Assignment 4 - 5 Project

Soham Prabhakar Patil
Rhishabh Suhas Hattarki
Anmol Girish More
Sanket Surendra Kapse

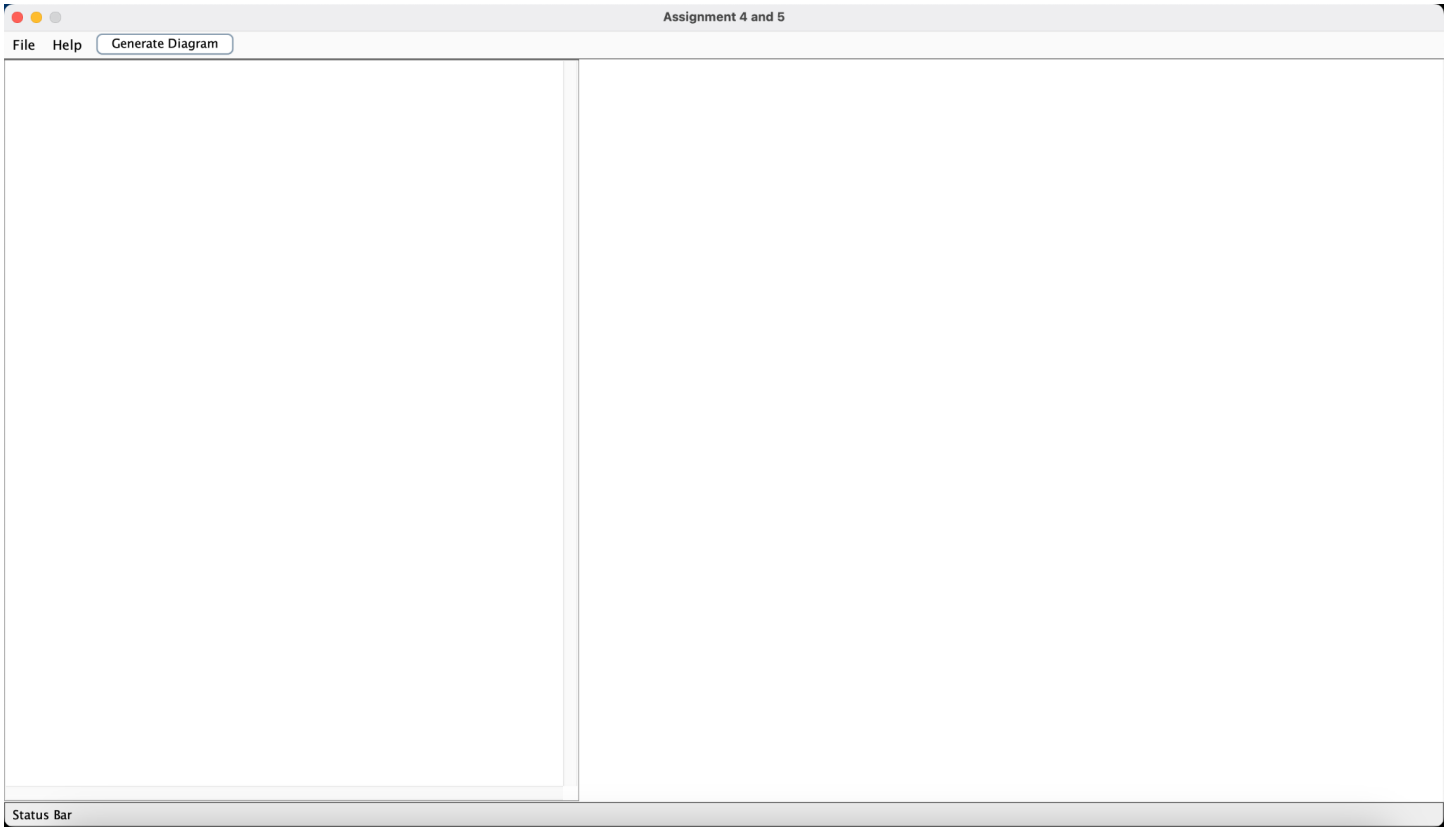
Class Diagram:

Attached in zip folder - Asta as well as png file.

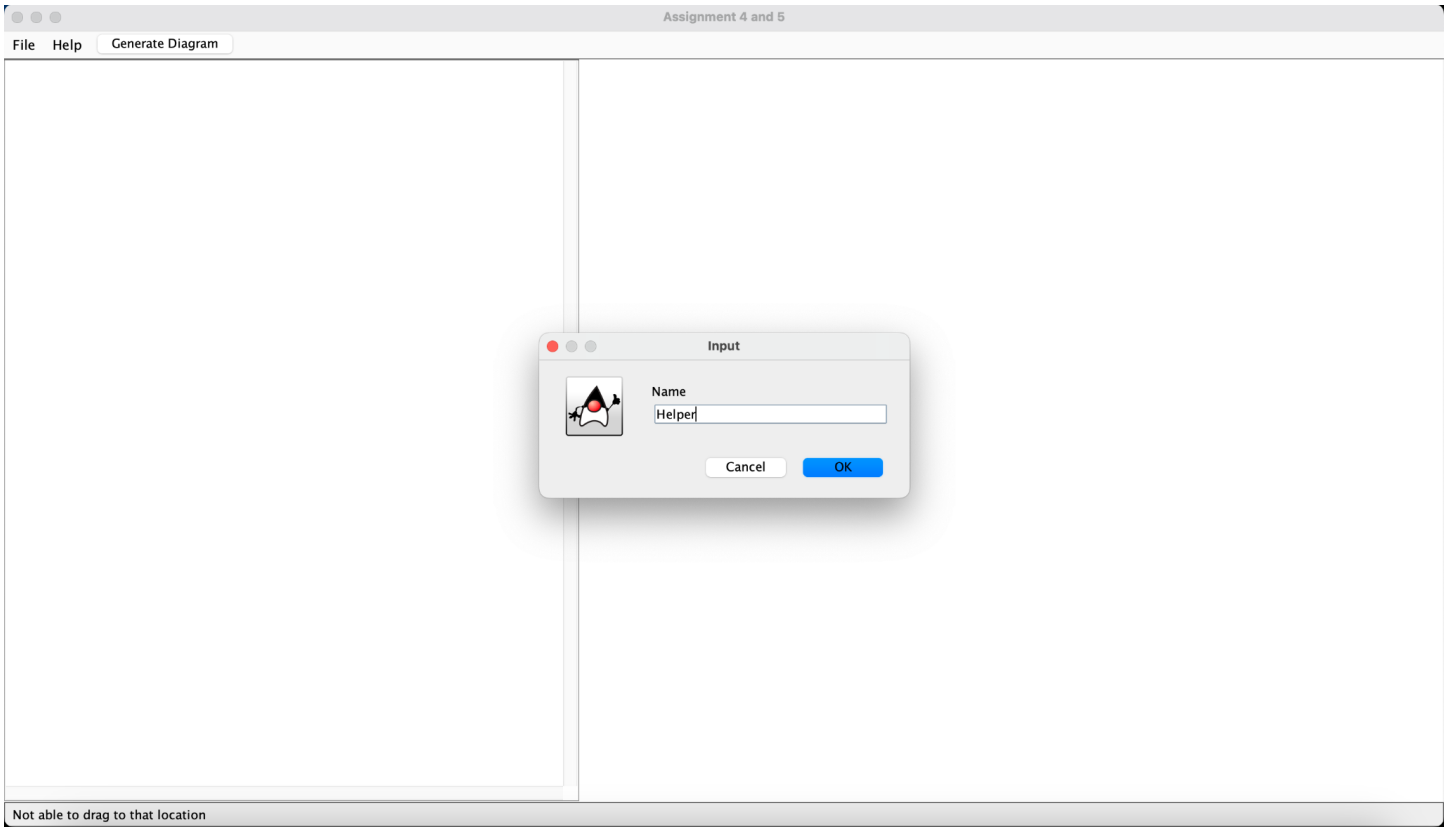
Class: Main -> Starting point of the application.

