Assignment 1

- Finish the hash table
  - from the previous practical
    - keys String
    - values Object
    - methods at least
      - Object get(String key)
      - void set(String key, Object value)
      - iterator over keys

- add the forEachValue method, which applies an operation (supplied as an argument) on each value in the table

  - design the method so a lambda expression can be used as the operation
    - design a suitable functional interface

Assignment 2

- create a simple calculator that reads the input in the Reverse Polish notation (postfix notation)

```
1 2 3 + +
        =>   6
```

  - reads from the std input
  - prints out to the std output
  - only the int type
  - expressions are separated by a new-line

Tests...

Test 1

- What does the following program?

```
public class TestString {
  public static void main(String[] args) {
    String s = new String("Hello world");
    System.out.println(s);
  }
}

class String {
  private final java.lang.String s;
  public String(java.lang.String s) {
    this.s = s;
  }
  public java.lang.String toString() {
    return s;
  }
}
```

A cannot be compiled
B prints Hello world
C something else happens

The example based on code from *J. Bloch, N. Gafter: Java Puzzlers*

Test 2

- Is it possible to declare the class B, so that the program prints false? But without overriding the method equals!

```
public class A {
  public static void main(String[] args) {
    B b = new B();
    System.out.println(b.equals(b));
  }
}
```

Test 2

- Solution
  - overload the method equals
  - i.e. define the method

```
public boolean equals(B b) {
   return false;
}
```

- Continuation – and without overloading?

Test 2

- Also yes

```
class B {
  public B() {
    System.out.println(false);
    System.exit(0);
  }
}
```