```java
import java.util.ArrayList;
import java.util.Collections;
import java.util.Random;


public class BulgarianSolitaire {

    public static void main(String[] args) {
        bulgarianSolitaire(45);
    }

    public static void bulgarianSolitaire(int numCards) {
        // Check if given number of cards is triangular
        int n = (int) Math.sqrt(2 * numCards);
        if (n * (n + 1) / 2 != numCards) {
            System.out.println(numCards + " is not triangular");
            return;
        }

        // Create an ArrayList to represent piles
        ArrayList < Integer > piles = new ArrayList < > ();
        Random random = new Random();

        // Randomly divide the cards into piles
        for (int i = 0; i < n; i++) {
            int cardsInPile = random.nextInt(numCards) + 1;
            piles.add(cardsInPile);
            numCards -= cardsInPile;
        }

        // Print the initial configuration
        System.out.println("Initial Configuration: " + piles);

        // Play Bulgarian Solitaire until the final configuration is reached
        while (!isFinalConfiguration(piles)) {
            piles = performSolitaireStep(piles);
            System.out.println("After Solitaire Step: " + piles);
        }

        System.out.println("\nFinal Configuration Reached!");
    }

    private static ArrayList < Integer > performSolitaireStep(ArrayList < Integer > piles) {
        ArrayList < Integer > newPiles = new ArrayList < > ();

        for (int pile: piles) {
            if (pile > 1) {
                newPiles.add(pile - 1);
            }
        }

        newPiles.add(piles.size());

        return newPiles;
    }

    private static boolean isFinalConfiguration(ArrayList < Integer > piles) {
        int expectedSize = 1;

        for (int pile: piles) {
            if (pile != expectedSize) {
                return false;
            }
            expectedSize++;
        }

        return true;
    }
}
```

PDF document made with CodePrint.org