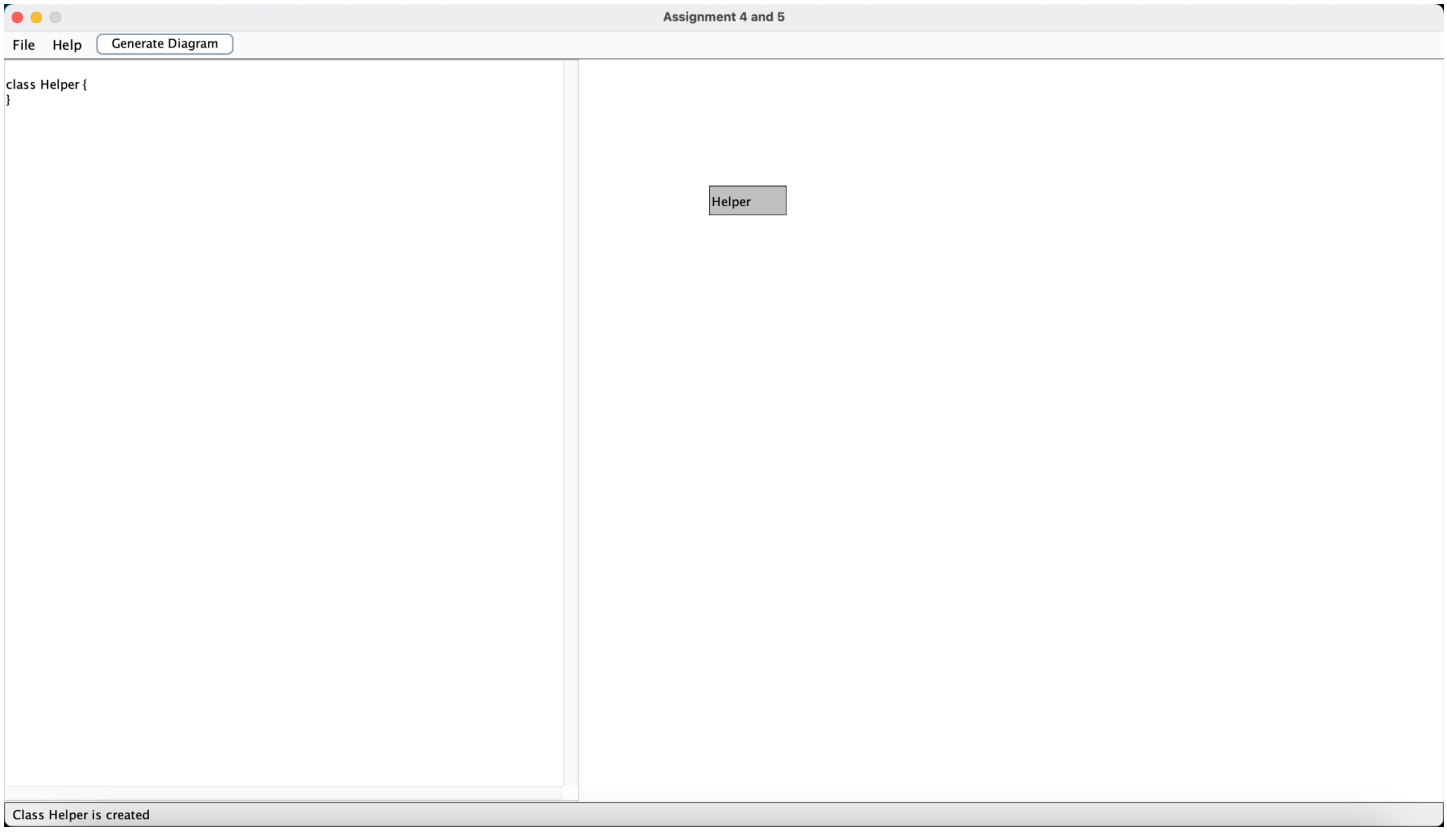
Functionality implementation

1. **Screen with code panel, drawing panel and menus.**

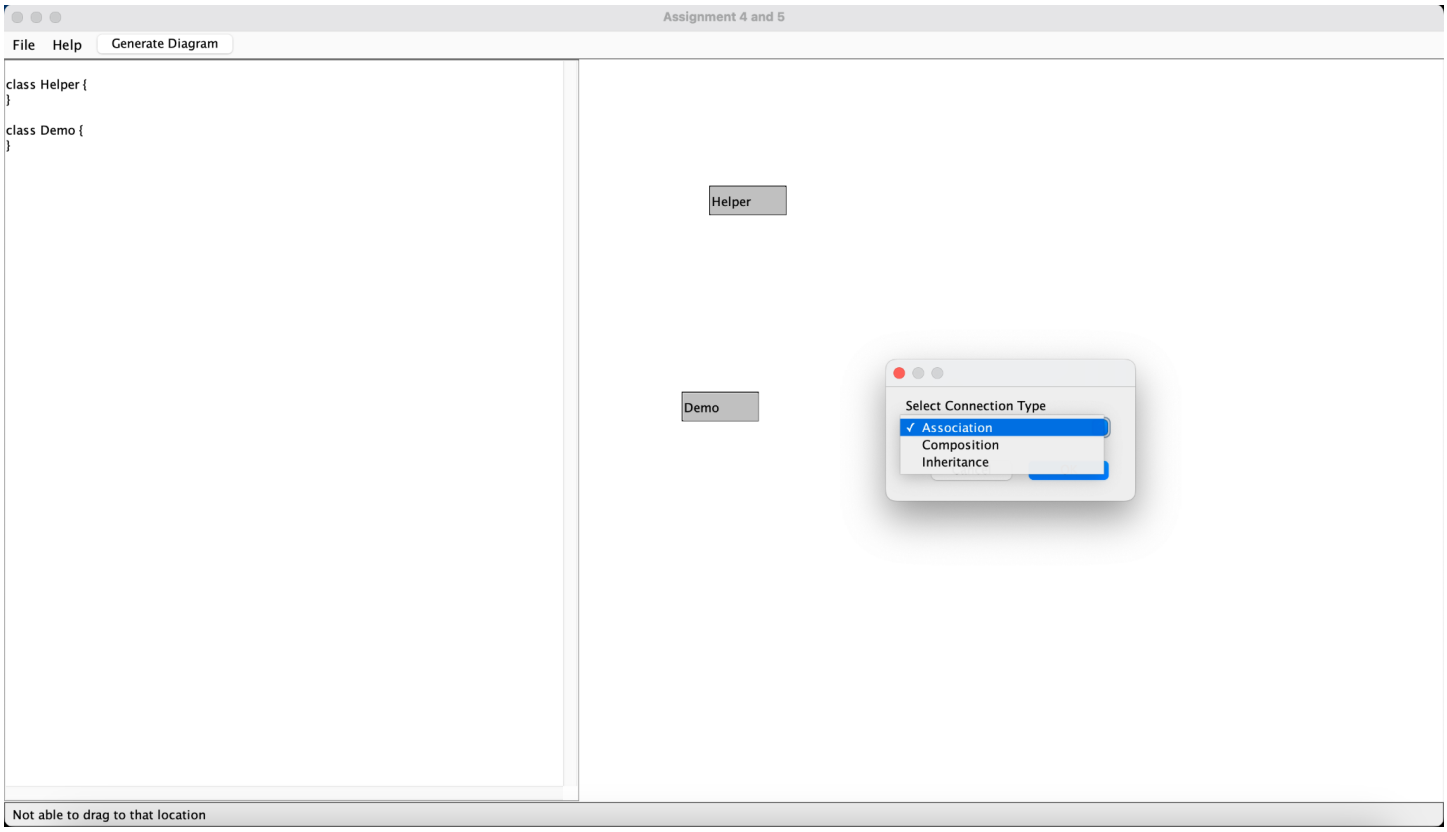


2. **Create a class box once click on canvas and create code related to it in the code panel.**

