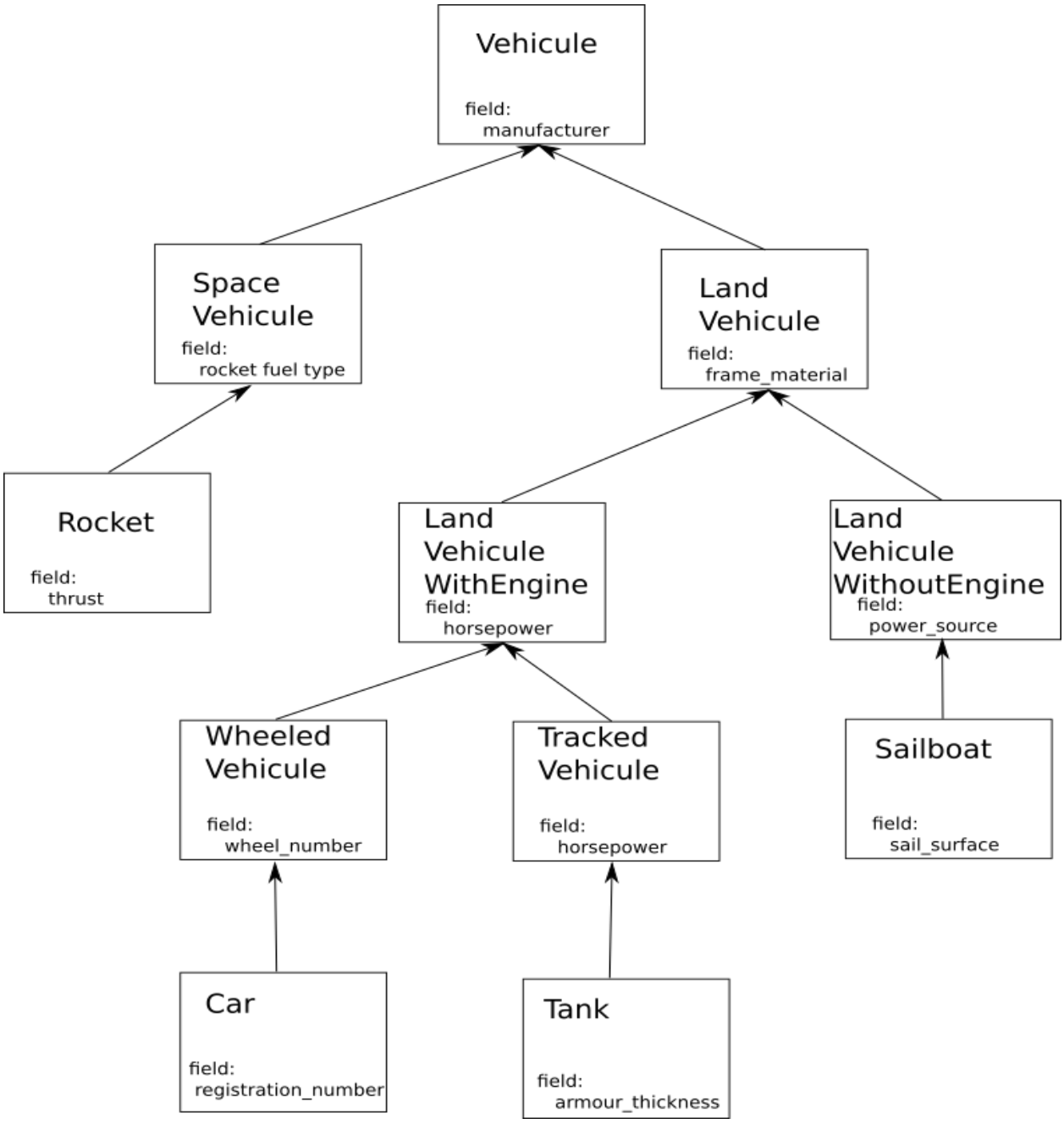

Fig.1 Example class library with inferitance

In the figure 1, each box represents a class. The name of the class is given inside the box. Every class contains one unique field that is given inside the box. For simplicity, define all fields in the library as type String, even for fields that are numbers (like thrust or horsepower).

Task to do in the exercise 6 :

1. Create a new GUI project with a form that contains one button.
2. Create all classes from the library shown in figure 1. For each class create a new file. The Vehicule class and the subtree leading to Rocket class are given below in example 2.
3. After doubleclick on the button in the design window, enter the code from the example 1. Then fill the code by creating objects for all classes from Fig.1. After that, add these objects to the ArrayList like others.

Note!

- The name of the classes should be written in one line (see example 2 SpaceVehicule).
- You create a new class for library by right click on the package (in the project tree) and choosing New Java Class from the menu.
- In Java, you have to use the **same name for the name of the file and the name of the class**.
- Code from example 1 will produce output in console (seen inside NetBeans IDE). If you want, you can use TexArea to produce code inside GUI.

Example 1

(code inside the button pressed event funtion)

```
Vehicule vehicule1 = new Vehicule("Audi");
SpaceVehicule spaceVehicule1 =
    new SpaceVehicule("RP-1 + LOX","NASA");
Rocket rocket1 =
     new Rocket("3034 kN","RP-1 + LOX","ArianeSpace");

//create new library objects

ArrayList<Vehicule> vehiculeTable =
                                    new ArrayList<Pojazd>(0);
vehiculeTable.add(vehicule1);
vehiculeTable.add(spaceVehicule1);
vehiculeTable.add(rocket1);
//add code here

for(int i=0; i<vehiculeTable.size();i++)
  System.out.println( vehiculeTable.get(i).toString()  );
```

Example 2

(Vehicule, SpaceVehicule and Rocket classes definitions)

```
public class Vehicule
{
    public String manufacturer;

    public Vehicule(String manufacturer)
    {
        this.manufacturer = manufacturer;
    }

    public String toString() {
        return "Vehicule produced by "+manufacturer;
    }
}

public class SpaceVehicule extends Vehicule
{
    public String rocket_fuel_type;

    public SpaceVehicule(String rocket_fuel_type,
                         String manufacturer)
    {
        super(manufacturer);
        this.rocket_fuel_type = rocket_fuel_type;
    }

    public  String toString() {
        return "Vehicule produced by "+manufacturer+
              " fuel type is "+rocket_fuel_type;
    }
}
```

```
public class Rocket extends  SpaceVehicule
{
    public String thrust;

    public Rakieta(String thrust, String rocket_fuel_type,
                   String manufacturer)
    {
        super(paliwo_rakietowe, producent);
        this.nazwa = nazwa;
    }
    public  String toString() {
        return "Vehicule produced by "+manufacturer+
              " fuel type is "+rocket_fuel_type+
              " thrust is "+thrust;
    }
}
```