Java Programming  
  
Report #1: Object-oriented programming  
Geometry Visualizer

**Class : 18CLC2-KTPM**

|  |  |
| --- | --- |
| **Your group**: | **Nguyễn Quý Thanh – 18127210**  **Nguyễn Nhật Thảo – 18127220**  **Nguyễn Lê Hoàng Nam – 18127160** |

Table of content

[Revision History 3](#_Toc54770434)

[Introduction 4](#_Toc54770435)

[Analysis and design 5](#_Toc54770436)

[Implementation 6](#_Toc54770437)

[Result 7](#_Toc54770438)

[Plan 8](#_Toc54770439)

[References 9](#_Toc54770440)

# Revision History

[*Provide in this section a revision history table. A such sample table is given below*]

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 15/10/2000 | 0.1 | Very early implementation with barebones graphics, no feature yet | Thanh, Nam, Thao |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Introduction

* *Geometry has always been a hard subject for students in secondary schools, not because it is hard on logic but it requires a rich imagination for students to be able to solve these problems.*
* *That is why we choose this topic as our application, we want to help students to sketch and interact on the geometry model in which they’re working on.*
* *Currently there are lots of application to sketch a model such as Sketch, Adobe Illustrator, Gimp, … but what these applications have in common is they’re very complex and not mainly aim for geometry visualization.*
* *The application must be simple but powerful, easy to maintain and adding support for 3D visualization in the future. User interface should be clean, simple, consistent. Also include cloud database to store accounts and models online.*

# Analysis and design

*[ Provide the class diagram to show the organization of your code to be implemented. (If possible, present a general diagram to better show the class hierarchy and then the detail of each class (with main attributes and operations). You are encouraged to draw UML class diagram with Visual Paradigm).*

*Give the package diagram to show the decomposition of your code into packages (if any). Also give a brief description for each package.*

*Give the explanation to describe each figure or each class and the reason for your program's structural design.*

*Present and give explanation for all the design patterns, algorithms you use in the project.*

*]*

# Result

*[Explain what you have achieved until now, advantages, disadvantages and planned solutions (if possible)]*

# Plan

*[Give your project plan (in detail) until the end of the project: task decomposition, ressources allocation, duration of each task, etc.]*

# References

*[Provide all the resources to use in your project, including existing codes, algorithms used, books, reports, links to consult, etc. ]*