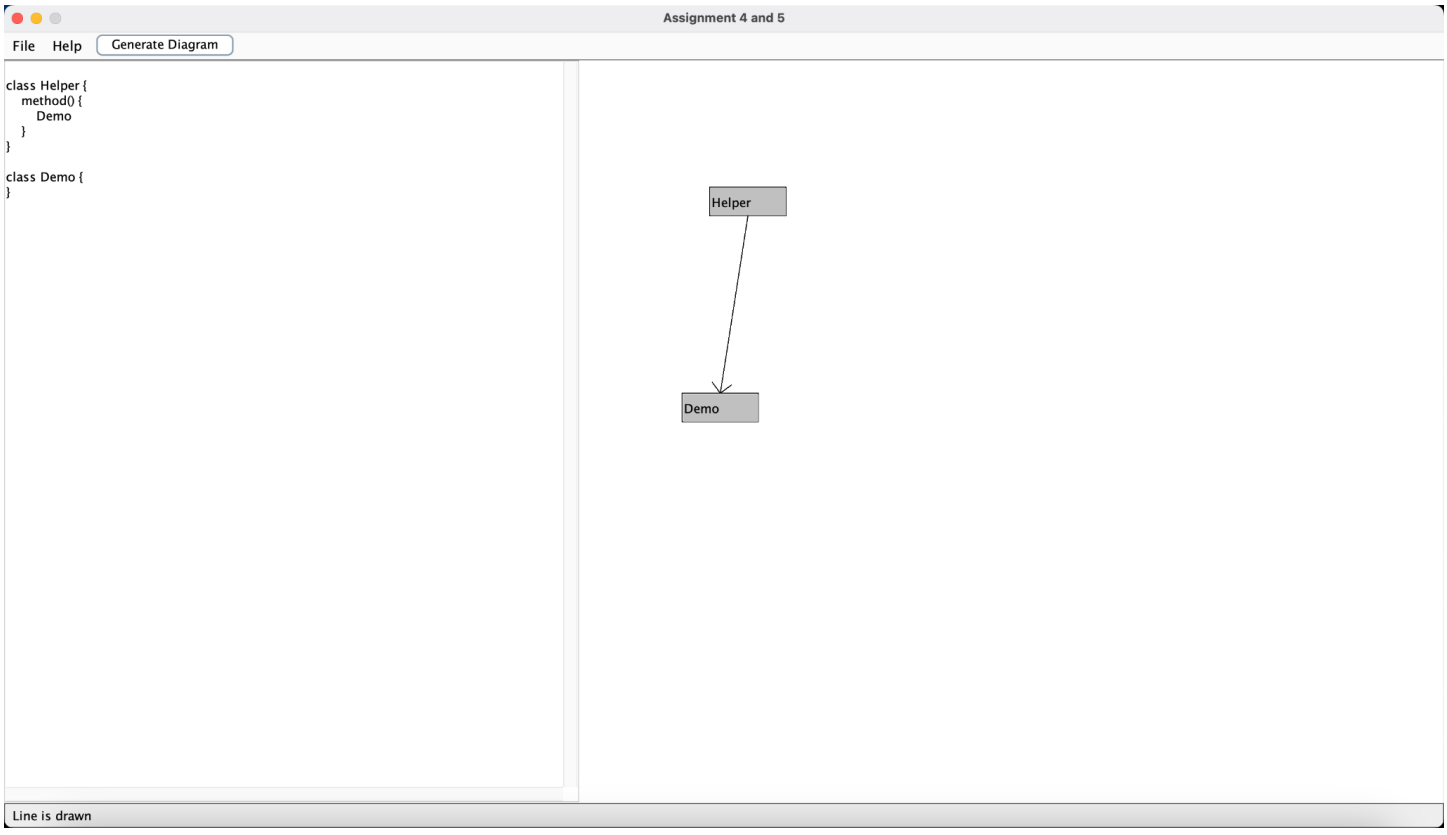




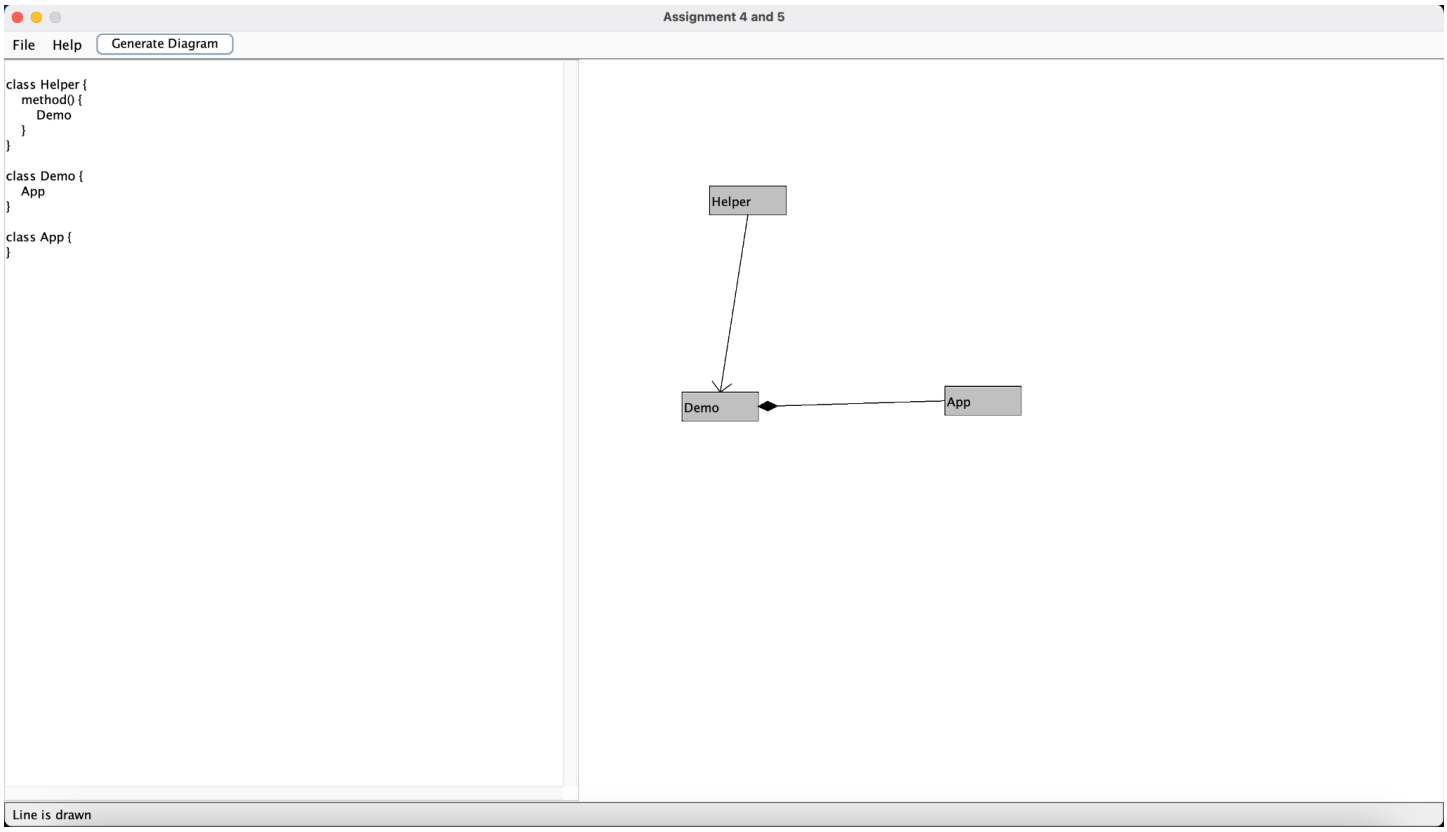
3. **Once 2 classes are selected, create a connection and select relation type.**



