

## Lab classes 9

In the lab-classes we experiment with *hash tables*.

1. You investigate hashing, using collision resolution by chaining and using the division method for the hash function.
2. For this the program `Chaining.java` is to be investigated.

### 1 Understanding the implementation

Look into `Chaining.java` and understand the code. Explain questions in the comments.

1. What are insert, search and delete doing?
2. Explain the output in the comments, when running the program with  $m = 9$  and  $m = 4$ .
3. Is it possible for other  $m$  to get all three displayed hash tables into a form where no slot contains more than one element?

### 2 Creating special cases

Change file `Experimenting.java` appropriately:

1. For  $m = 13$ , show (in code) how to insert 5 different elements which all occupy the same slot.
2. For the same  $m$ , write a loop which inserts keys  $1, \dots, 100$  into a hash table: Explain the distribution. Did the hash function a good job here?

### 3 Comparing different hash maps for equality

1. When would the method `equals` declare two hash tables as equal? Think especially about the case of having elements with equal keys. Is this appropriate? And why does this implementation not compile?

Explain the question in the comments.