# Funktionell Nedbrytning

Joachim von Hacht

# Funktionell Nedbrytning (1)

```
int[][] m1 = {
        {1, 1, 3},
        {1, 2, 2},
        {1, 0, 3},
};
int[][] m2 = {
        {0, 1, 3},
        {1, 3, 2},
        {0, 1, 2},
};

// There are 2 pairs (x,y) with x from m1 and y from m2 that
// sums to 0
out.println(countPairs(m1, m2, 0));  // 2
// There are 8 pairs (x,y) with x from m1 and y from m2 that
// sums to 5
out.println(countPairs(m1, m2, 5)); // 8
```

Hur bryta ner metoden countPairs?

# Funktionell Nedbrytning (2)

```
int[][] matrix1 = {
        {7, 1, 3, 6},
        {6, 2, 7, 1},
        {8, 9, 1, 3},
        {5, 6, 9, 1},
};
int[][] matrix2 = {
        {7, 1, 3, 9},
        {6, 9, 7, 1},
        {7, 9, 1, 3},
        {5, 7, 9, 1},
};

// Output [1]
out.println(Arrays.toString(getCommonRowElements(matrix1)));
// Output [7, 1, 9]
out.println(Arrays.toString(getCommonRowElements(matrix2)));
```

Hur bryta ner metoden getCommonRowElements?

# Funktionell Nedbrytning (3)

```
int[][] matrix1 = {
        {7, 1, 3, 6},
        {6, 2, 7, 1},
        {8, 9, 1, 3},
        {5, 6, 9, 1},
};
int[][] matrix2 = {
        {7, 1, 3, 9},
        {6, 9, 7, 1},
        {7, 1, 3, 9},
        {7, 1, 3, 9},
};

// Output: 4
out.println(uniqueRows(matrix1));
// Output: 2
out.println(uniqueRows(matrix2));
```

Hur bryta ner metoden uniqueRows?

# Funktionell Nedbrytning (5)

```
String str1 = "Sirap i paris";
String str2 = "Madam I'm Adam";
String str3 = "Was  It A Rat I Saw?";
String str4 = "Can a get a hot dog?";
```

Hur bryta ner metoden isPalindrome?

```
out.println(isPalindrome(str1));     // True
out.println(isPalindrome(str2));     // True
out.println(isPalindrome(str3));     // True
out.println( ! isPalindrome(str4));  // True
```