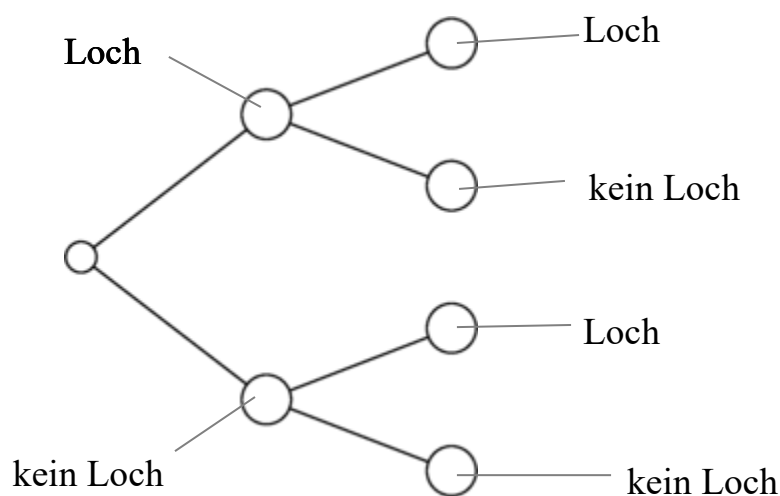


## Dokumentation Aufgabe 4a

### Lösungsidee

Das Programm soll ausrechnen wie viele Möglichkeiten es für die Schlüsselkarten gibt. Beide Teilaufgaben kann man mit Kombinatorik lösen. Da die Reihenfolge wichtig ist und es zu Wiederholungen kommt, handelt es sich um Variation mit Wiederholungen. Man kann sich das ganze als einen Baum vorstellen. Das Erste Loch ist ganz oben und hat zwei Möglichkeiten: Entweder es existiert oder eben nicht. Das gleiche gilt auch für die folgenden Löcher. Jedes Loch hat zwei Möglichkeiten und am Ende braucht man nur alle möglichen Pfade zu addieren. Das ganze kann man sich in etwa so vorstellen:



Bei der ersten Teilaufgabe sind von 25 Schlüssellöchern 2 festgelegt, nämlich die die ein Loch bzw. keins haben. Diese beiden Löcher befinden sich links und rechts oben. Das heißt, dass nur noch 23 Löcher übrig bleiben bei denen man entscheiden muss, ob es ein Loch gibt oder nicht.

Bei der zweiten Teilaufgabe muss die Schlüsselkarte symmetrisch sein, d.h. dass man 10 Löcher vernachlässigen kann, da diese ja identisch zu ihrem gespiegelten Partner-Loch sind. Somit gibt es nur noch 15 Löcher bei denen man entscheiden muss.

## Umsetzung

Da nun geklärt ist, wie die Theorie funktioniert müssen wir uns jetzt nur noch die konkrete Rechnung anschauen und in ein Programm umsetzen. Beide Aufgaben können mit folgender Gleichung gelöst werden:

Anzahl Schlüsselkarten = [Löcher, bei denen man eine Entscheidung treffen muss] (hoch die Anzahl Möglichkeiten)

oder noch konkreter: Teilaufgabe 1:  $AS = 23^2$

Teilaufgabe 2:  $AS = 15^2$

Die Klassen werden mit CamelCase und auf Englisch benannt. Außerdem wird eine Math-Methode von Java zu Hochrechnung genutzt.

## Beispiele

Da der Code so kurz ist, wird er hier vollständig mit den Beispielen aufgeführt.  
Teilaufgabe 1:

```
public class KeyHoles {  
  
    public static void main(String[] args) {  
  
        int possibilities = (int)Math.pow(2,23);  
  
        System.out.println(possibilities);  
  
    }  
}
```

Konsole: 8388608

Teilaufgabe 2:

```
public class KeyHolesSymmetry {  
  
    public static void main(String[] args) {  
  
        int possibilities = (int)Math.pow(2,15);  
  
        System.out.println(possibilities);  
  
    }  
}
```

Konsole: 32768