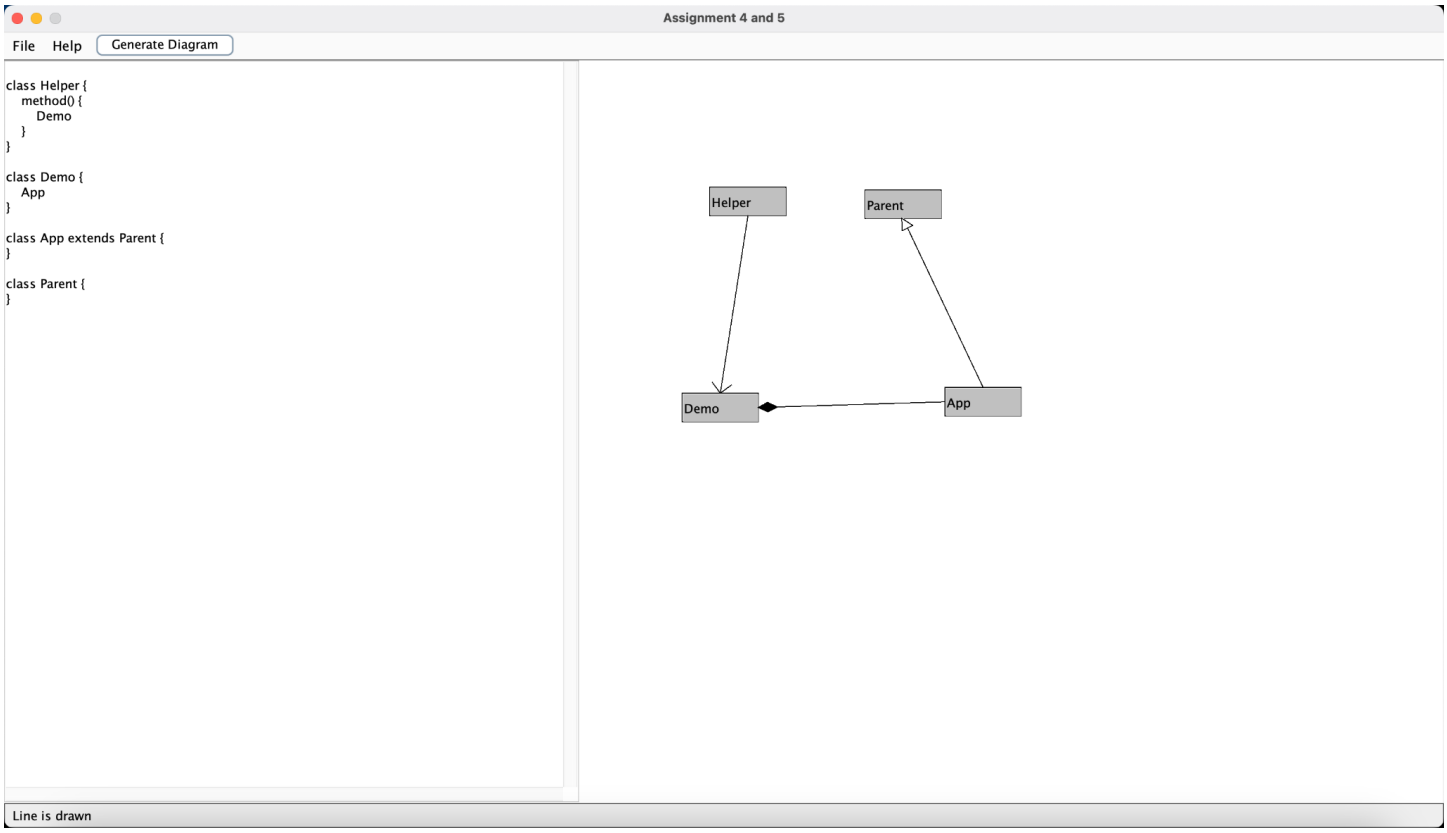
a. Association



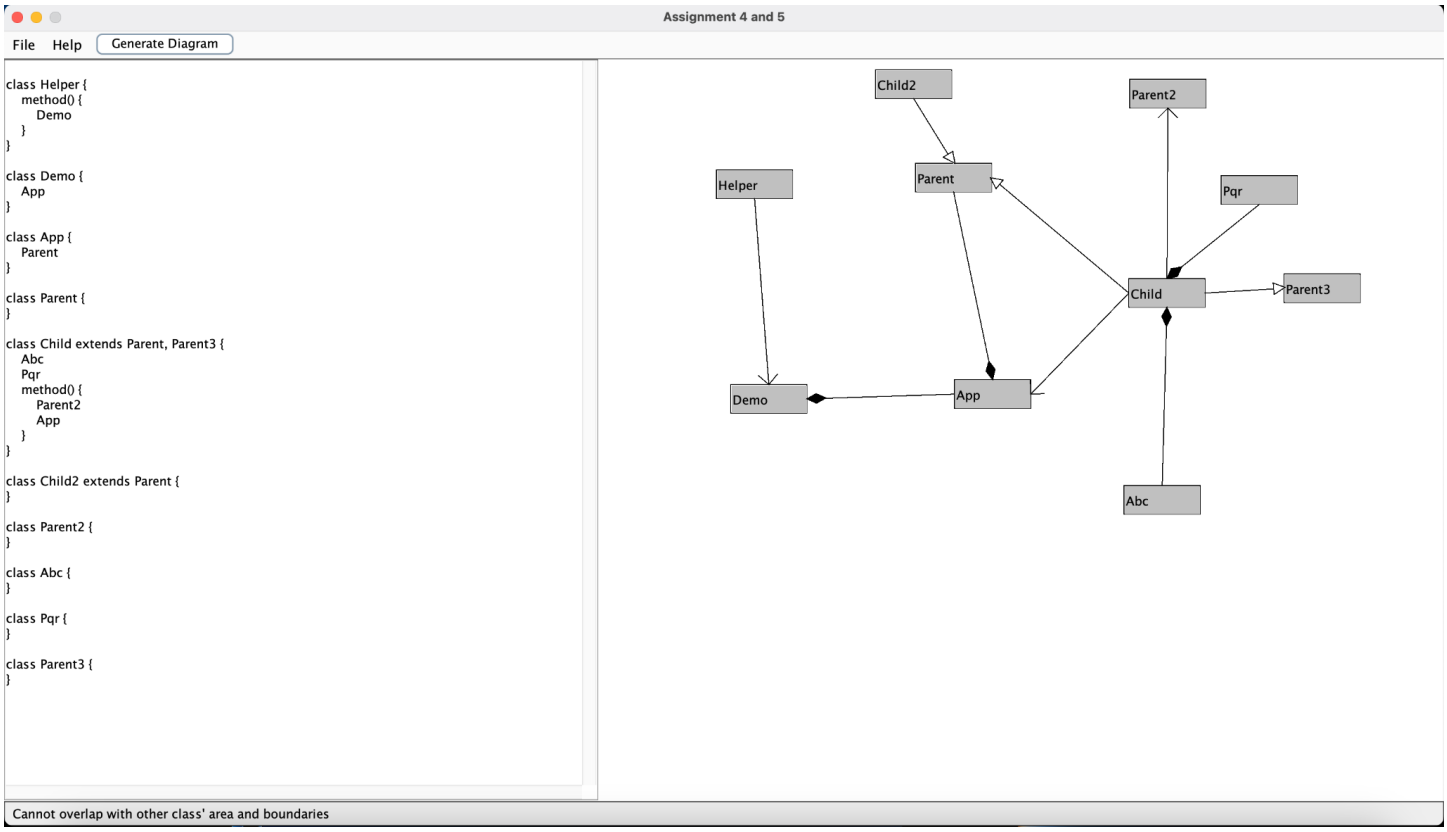
b. Composition



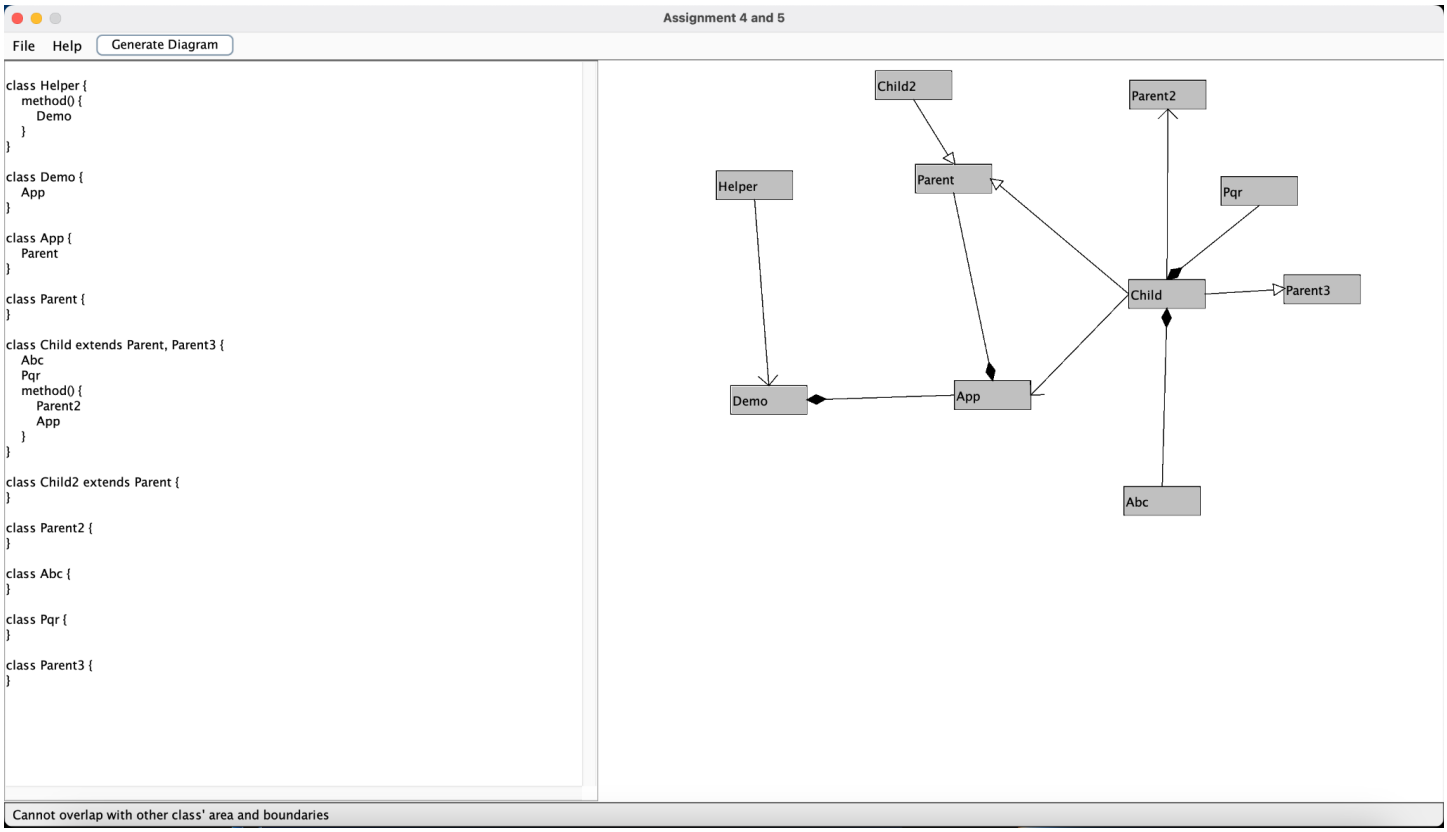
c. Inheritance

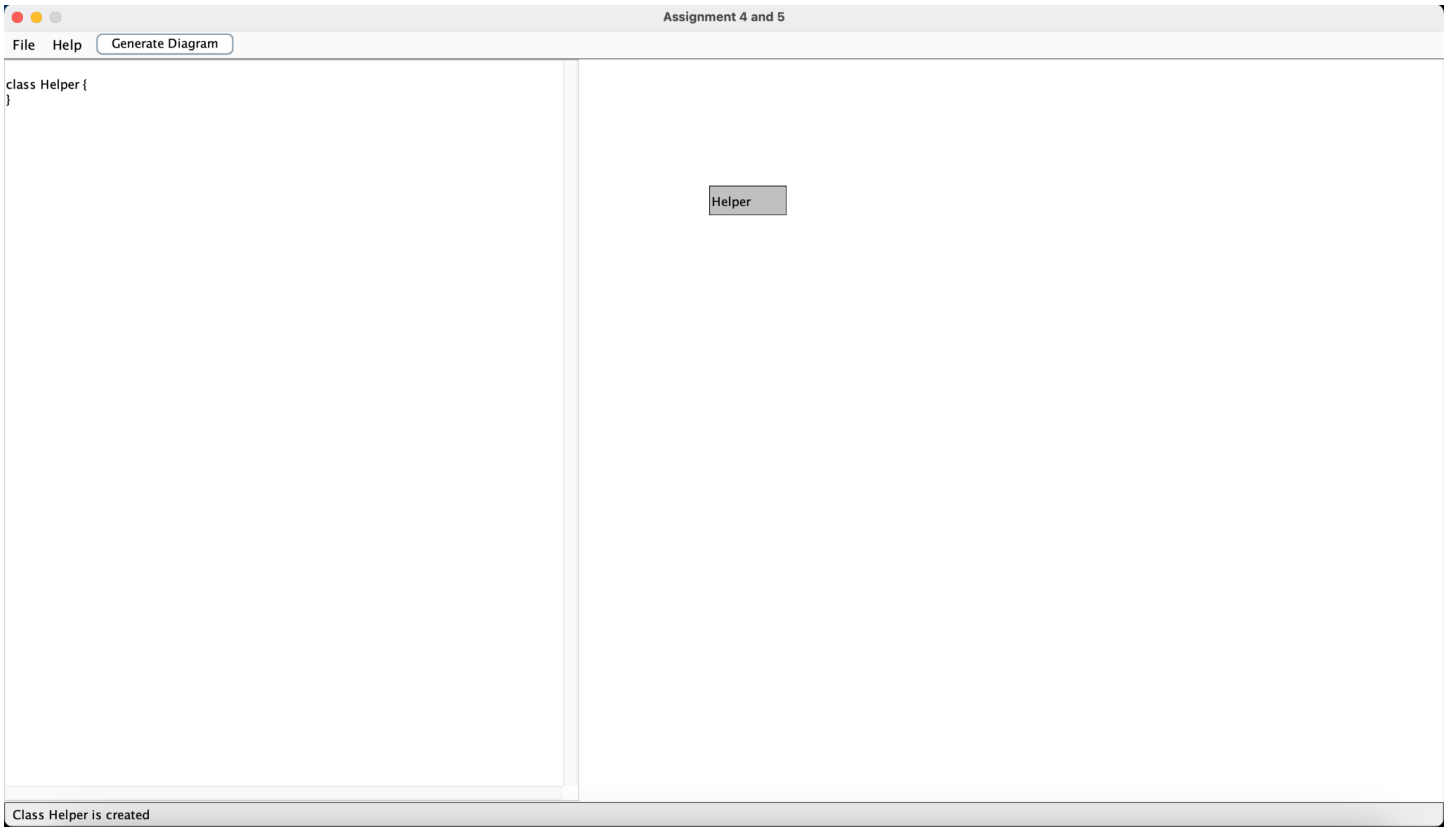


