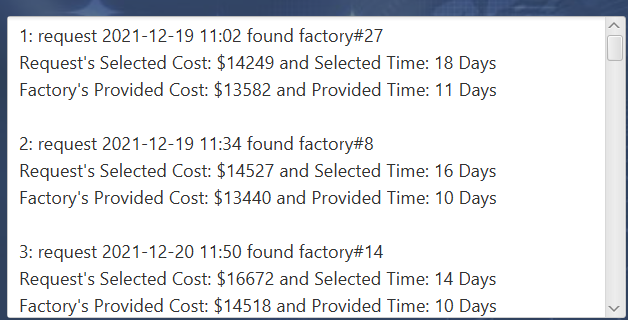
**Correct (necessary)**

Output.1 (Simulation life in steps(min/hours/days))



Output.2 (All changes in factors that affect results are easy to follow)

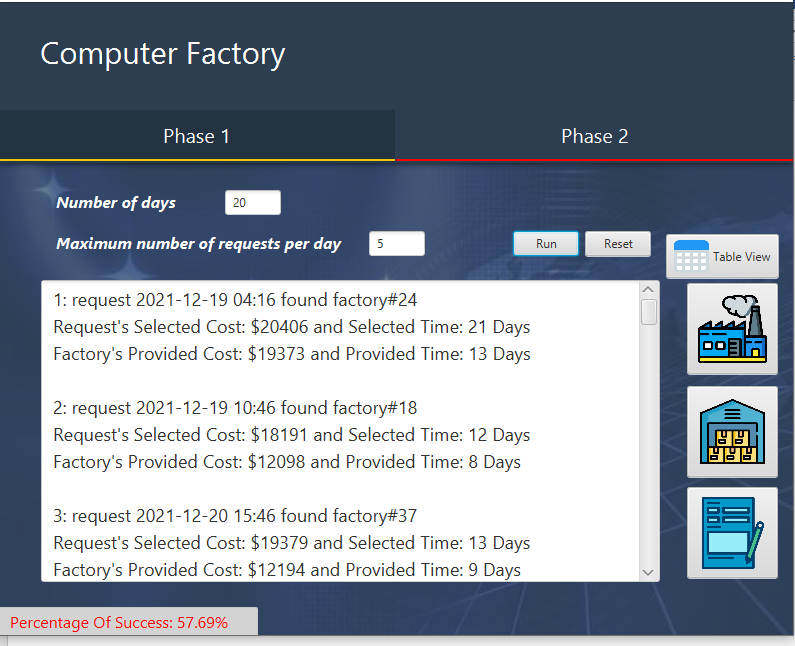
Output.3 (Phase#1 output: the problem definition is easy to identify)

Output.4 (Phase#2 output: the problem solution is effective)

Output.5 (The analyzed data is well thought and help decision maker to notice the problem)

Output.6 (A set of options provided after some steps in the simulation life)

Output. 7 ( Improvement percentage after applying the solution is present in pahse#2 output





Output. 8 (The results in the output are getting worse or better (based on the running phase of the project) in every step of the simulation.

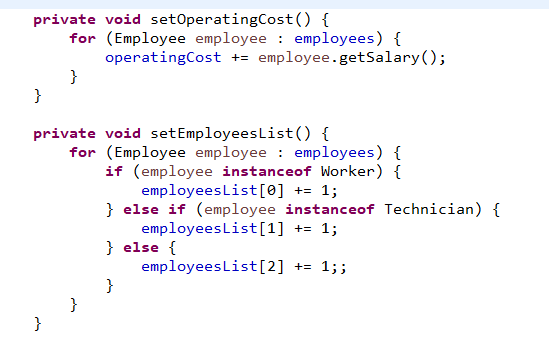
Output. 9(JavaFx Graphical user interface (GUI)



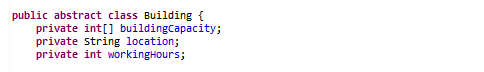
**Classes and methods**

Classes and methods.1 (Attributes are related to class type)

