Java - Inheritance; abstract classes

Parent class

Child class extends Parent class

super (something that Parent has)

Parent

- Data
- Methods

Child

- More data
- More methods
- Other versions
public class PositivePoly extends Poly {
    public PositivePoly () {
        super(); // call to parent constructor
    }
    public PositivePoly (double[] a) {  // throws Exception
        super(a); // Poly(a) -> allocates coefficients
                    // -> copy elements from a
        for (int i=0; i < coefficients.length; i++)
            if (coefficients[i] < 0)
                throw new Exception("No neg coeff");
    }
    // Similarly
    public PositivePoly (Poly p) throws Exception {
        super (p); // Poly (p)
        // check as above
Poly → p.add(2)
Positive Poly → p.add(2)

// in PositivePoly

Positive public PositivePoly add (PositivePoly g) {
    Poly n = super.add(g);
    // look at parent methods; call add
    // children can be used as variables of their parent type
    // NOT the other way around!
    return new PositivePoly (n);

    // alternatively
    return new PositivePoly (super.add(g));
public class TestPoly {

    public static void main(String[] args) {

        PositivePoly p = new PositivePoly(4, 1, 2, 3);
        PositivePoly q = new PositivePoly(4, 4, 5);
        PositivePoly r = p.add(q);

        r.displayPoly(); // Same effect as for any Poly
        // print coef. array
    }
}

3

What extensions are allowed in Java

public class A {
    ...
}

public class B extends A {
    ...
}

public class C extends B {
    ...
}

public class D extends A {
    ...
}

public class E extends B, D {
    ...
}

In Java this is not allowed!

May be naming conflicts
```java
public class A extends B { ... }
public class B extends C { ... }
public class C extends A { ... }
```

Compiler error:

Circular (class would end up depending on itself)
Abstraction → Circular objects

Circular class
- Some things are known: all such objects will need a radius, \( \pi \)
- Some things are not known: exact formula to compute area.

public abstract class Circular {
    not everything will be specified now
    double radius;
    // \( \pi \) is a constant; every body shares it
    // no one can re-declare it
    static final double \( \pi \) = 3.14...
    // shared among all objects of this class
    no children can re-declare/modify
    
    // constructor
    // accessor/mutator for radius
    public abstract double getArea();
    // this method will be written by children.
public class Circle extends Circular {
    // radius
    // constructor

    public double getArea () {
        return PI * radius * radius;
    }

    // MUST provide an implementation for all abstract methods
    // same signature for methods
    // else this class would be abstract too

    public class Sphere extends Circular {

        public double getArea () {

        }
    }
}