

# Practical Course: Logic App Development

## Exercise sheet week 3

*To be completed before 3rd November 2023, 11am.*

1. Write a dart class `Dice` in a separate file for randomly rolling two dice. In this file, you should introduce two constants `minDie` and `maxDie` which are set to 1 and 6, respectively. The class `Dice` should have the following properties:

- A boolean `equalDistr`.
- An integer array `die` of length 2 for the latest throw of the two dice.
- An integer `numberOfThrows` for the number of throws of the two dice.
- An integer array `sumStatistics` which for each sum of the two dice counts the number of occurrences; i.e., how often did we get the sum 2, the sum 3, ..., or the sum  $2 \cdot \text{maxDie}$ .
- A two-dimensional integer array `dieStatistics` which for each combination of the two dice counts the number of occurrences; here, for instance, the combination (3,5) is different from (5,3).

The class `Dice` should have the following methods:

- `void throwDice() {...}` for throwing the two dice. You can use the random generator in `dart:math`. If `equalDistr` is false, then the two dice should be thrown independently of one another; the possible outcomes of every dice should have equal probability. If `equalDistr` is true, then the sum of the two dice should have equal probability (i.e., for two dice, each of the possible sums 2,...,12 should have the same probability). Moreover, for each sum, the possible combinations which yield this sum should all have the same probability (e.g., if the sum is 5, then (1,4), (2,3), (3,2), (4,1) should all have the same probability). Moreover, in any case, the method `throwDice` should update `numberOfThrows`, `sumStatistics` and `dieStatistics` accordingly.
- `void resetStatistics() {...}` for resetting all statistics to 0.

If the constants `minDie` and `maxDie` are changed to other values with `minDie`  $\leq$  `maxDie`, then the class `Dice` should still work accordingly (e.g., there could be dice with 8 faces and the values on the faces could range from 9 to 16).

2. Write a dart program in another file which takes a number  $n$  from the command line. The program should have one `Dice` object. Firstly, it should set `equalDistr` to false, call `throwDice` exactly  $n$  times, and then print `sumStatistics`. Then it should call `resetStatistics` and set `equalDistr` to true. Then the program should again call `throwDice` exactly  $n$  times and print `sumStatistics`.