

### Exercise 1 (5 points)

Describe the tree call from the below stack trace in the stdout console. Make an example of when you get such an exception.

- 1 Exception in thread "main" java.lang.IllegalArgumentException:
- 2 at it.unibz.Example.Engine.start(Class2.java:12)
- 3 at it.unibz.Example.Flight.takeOff(Class1.java:14)
- 4 at it.unibz.Example.Simulator.main(Simulator.java:20)

### Exercise 2. (8 points)

Write a method that receives an integer parameter *k* and a string *S*. Declare an array of Strings of a fixed size 10, and attempt to write the string *S* in the position *k* of the array. Use a try/catch structure to manage the `IndexOutOfBoundsException`. Use a Finally block to send a message to the standard output. Discuss a case when the exception is caught.

### Exercise 3. (12 points)

Draw the stack and the heap describing the execution of the following code.  
Global address in Stack = @200; RA of main @[-1]

```
[1] public class Railway {
[2]     public static void main(String[] args) {
[3]         Train myTrain = new Train();
[4]         myTrain.speed = 20;
[5]         myTrain.direction = "North";
[6]         Train yourTrain = new Train();
[7]         yourTrain.speed = 40;
[8]         yourTrain.direction = "South";
[9]         float diff = myTrain.diffSpeed(yourTrain);
[10]        String text = myTrain.compareDirection(yourTrain);
[11]        System.out.println("our trains run at " + diff + "kilometers/hrs of difference and have " +text + " direction");
[12]    }
[13] }
[14] public class Train {
[15]     public float speed;
[16]     public String direction;
[17]     public String compareDirection (Train train){
[18]         if(direction == train.direction){
[19]             return "the same";
[20]         }else{ return "opposite";}
[21]     }
[22]     public float diffSpeed(Train train){
[23]         float diff = train.speed - this.speed;
[24]         return diff;
[25]     }
[26] }
```

**Exercise 5 (5 points)**

Illustrate with examples the three new features of JDK 7 and higher: try-with-resources; multi-catch and re-throw with precision

**Exercise 6. (only to get 33 points, do not do it if you did not complete all the previous exercises)**

Describe the testing strategy using scaffolding. Describe with an example the concepts of stubs, drivers, oracles and harness.