

Object Oriented Programming 2021/22

Project self-evaluation form

Oral discussion date Quinta 23/06 18:00

Group number 43

Student Number	Student Name	Percentage of participation (must sum 100)	Expected mark (0-10 points)
96244	João Vera	32,5	8
104086	Tiago Miranda	32,5	8
88035	Joel Neves Pastilha	25	7

Mark	Command/feature	Correctly implemented	Implemented with faults	Not implemented	Prof notes
UML					
5/20	Tool used: <u>Visual Paradigm</u> Was it done with reverse Engineering? <u>No</u>				
Basic game					
1/20	Bet command	x			
1/20	Credit command	x			
1/20	Deal command	x			
1/20	Hold command	x			
Strategy and statistics					
3/20	Perfect strategy		x		
0.5/20	Advice command	x			
0.5/20	Statistics command	x			
Modes					
1/20	Debug mode Reading card and command files and running commands	x			
1/20	Simulation mode Shuffling/Re-shuffling, commands with perfect strategy and statistics	x			
Documentation					
1/20	Examples of debug files Examples to test the game in debug mode	x			
2/20	Java doc Packages, interfaces, classes, methods, fields		x		

Visualization			
1.5/20	Correct in example files without errors A correct output should give a correct info in all commands/advice/statistics	Prof notes:	
0.5/20	Correct in example files <u>with</u> errors A correct output should not crash and give information to the user	Prof notes:	
Discounts			
-3/20	Interfaces and polymorphism used incorrectly	Give here the name of all interfaces in your project: Give here the name of all abstract classes in your project: Give here the name of all polymorphic methods in your project: hold build addCard removeCard execute	Prof notes:
-2/20	Open-closed principle used incorrectly	How many packages? 5 Visibility of the attributes (choose all used): - ~ # + + -	Prof notes:
-1/20	Object class / collections used incorrectly	Which classes override equals? Did you use any sorting method or sorted collection from Java? we sorted through Collections.sort with a custom comparator How is the deck shuffled? using Collections.shuffle() Did you provide your own exceptions? no, we used only java's	Prof notes:
-1/20	Incorrect data structures	Collections used (apart from arrays): ArrayList	Prof notes:
-1/20	Prints outside the format	Professor notes:	
-0.5/20	Problems with the executable Incorrect MANIFEST.MF and JAR w/out java sources		
-0.5/20	Files submitted outside the format Other compression than .zip and incorrect folders		
-2 ⁿ /day	Projects submitted after the established date		