

Process Documentation - B

B_1

- Minutes required: 32 minutes
- Black box testing through boundary value testing (omitting that we have written the tested function)
- Number of executions: Seven executions. Test-by-test basis, checking our assumptions on one variable at a time.
- We've used boundaries of variables domains for testing purposes

Number	EC	Condition	Valid EC	Invalid EC
1			point.getValue() ∈ [0, 10]	
2				point.getValue() < 0
3		point.getValue() ∈ [0,10]		point.getValue() > 10
4			point.getMaxValue() ∈ [0, 10]	
5				point.getMaxValue() < 1
6		point.getMaxValue() ∈ [0, 10]		point.getMaxValue() > 10
7			point.getType() ∈ [Laboratory, Seminar, Bonuses, Exam]	
8		point.getType() ∈ [Laboratory, Seminar, Bonuses, Exam, Granted]		point.getType() == Granted
9		point.getValue() > point.getMaxValue()		point.getValue() > point.getMaxValue()
10			allPoints.hasExam() == False	
11		point.getType() == Exam		allPoints.hasExam() == True

No TC	EC	input data			output data	
		value	maxValue	Type	expected	actual
1	1,4,7,10	8	10	Exam	1	1
2	2	-2	10	Seminar	0	0
3	3	13	10	Seminar	0	0
4	5,9	5	0	Laboratory	0	0
5	6	6	13	Laboratory	0	0
6	8	10	10	Granted	0	0
7	9	5	-7	Bonuses	0	0
8	11	7	10	Exam	0	0

No TC	BVA	input data			output data	
		value	maxValue	type	expected	actual
1	1	0	10	Exam	1	1
2		10	10	Seminar	1	1
3		-1	10	Seminar	0	0
4		1	10	Seminar	1	1
5		9	10	Seminar	1	1
6		11	10	Seminar	0	0
7	2	1	1	Laboratory	1	1
8		2	10	Laboratory	1	1
9		3	0	Laboratory	0	0
10		1	2	Laboratory	1	1
11		5	9	Laboratory	1	1
12		6	11	Laboratory	0	0
13	3	9	10	Bonuses	1	1

- Describe how you have solved the assignment. Did you use “manual” testing during implementation? If yes, with what data have you test it? How many executions before you declared “satisfied” with the implementation?

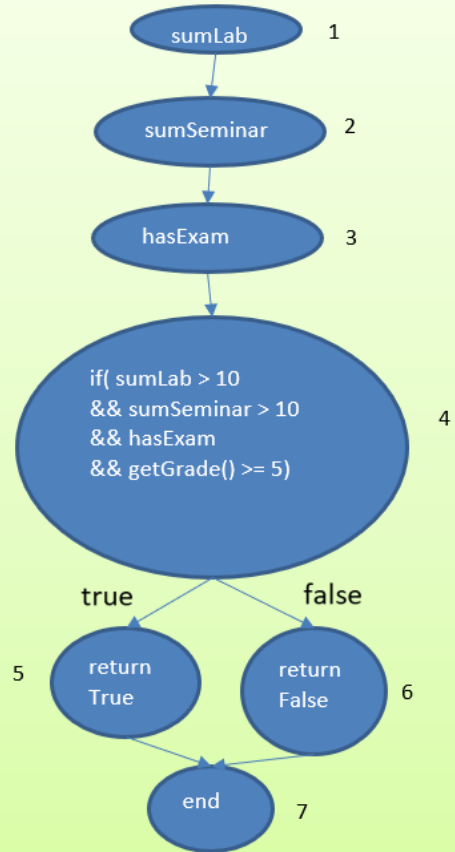
B_2

- Minutes required: 57 minutes
- Used white box testing since we had access to getGrade() method
- Number of executions: Seven executions. Test-by-test basis, checking our assumptions on one variable at a time.
- We've used boundaries of variable domains for testing purposes. Specifically we've played around the boundary of 10 for multiple variables

```

public boolean isPassing() {
    double sumLab = points.stream()
        .filter(p -> p.getType() ==
GradeType.Laboratory)
        .mapToDouble(Point::getPoint)
        .sum();
    double sumSeminar = points.stream()
        .filter(p -> p.getType() ==
GradeType.Seminar)
        .mapToDouble(Point::getPoint)
        .sum();
    // Validation already prevents more than one
Exam grade
    boolean hasExam =
points.stream().anyMatch(p -> p.getType() ==
GradeType.Exam);
    return sumLab > 10 && sumSeminar > 10 &&
hasExam && getGrade() >= 5;
}

```



Cyclomatic complexity		
CC_1	regions	2
CC_2	E-N+2	$7 - 7 + 2 = 2$
CC_3	Predicate+1	2

Individual Paths	
1	1,2,3,4,5,7
2	1,2,3,4,6,7