

QAIS
(A.Y. 2023/2024) – Duration: 1h
July 24th, 2024

Exercise 1.

Consider the following program written to establish the total amount that a student has to pay to enroll at the university¹.

```
1 package fst.unicam.it;
2
3 public class StudentManagement {
4
5     public static Date lateEnrollmentDate; //this variable is set to 30/04/2024
6
7     public static double taxes2Pay(Student student) {
8         double lateEnrollmentRate = 0;
9         double toBePayed = 500;
10        double extraTax = 200;
11
12        switch (student.status) {
13            worker:
14                if (!(Date.isAfter(student.currentEnrollmentDate, lateEnrollmentDate)))
15                    toBePayed = toBePayed * 3;
16                else {
17                    if (student.firstEnrollmentDate.year < 2015)
18                        lateEnrollmentRate = 0.20;
19                    toBePayed = toBePayed * 3;
20                }
21                break;
22            foreigner:
23                if (!(Date.isAfter(student.currentEnrollmentDate, lateEnrollmentDate)))
24                    toBePayed = toBePayed * 4;
25                else {
26                    if (student.firstEnrollmentDate.year < 2015)
27                        lateEnrollmentRate = 0.30;
28                    toBePayed = toBePayed * 4;
29                }
30                break;
31            standard:
32                if (!(Date.isAfter(student.currentEnrollmentDate, lateEnrollmentDate)))
33                    toBePayed = toBePayed * 2;
34                else {
35                    if (student.firstEnrollmentDate.year < 2015)
36                        lateEnrollmentRate = 0.10;
37                    toBePayed = toBePayed * 2;
38                }
39            }
40        if (student.dateOfBirth.getYear() >= 2000)
41            lateEnrollmentRate = lateEnrollmentRate - 0.02;
42        extraTax = 200 * (1 + lateEnrollmentRate);
43        toBePayed = toBePayed + extraTax;
44        return toBePayed;
45    }
46 }
```

Apply a test derivation strategy that intends to spot all the possible mistakes (boolean, relational and expressions) related to the conditions of the program. As any good programmer, in case you judge it useful, you can revise the code to make it simpler and easier to understand. Obviously the global behaviour should not change.

16 points

¹At the end of the document you find all the complementary “java” classes needed to better understand the behaviour of the program. In the code some simplification have been adopted to make it less verbose. In particular the use of public attributes makes the execution of the exercise more concise.