

Object Oriented Programming 2021/22

Project self-evaluation form

Oral discussion date 23 / 06 16:00 Group number 6

Student Number	Student Name	Percentage of participation (must sum 100)	Expected mark (0-10 points)
93014	Andre Silva	34	9
93198	Vasco Rodrigues	33	9
96219	Gonçalo Midoes	33	9

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>Partially</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: <u>IAdvisor, ICardAnalyzer, IPayoutTable, IStats, IHand, IPlayer, IGame, IComments, IDeck</u> Give here the name of all abstract classes in your project: <u>Game, Deck</u> Give here the name of all polymorphic methods in your project: <u>toString, equals, giveHand, play, printDeck</u>	Prof notes:
-2/20	Open-closed principle used incorrectly	How many packages? <u>1 package that contains 4 sub packages</u> Visibility of the attributes (choose all used): <u>○ ~ # ⊕</u>	Prof notes:
-1/20	Object class / collections used incorrectly	Which classes override equals? <u>Card</u> Did you use any sorting method or sorted collection from Java? <u>Yes</u> How is the deck shuffled? <u>Collection's default shuffle method</u> Did you provide your own exceptions? <u>Yes</u>	Prof notes:
-1/20	Incorrect data structures	Collections used (apart from arrays): <u>HashMap, LinkedHashMap, ArrayList</u>	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		