```
/** A Dice (or Die!) */
class Dice {

   private int top;    // top face
   private int sides;  // number of sides

   /** A Dice has numFaces sides and the top face is random */
   public Dice(int numFaces) {
     sides = numFaces;
     roll();
   }

   /** top gets a random value in 1..sides */
   public void roll() { setTop(randInt(1,getSides()));  }

   /** = random int in low..high */
   public int randInt(int low, int high) {
     return (int) (Math.random()*(high-low+1))+low;
   }

   /** = Get top face */
   public int getTop() { return top; }

   /** = Get number of sides */
   public int getSides() { return sides; }

   /** Set top face -- a random value */
   public void setTop(int face) { top= face; }

   /** = String description of this Dice */
   public String toString() {
     return  getSides() + "-sided dice shows face " + getTop();
   }

}

//***********************************************************

/** A TrickDice has one weightedSide such that the
 *  weightedSide appears weight times as often as other sides
 */
class TrickDice extends Dice {

  private int weightedSide;  //weighted side appears more often
  private int weight;        //weighted side appears weigth
                             //times as often as other sides

  /** TrickDice has side s appearing with weight w */
  public TrickDice(int numFaces, int s, int w) {
    super(numFaces);
    weightedSide= s;
    weight= w;
    roll();
  }
  /** top gets random value in 1..sides given trick property */
  public void roll() {
    //int r= (int) (Math.random()*(getSides()+weight-1))+1;
    int r= randInt(1,(getSides()+weight-1));
    if (r<=getSides())
      setTop(r);
    else
      setTop(weightedSide);
  }

  /** = Get weighted side */
  public int getWSide() { return weightedSide; }

  /** = Get weight of weighted side */
  public int getWeight() { return weight; }

  /** = String description of this Dice */









}


//*************************************************************

class Room {
  private static int nextID = 1; //id of next room to be
                                 //created
  protected int id; //room number
  private int mess; //mess level

  /** A Room has unique id and messiness level mess */
  public Room(int mess) { this.mess = mess;
                          id = nextID;   nextID++; }

  /** = String description of this Room */
  public String toString() { return "Room " + id; }

  /** Reduce mess by 1 but keep mess>=0 */
  public void clean() { mess--;
                        if (mess<0)  mess=0; }

  /** Print status of Room */
  public void report() { System.out.println(toString() +
                         ", has mess level " + mess);     }

  /** Print how many rooms have been created */
  public static void countRooms() {
    System.out.println((nextID-1)+ " rooms in total"); }
} //class Room
```