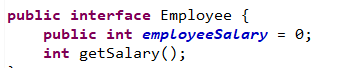
Classes and methods.2 (Polymorphism used to make coding easier)

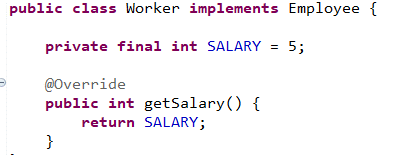


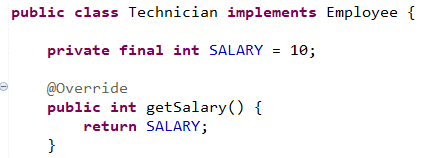
Classes and methods.3 (Abstract used to make coding easier)

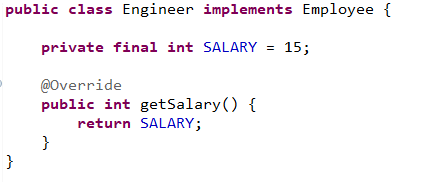


Classes and methods.4 (Interface used to make coding easier)









Classes and methods.5 (All methods have one task to do and the name of each method is related to the task it is performin)

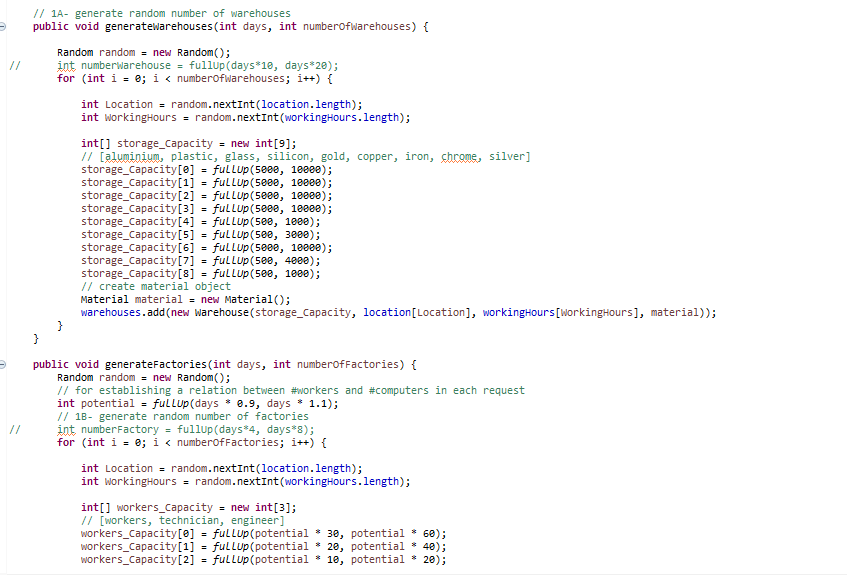
|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

Classes and methods.6 (Arrays type are well chosen and helps to use polymorphism, abstract and interface. )

Classes and methods.7 (Effective use of inner classes to help make your code easier to read)

**Main loop**

Main loop.1 (The loop has steps to follow)



