Topics: Method overloading, OOP review

Reading (JV): Sec 4.3 (review 4.0-4.2, 5.0, 5.1)

Method overloading

Each method has a signature: the method name and the parameter types (including the order). In a class, methods can have the same name as long as the signatures are different. The constructors in a class are often overloaded.

class Interval {
    private double base;                      // low end
    private double range;                     // interval width
    public final static double maxWidth = 5;  // max width of interval

    public Interval(double base, double r)
    { this.base = base; range = Math.min(r,maxWidth); }
    public Interval() {}
    public Interval(double base)
    { this.base = base; range = maxWidth; }

    public double getEnd() { ... }
    public boolean isIn(Interval o) { ... }
    public static Interval overlap(Interval a, Interval b) { ... }
    public String toString() { ... }
}

Invoking methods

Assume three Intervals have been instantiated: i1, i2, i3. Assume i1 and i2 overlap. Write code to find if the overlapped Interval of i1 and i2 is in Interval i3.

Why doesn’t this swap work?

/* Try to swap two values */
public class BadSwap {
    public static void main(String[] args){
        int x = Keyboard.readInt();  int y = Keyboard.readInt();
        swap(x,y);
    }
    public static void swap(int x, int y) {
        int tmp;
        tmp = x;
        x = y;
        y = tmp;
    }
} //class BadSwap
A different class

/* Organize data for any Person: name, age, best friend */
class Person {
    private String name;
    private int age;

    public final static int MATURE=18;

    // Constructor
    public Person(String name, int age) {
        this.name = name;
        this.age = age;
    }

    // Determine if self is an adult
    public boolean isAdult() { return age >= MATURE; }

    // Make a friend

    // Become a friend

    // Show object data
    public String toString() {
        return name + " is " + age;
    }
}

public class Client {
    public static void main(String[] args){
        Person a = new Person("AJ",9);
        Person b = new Person("BP",7);
        a.beFriendOf(b);
        a.makeFriend(b);
    }
}