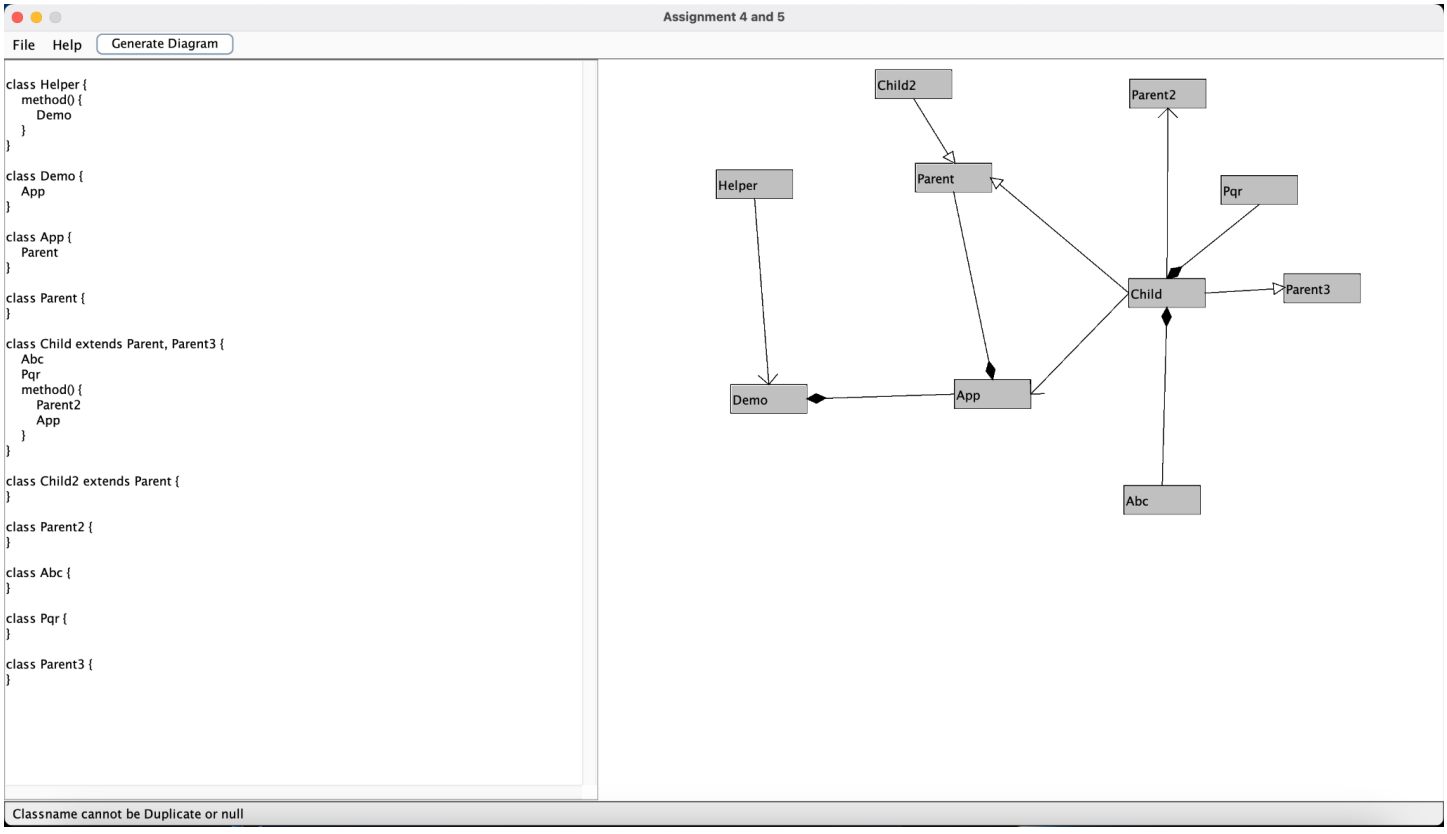
d. Aggregated view


Assignment 4 and 5
File
Help
Generate Diagram
class Helper {
method() {
Demo
}
}
class Demo {
App
}
class App {
Parent
}
class Parent {
}
class Child extends Parent, Parent3 {
Abc
Pqr
method() {
Parent2
App
}
}
class Child2 extends Parent {
}
class Parent2 {
}
class Abc {
}
class Pqr {
}
class Parent3 {
}
Child2
Parent2
Helper
Parent
Pqr
Child
Parent3
Demo
App
Abc
Cannot overlap with other class' area and boundaries

