abstract classes

Every shape has a position (x, y) in the plane, so use a super class Shape to hold the point. Subclass has necessary fields to describe shape.

Motivating abstract classes

b(1).area() is illegal, even though each subclass object has function area().

Don’t want to cast down. Instead, define area( ) in Shape.

Motivating abstract classes

Area( ) in class Shape doesn’t return useful value. Public double area( ) { return 0.0; }

Problem: How to force subclasses to override area( )?

Problem: How to ban creation of Shape objects.

Abstract class and method solves both problems

Abstract class. Means can’t create object of Shape:

c new Shape( ); syntactically illegal

c new Shape Công;

public abstract double area( )

public abstract double area( )

Place abstract method only in abstract class.

Body is replaced by:

Use abstract classes? Seems OK, because method bodies not given!

But Java does not allow this, because abstract classes can have non-abstract methods.

Instead, Java has a construct, the interface, which is like an abstract class but has more restrictions.

Can extend only one class

public class C extends C2 { }

Use abstract classes? Seems OK, because method bodies not given!

But Java does not allow this, because abstract classes can have non-abstract methods.

Instead, Java has a construct, the interface, which is like an abstract class but has more restrictions.

Use abstract classes? Seems OK, because method bodies not given!

But Java does not allow this, because abstract classes can have non-abstract methods.

Instead, Java has a construct, the interface, which is like an abstract class but has more restrictions.

Can extend only one class

public class C extends C2 { }
Casting with interfaces

class B extends A implements C1, C2 { … }

interface C1 { … }  

class A { … }  

Want to sort b by shape areas. Avoid duplication of effort!  

Don't want to write a sort procedure —many already exist. Avoid duplication of effort!

Circle@x  

A extends B implements C1, C2 { … }

class B extends A implements C1, C2 { … }  

…  

Object

b

class

B

C1

C2

…

object

b

B

Circle

Rect

Trian

Object

Class arrays has many other useful static methods

Beauty of interfaces:

Arrays.sort(Comparable[] a)  

Java Library static methods:

Arrays.sort(Comparable[] a)  

Casts needed so that area() can be used. If c is not a Shape, exception thrown.

public static void sort(Comparable[] a) { … }