

Environment	(max)	Points	
Integrated development environment (IDE) can be used in a professional manner (short-cuts, importing APIs, etc.)	1 P	1	✓
Uses a CASE-Tool for UML (stand-alone, plugin, etc.)	1 P	2	
Uses a version control tool (GitHub, GitLab)	1 P	3	✓
Implementation	2 P		
Your own project has been implemented...	+ 2 P		✓
with use cases (written)	+ 2 P		
with a sequence diagram	+ 1 P		✓
with class diagrams	+ 2 P	12	
(a rough design at beginning, detailed at end of project)			
Code is runnable	1 P		
should run, but doesn't	-1P		
it runs for certain use cases, with others it crashes	+ 1 P	14	
runs fully			
Exception-Handling is being used	2 P		
own Exceptions implemented	+ 2 P	18	
Input and output via console	1 P		
using files / data source	+ 2 P		
very user friendly	+ 2 P	23	
JavaDoc is used	2 P	25	
for min. 80% of classes and min. 80% of methods			
JUnit is applied	1 P		
for min. 1 class with min. 2 test methods	+ 2 P	28	
for a higher test coverage			
Object-orientated concepts in code	1 P	29	✓
OO-concepts are shown with own written code			
Encapsulation / Information Hiding is applied	1 P		✓
Delegation is being used	1 P		✓
Aggregation / Composition is implemented	1 P		✓
Classes are well structured (ie. what they need to do with clearly defined methods)	+ 1 P	33	
Clean code (ie. main-method has no logic), distinction between input and logic (and data)	+ 2 P	35	
Project is briefly presented in class / or to teacher	+ 2 P	37	
Documentation covering design, planning and test cases (use cases) is in a repository (progress is documented), format in markdown	4 P	41	

Maximum points: 41; for grade 6 you need min.39pts.; for grade 5: min.31pts; for grade 4: min.24pts; for grade 3 min.16pts. If you have below 23pts the third column in the c-grid remains empty.