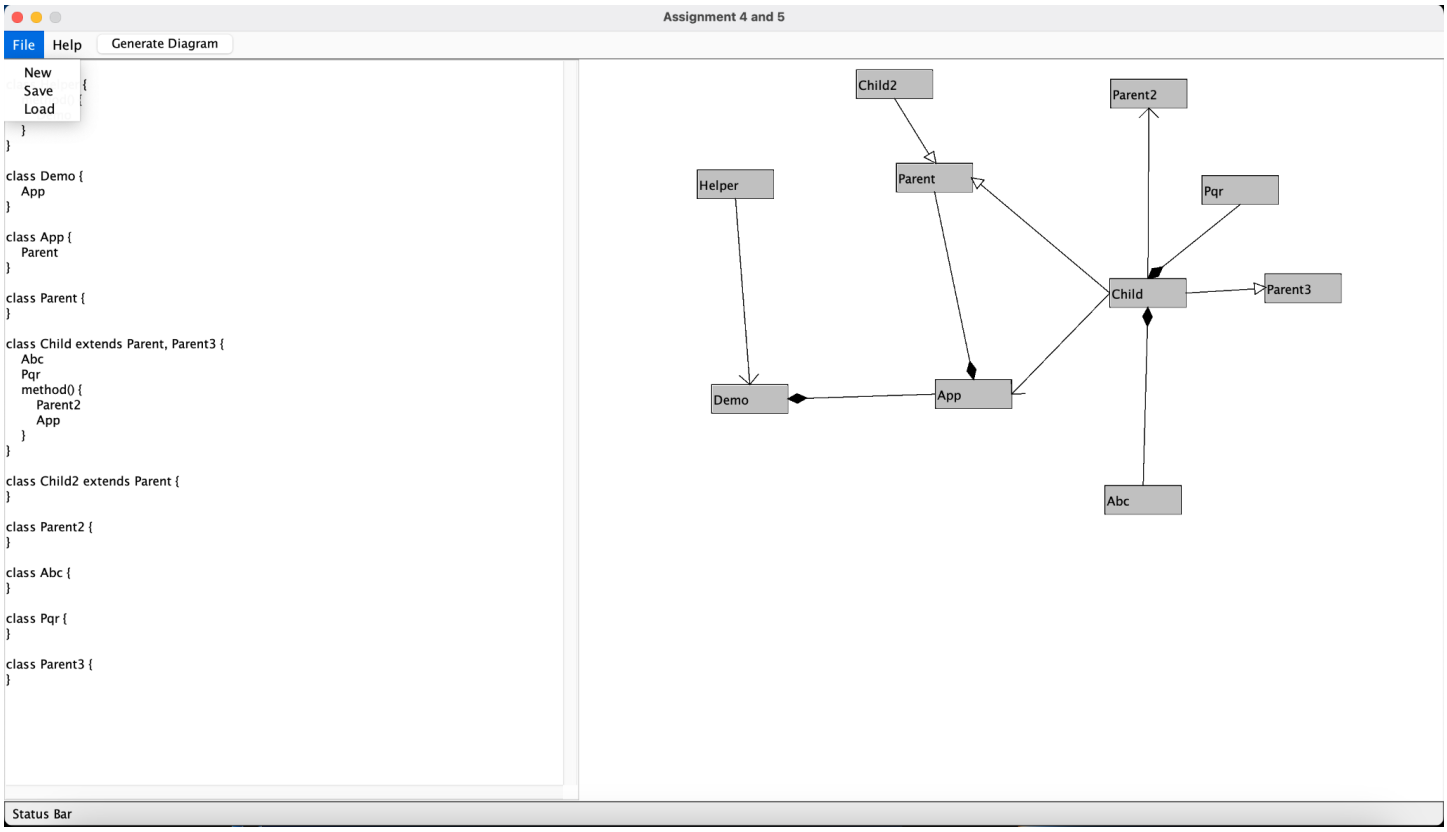
4. Status Bar


Assignment 4 and 5
File
Help
Generate Diagram
class Helper {
method() {
Demo
}
}
class Demo {
App
}
class App {
Parent
}
class Parent {
}
class Child extends Parent, Parent3 {
Abc
Pqr
method() {
Parent2
App
}
}
class Child2 extends Parent {
}
class Parent2 {
}
class Abc {
}
class Pqr {
}
class Parent3 {
}
Child2
Parent2
Helper
Parent
Pqr
Child
Parent3
Demo
App
Abc
Cannot overlap with other class' area and boundaries

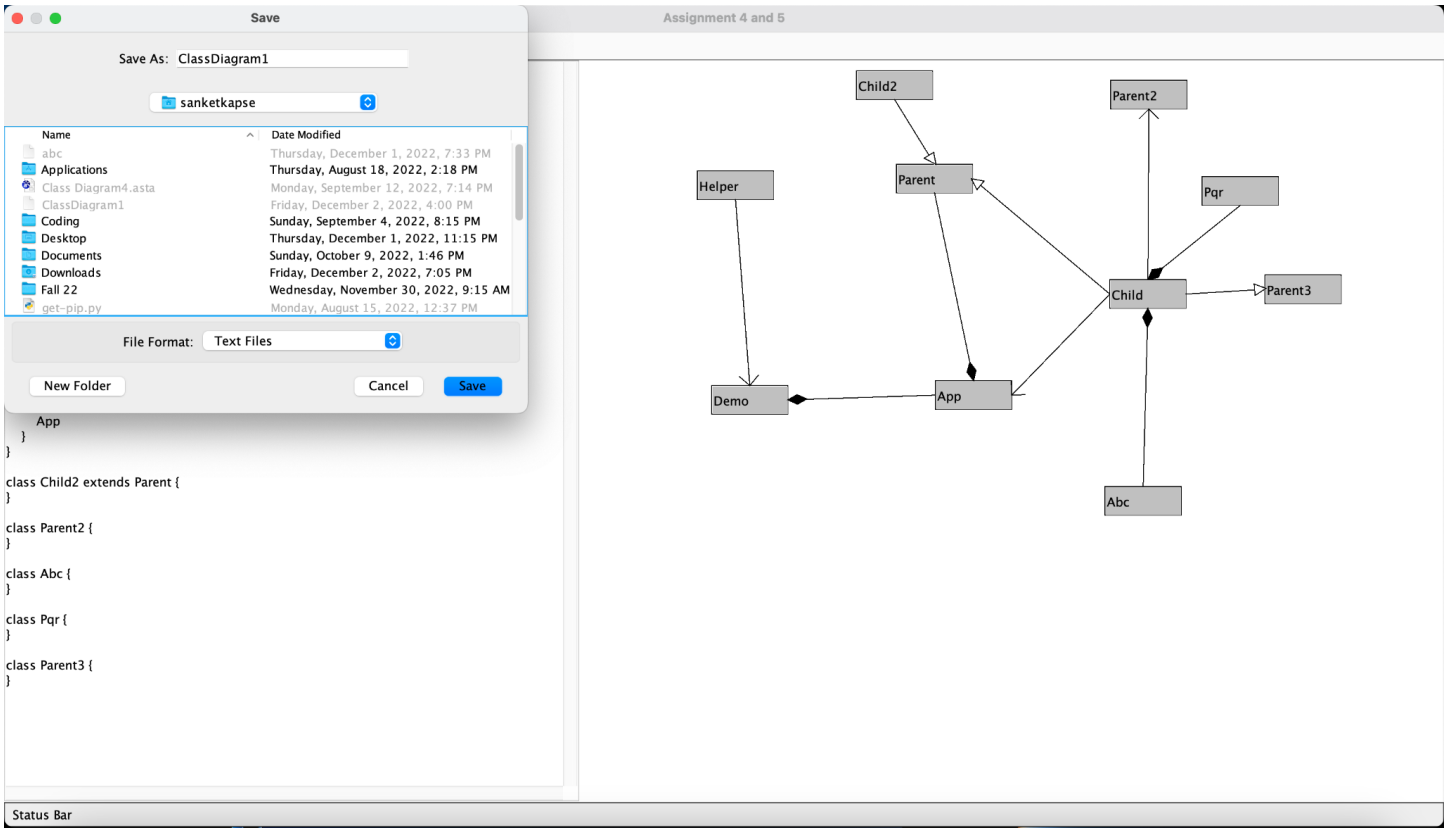

Assignment 4 and 5
File
Help
Generate Diagram
class Helper {
}
Helper
Class Helper is created


Assignment 4 and 5
File
Help
Generate Diagram
class Helper {
method() {
Demo
}
}
class Demo {
App
}
class App {
Parent
}
class Parent {
}
class Child extends Parent, Parent3 {
Abc
Pqr
method() {
Parent2
App
}
}
class Child2 extends Parent {
}
class Parent2 {
}
class Abc {
}
class Pqr {
}
class Parent3 {
}
Child2
Parent2
Helper
Parent
Pqr
Child
Parent3
Demo
App
Abc
Classname cannot be Duplicate or null

