```
import java.util.ArrayList;
import java.util.Arrays;
import java.util.Collections;
import java.util.Scanner;

import javax.print.attribute.IntegerSyntax;

public class BulgarianSolitaire {

 public static void main(String[] args) {

 Scanner input  = new Scanner(System.in);
 System.out.println("Please enter an integer k for  the upperbound to calculate a triangular number:");
 int k = input.nextInt();
 int triangular = 0;
 for (int i = 1; i <= k; i++) {
  triangular = triangular + i;
 }
 System.out.println("Triangular number: "+ triangular);
 ArrayList<Integer> piles = new ArrayList<Integer>();
 //



 int add = 0;
 int add_sum = 0; // creates random piles of virtual cards
 while(sum(piles) != triangular) {
  int lowerbound = 1;
   int upperbound = triangular - add_sum;
  add = (int) (Math.random() * (upperbound - lowerbound + 1) + lowerbound);
  add_sum = add_sum + add;
  piles.add(add);
 }

 System.out.println("random piles array:" + piles);

 int step_num = 1;
 while(!CheckTriangular(piles)) {  //while piles is not triangular
  SolitaireStep(piles);
  System.out.println("Step #" +step_num + ": " + piles);
  step_num++;
 }
 System.out.println("final: " + piles);

 input.close();

}
public static ArrayList<Integer> SolitaireStep(ArrayList<Integer> array){ //returns the next soitair step
 //int size = piles.size();
 for (int i = 0; i < array.size(); i++) { //replaces the current values with values -1
  array.add(i, array.get(i) - 1);
  array.remove(i + 1);
 }
 int lastint = array.size();
 for (int i = 0; i < array.size(); i++) { //deletes 0
  if (array.get(i) ==0) {
   array.remove(i);
   i = i - 1;
  }
 }
 array.add(lastint);  //adds all of the subtracted cards to the ends of the pile
 return array;
}

 public static Boolean CheckTriangular(ArrayList<Integer> array) { //returns true if the numbers are triangular in the array
```

```
66
67    ArrayList<Integer> temp = new ArrayList<Integer>();
68    temp = array;
69    Collections.sort(temp);
70
71    if(temp.size() == 1) {
72     if (temp.get(0) ==1) {
73      return true;
74     }else {
75      return false;
76     }
77    }
78
79    boolean bool  = true;
80    for (int i = 0; i < temp.size(); i++) {
81     if(i == 0 && temp.get(i) !=  temp.get(i +1)-1) {
82      bool = false;
83     }else if(i == temp.size() -1 && temp.get(i) != temp.get(i-1)+1) {
84      bool = false;
85     }else if(i != 0 && i != temp.size() -1 && (temp.get(i) != temp.get(i-1) +1 || temp.get(i) !=  temp.get(i +1)-1)){
86      bool = false;
87     }
88    }
89
90    return bool;
91   }
92
93   public static int sum(ArrayList<Integer> array) {
94
95    int sum = 0;
96    for(Integer d : array)
97        sum += d;
98    return sum;
99
100   }
101
102
103  }
```