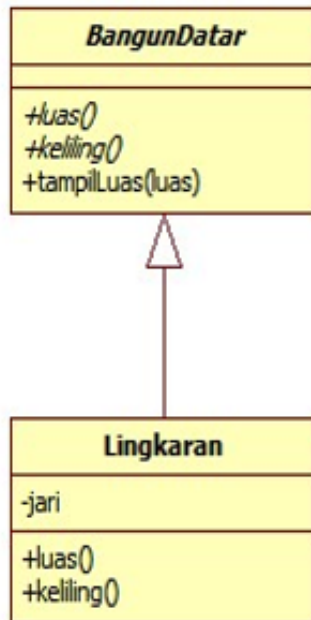


Nama	: Dhanar Agastya Rakalangi
NRP	: 5002221075

1. Buatlah Coding dari UML berikut :



BangunDatar berupa kelas abstract
Sedangkan Lingkaran berupa kelas
konkret/reguler.

1. Tambahkan di kelas abstract metoda tampilKeliling(keliling)
2. Tambahkan metoda static void metodaReguler() yang menampilkan "Perhitungan Luas dan keliling"
3. Lengkapilah kelas utama berikut ini

```
public class TugasKamis1
{
    public static void main(String[] s){
        Lingkaran li = new Lingkaran(7.0);
        .....;
        .....;
        .....;}
}
```

Output hasil running :
Perhitungan Luas dan keliling
Luas = 154
Keliling = 44

Jawab:

```
abstract class BangunDatar {  
    abstract int luas();  
    abstract int keliling();  
  
    void tampilLuas(int luas) {  
        System.out.println("Luas = " + luas);  
    }  
  
    void tampilKeliling(int keliling) {  
        System.out.println("Keliling = " + keliling);  
    }  
  
    static void metodaReguler() {  
        System.out.println("Perhitungan Luas dan  
            keliling");  
    }  
}  
  
class Lingkaran extends BangunDatar {  
    int radius;  
  
    Lingkaran(double radius) {  
        this.radius = (int) radius;  
    }  
    int luas() {  
        return (int) Math.round(Math.PI * radius *  
            radius);  
    }  
    int keliling() {  
        return (int) Math.round(2 * Math.PI * radius);  
    }  
}  
  
public class TugasKamis1 {  
    public static void main(String[] args) {  
        Lingkaran obj = new Lingkaran(7.0);  
        BangunDatar.metodaReguler();  
        obj.tampilLuas(obj.luas());  
        obj.tampilKeliling(obj.keliling());  
    }  
}
```

2. Betulkan kesalahan yang ada pada potongan coding berikut Dan tampilkan outputnya.

```
public class TugasKamis2 {  
    public static void main(String args[])  
    {  
        try  
        {  
            int x =0;  
            int y=5/x; }  
        catch (Exception e)  
        {  
            System.out.println("Exception"); }  
        catch (ArithmeticException ae)  
        {  
            System.out.println("Arithmetic Exception"); }  
        System.out.println("finished");  
    }  
}
```

Jawab:

```
public class TugasKamis2 {  
    public static void main(String args[]) {  
        try {  
            int x = 0;  
            int y = 5 / x;  
        } catch (ArithmeticException ae) {  
            System.out.println("Arithmetic-Exception");  
        } catch (Exception e) {  
            System.out.println("Exception");  
        }  
        System.out.println("finished");  
    }  
}
```

Output :

Arithmetic Exception
finished

3. Lengkapilah program

```
interface AA
{.....}

abstract class BB
{.....}

class CC extends BB implements AA{
    public void displayB(){
        System.out.println("Abstract Display-B");
    }
    public void displayA() {
        System.out.println("Display-A");
    }
}

public class TugasKamis3 {
    public static void main(String args[])
    {.....}
}
```

Tambahkan metode ini pada
sebuah bagian coding
public void display()
{
 System.out.println("Display-B");
}

Jika hasil running TugasKamis3 :
Display-B
Display-A
Abstract Display-B

Jawab:

```
interface AA {
    void displayA ();
}

abstract class BB {
    abstract void displayB ();
}

class CC extends BB implements AA {
    public void displayB () {
        System.out.println("Abstract-Display-B");
    }

    public void displayA () {
        System.out.println("Display-A");
    }

    public void display () {
        System.out.println("Display-B");
    }
}

public class TugasKamis3 {
    public static void main(String args []) {
        CC obj = new CC();
        obj.display();
        obj.displayA();
        obj.displayB();
    }
}
```