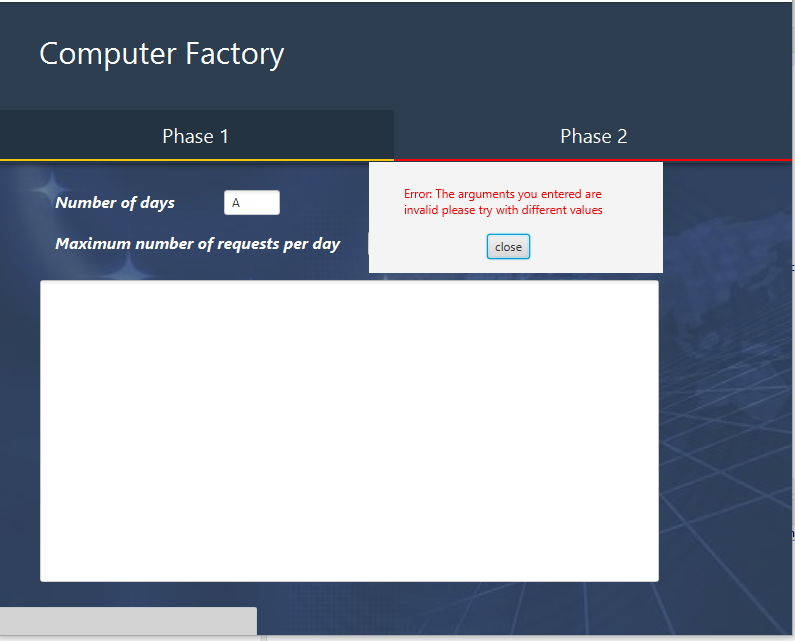
Main loop.2 ( Tasks in the main loop are sorted in the right order)

Main loop.3 ( The main loop stops to apply changes according to user preference or to show more details (ex. history of each object) )

Main loop.4 (Methods are used to minimize the code in the main loop (methods used as bullet points))

**Coding**

Coding .1 (No runtime exceptions)

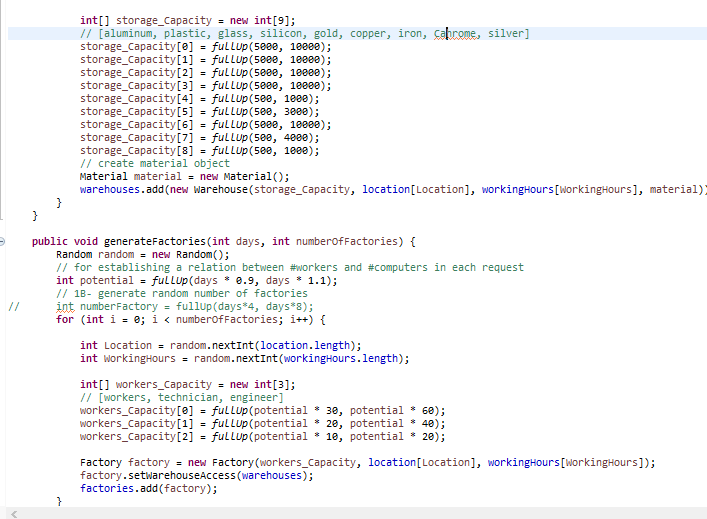
Coding.2(Clean code)

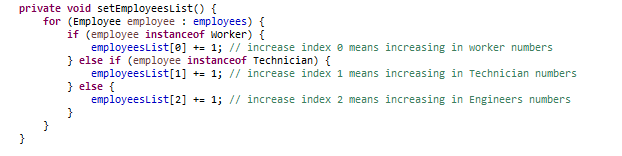
Coding.3 ( All variables have meaningful names related to the values it is holding)

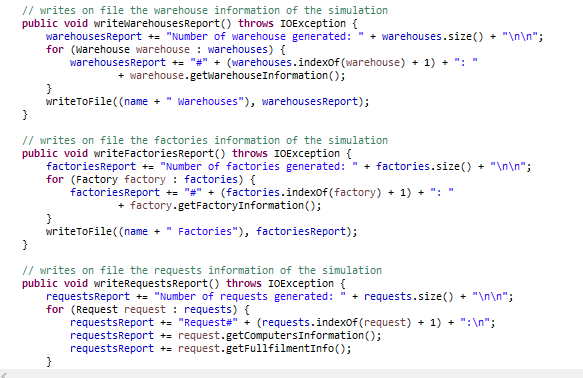
|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

Coding.4 (The code is divided to models (methods)

Coding.5 (Every line in the code is commented with either the goal, task, or necessity of written code.







Coding.6 (The code is published in Javadoc form. )