Nama : M. ALFIN NUR KHILMI
NIM : 19051397046
Kelas : D4 MI 2019 B

1.1. Ubahlah class diagram diatas ke dalam bentuk standar UML class diagram

Jawab :

Robot
 # nama : String **# pemilik : String** **# tahun : int**
setTahunPembuatan(int tahun) : void **setNama(String nama) : void** **displayData() : void**

1.2. Buatlah 2 kelas berdasarkan digram kelas diatas (beserta kelas Main)

Jawab :

- Class Robot

```
Public abstract class Robot {

Protected String nama ;
Protected String pemilik ;
Protected int tahun ;
Public void setTahun Pembuatan {
This tahun =  tahun ;
}
Public void setNama
Public void displayData () ;
    System.out.println("nama:  " + nama) ;
    System.out.println("pemilik: " + pemilik) ;
    System.out.println("tahun: " + tahun) ;

}
```

- Class DoraMini

```
Public class  Robot {

Public class DoraMini extends Robot {
System.out.println("Halo, Saya Dora Mini") ;
}
Public void setNama (String nama) {
Super.nama = nama ;
}
```

```
Public void displayData() {
System.out.println("Nama:  " + super. nama) ;
System.out.println("Pemilik: " + super. pemilik) ;
System.out.println("Tahun: " + super.tahun) ;
}
}
```

- Main Class

```
Public class MainAbstrack {
    Public static void main (Atring args[]) {
    // Robot r = new Robot () ;
    dm.setNama("Dora Mini Satu") ;
    dm.setTahunPembuatan(2014) ;
    dm.displayData() ;
    dm.sayDora
    }
}
```

2. Buatlah kode berdasarkan abstract class diatas dan Buatlah classes lain yang menggunakan abstract kelas

   Jawab :

```
abstract class permainan{

   String namaPemain;
   int levelPemain;

   public void setnamaPemain (String namaPemain){
      this.namaPemain = namaPemain;
   }

   public String getnama (){
      return this.namaPemain;
   }

   public void setlevelpemain (int level){
     this.levelPemain = level;
   }

   public int getlevel (){
      return this.levelPemain;
   }
```

```
    public void jalankan(){
        System.out.println("Nama Player     : " + getnama());
        if (this.levelPemain <=20){
            System.out.println("Level Player    : Normal" );
        }
        else if(this.levelPemain <=80){
            System.out.println("Level Player    : Medium");
            }
            else if(this.levelPemain<=100){
                System.out.println("level Player    : hard");
                }
                else {
                    System.out.println("Level Maksimum = 100");
                }
    }

    public abstract int hitungskor(int hit, int miss);

}

class permainanArcade extends permainan{

    public int hitungskor(int hit, int miss){
        return hit * 3 - miss * 1;

    }
}

class permainanStrategy extends permainan{

    public int hitungskor(int hit, int miss){
        return hit * 5;

    }
}

class app {
    public static void main(String[] args) {

        System.out.println("Ini Arcade");
        permainan aPermainan = new permainanArcade();

        aPermainan.setnamaPemain("namaPemain");
        aPermainan.setlevelpemain(20);
```

```
        aPermainan.jalankan();
        System.out.println("Hit Pemain      : " + aPermainan.hitungskor(321, 321)) ;

        System.out.println("");

        System.out.println("Ini Strategy");
        permainan bPermainan = new permainanStrategy();

        bPermainan.setnamaPemain("Andre");
        bPermainan.setlevelpemain(70);

        bPermainan.jalankan();
        System.out.println("Hit Pemain      : " + bPermainan.hitungskor(321, 321)) ;

    }
}
```

3.1 Tulislah dan jelaskan perbedaan antara abstract class dan interface

Jawab :

- Abstract class adalah sebuah class OOP yang tidak dapat diinstansi atau dibuat objeknya dan biasanya berisi fitur-fitur dari sebuah class yang belum diimplementasikan.
- Interface adalah sebuah class yang semua methodnya adalah abstract method. Karena methodnya abstract class maka diimplementasikan oleh child class juga.

Abstract Class	Interface
Bisa beris abstract dan nin-abstract method	Hanya boleh berisi abstrac method
Method boleh bersifat static	Method tidak boleh bersifat static
Suatu abstract class hanya bisa meng-extend satu abstract class lainnya	Suatu interface bisa meng-extend satu atau lebih interface lainnya

4. Buatlah 3 kelas berdasarkan diagram kelas diatas (dan juga class Main untuk membentuk objeknya)

- Class Robot

```
Public interface Robot {
    void setNama(String nama) ;
    void setTahun(int tahun) ;
    void displayData() ;
```

}

- Class Doraemon

```
Public interface Doraemon{
    void sayDora() ;
    void displayKantongAjaib() ;
}
```

- Class Doramini

```
Public class DoraMini implements Robot,Doraemon {
    String nama;
    String pemilik;
    int tahun;

 public string getPemilik(){
 this.pemilik =pemilik;
 }

  public void setPemilik(string pemilik){
  this.pemilik = pemilik;
  }

  @Override
   public void setNama(String nama){

  // TODO: Implement this method

  System.out.println("nama "+nama);
   }


  @Override
   public void setTahun(int tahun) {

  // TODO: Implement this method

  System.out.println("tahun "+tahun);
  }

  @Override
  public void displayData() {
```

```
        // TODO: Implement this method

        System.out.println("a");
        }


    @Override

    public void sayDora()
    {
        // TODO: Implement this method

        System.out.println("Hallo, saya Dora Mini");
    }


    @Override

    public void displayKantongAjaib()
    {
        // TODO: Implement this method

        System.out.println("Saya juga seperti Doraemon yang memiliki kantung ajaib");
    }


}

public class Main implements Doraemon, Robot
{

    @Override

    public void setNama(String nama)
    {
        // TODO: Implement this method
        System.out.println("My name is "+nama);
    }

    @Override

    public void sayDora()
    {
        // TODO: Implement this method
```

```
        System.out.println("Hallo, saya Dora Mini");
    }

    @Override

    public void displayKantongAjaib()
    {
        // TODO: Implement this method

        System.out.println("Saya juga seperti Doraemon yang memiliki kantung ajaib");
    }


}
```