Object Oriented Programming 2021/22
Project self-evaluation form

Oral discussion date _____ Group number 15

Student Number	Student Name	Percentage of participation (must sum 100)	Expected mark (0-10 points)
93047	Diogo Madeira Martins	33.(3)%	9
93106	José Carlos Oliveira Brito	33.(3)%	9
93202	Xavier Batista Fernandes	33.(3)%	9

Mark	Command/feature	Correctly implemented	Implemented with faults	Not implemented	Prof notes
UML					
5/20	Tool used: draw.io Was it done with reverse Engineering? No				
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 with 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: IMode, IVariant Give here the name of all abstract classes in your project: AbstractMode, AbstractDebug, AbstractSimulation, AbstractVariant, AbstractDoubleBonus Give here the name of all polymorphic methods in your project: play, bet, deal, hold, advice (both of them), credit, statistics, evaluate, getHandPayout, getHandKey, updateStats, Deck	Prof notes:
-2/20	Open-closed principle used incorrectly	How many packages? Three: cards, game, main Visibility of the attributes (choose all used): - ~ # +	Prof notes:
-1/20	Object class / collections used incorrectly	Which classes override equals? Card (to compare both Suit and Rank) Did you use any sorting method or sorted collection from Java? Arrays.sort(), to sort arrays of Cards and integers How is the deck shuffled? By applying the Collections.shuffle() method (Deck has a List of Cards) Did you provide your own exceptions? No	Prof notes:
-1/20	Incorrect data structures	Collections used (apart from arrays): List (implemented by LinkedList), Map (implemented by HashMap)	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		
-2ⁿ/day	Projects submitted after the established date		