OTH
OSTBAYERISCHE
TECHNISCHE HOCHSCHULE
REGENSBURG

IM
INFORMATIK UND
MATHEMATIK

Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

Lab Unit: Lambdas and Streams

Given the following list of city names:

```
List<String> cities = Arrays.asList( "Regensburg", "Basel", "Munich", "Bonn",
                                     "Hamburg", "Munich", "Berlin" );
```

Use the `cities` list and implement the following use cases using only streams and lambdas:

- print distinct list of cities on console:

```
Regensburg
Basel
Munich
Bonn
Hamburg
Berlin
```

- print first 3 cities in list:

```
Regensburg
Basel
Munich
```

- store in boolean variable whether city names have all at least 6 characters:

```
All names have length of at least 6 chars: false
```

- store list of distinct city names in descending order of name's length (and print to check):

```
Regensburg
Hamburg
Munich
Berlin
Basel
Bonn
```

- store set of city names in CAPITAL LETTERS in new TreeSet (and print to check):

```
BASEL
BERLIN
BONN
HAMBURG
MUNICH
REGENSBURG
```

- find first city name in natural order of list of given length `len` and – if present – store name in `String` variable or store string `"no city name of length ..."` (use terminal operation that returns `Optional<T>` object and continue using this object):

```
Basel   // for len == 5
no city name of length 11   // for len == 11
```

- print name of city with longest name (one if there are more):

```
Regensburg
```

- store length of longest (or shortest) city name (and print variable to check):

```
length of longest name: 10
// or: length of shortest name: 4
```

- reduce list of names to String of their initials:

```
Initials: RBMBHMB
```

- compute total sum of string length over all names (and print to check):

```
total string length over all names: 44
```

- store a `Map<Character,Long>` with number of cities grouped by their initials (and print to check):

```
B: 3   // there are 3 cities with names starting with "B" ...
R: 1
H: 1
M: 2
```

- as above but do not store but print directly to console

- as above but print map sorted by value:

```
R: 1
H: 1
M: 2
B: 3
```

- count number of letters in city names and print table to console sorted by key:

```
a: 2
b: 2
B: 3
c: 2
e: 4
g: 3
h: 2
H: 1
i: 3
l: 2
m: 1
M: 2
n: 6
o: 1
r: 3
R: 1
s: 2
u: 4
```