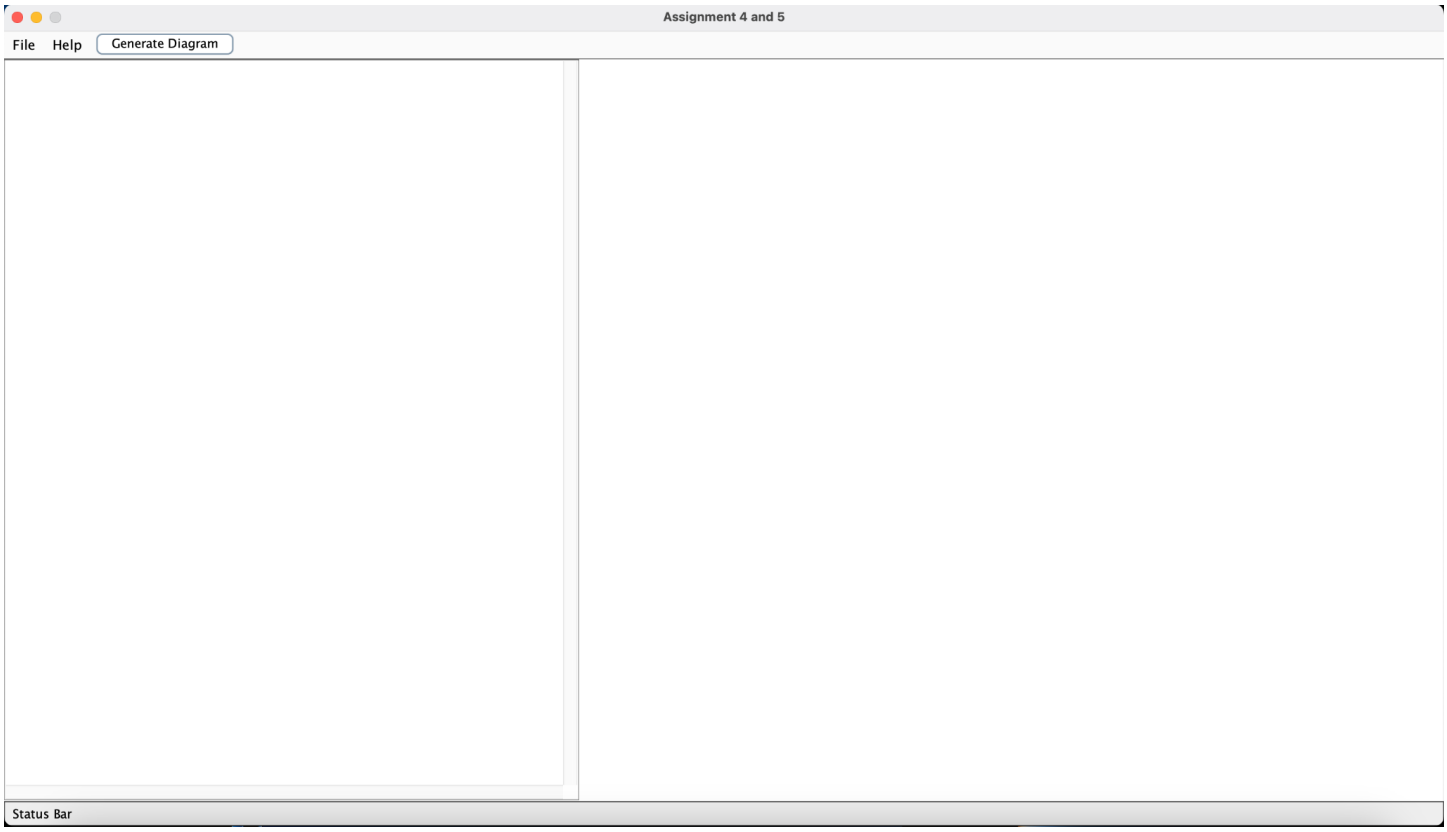
5. File Menu


Assignment 4 and 5
File
Help
Generate Diagram
New
Save
Load
}
}
class Demo {
App
}
class App {
Parent
}
class Parent {
}
class Child extends Parent, Parent3 {
Abc
Pqr
method() {
Parent2
App
}
}
class Child2 extends Parent {
}
class Parent2 {
}
class Abc {
}
class Pqr {
}
class Parent3 {
}
Child2
Parent2
Helper
Parent
Pqr
Child
Parent3
Demo
App
Abc
Status Bar

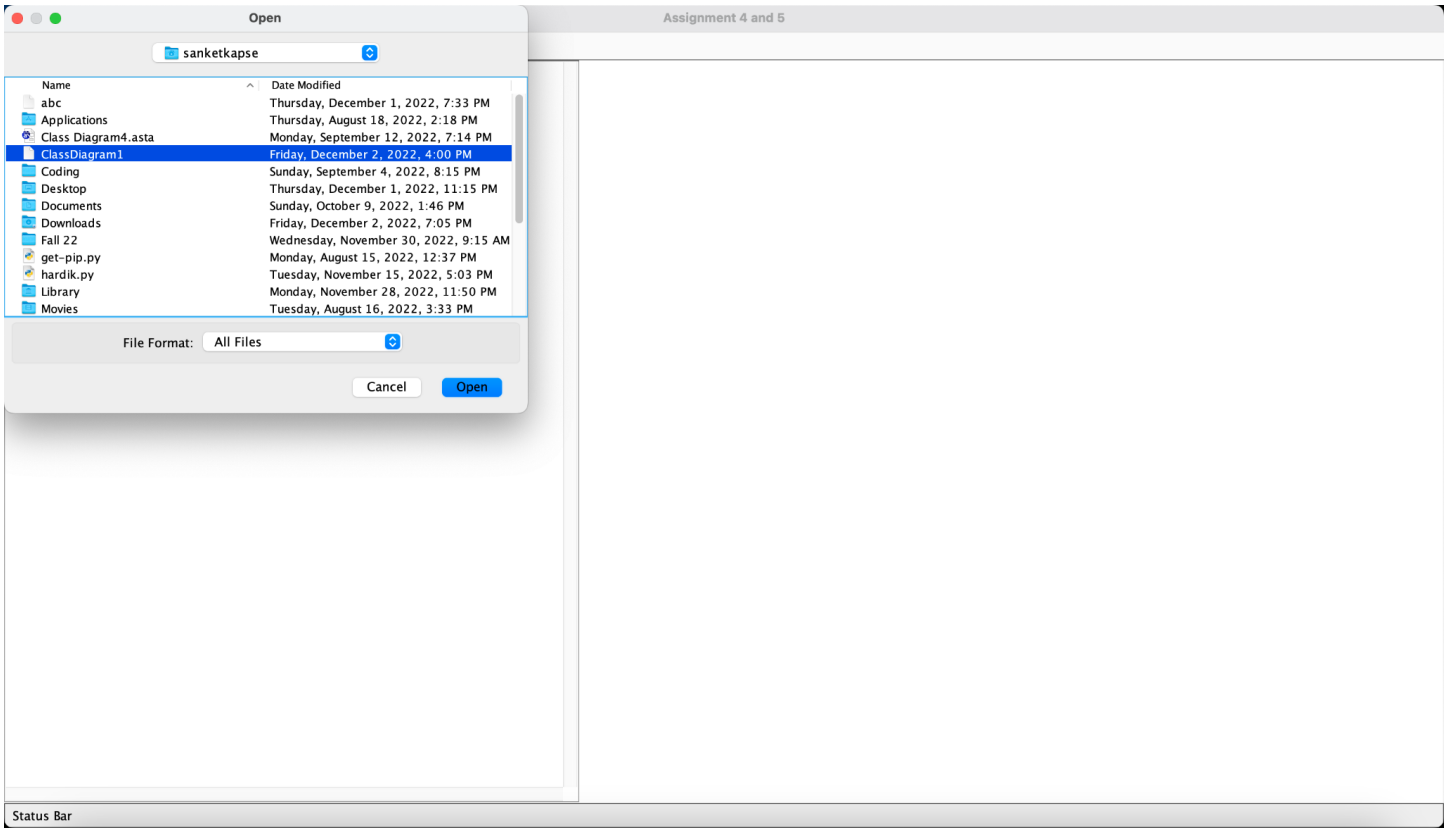
a. Save

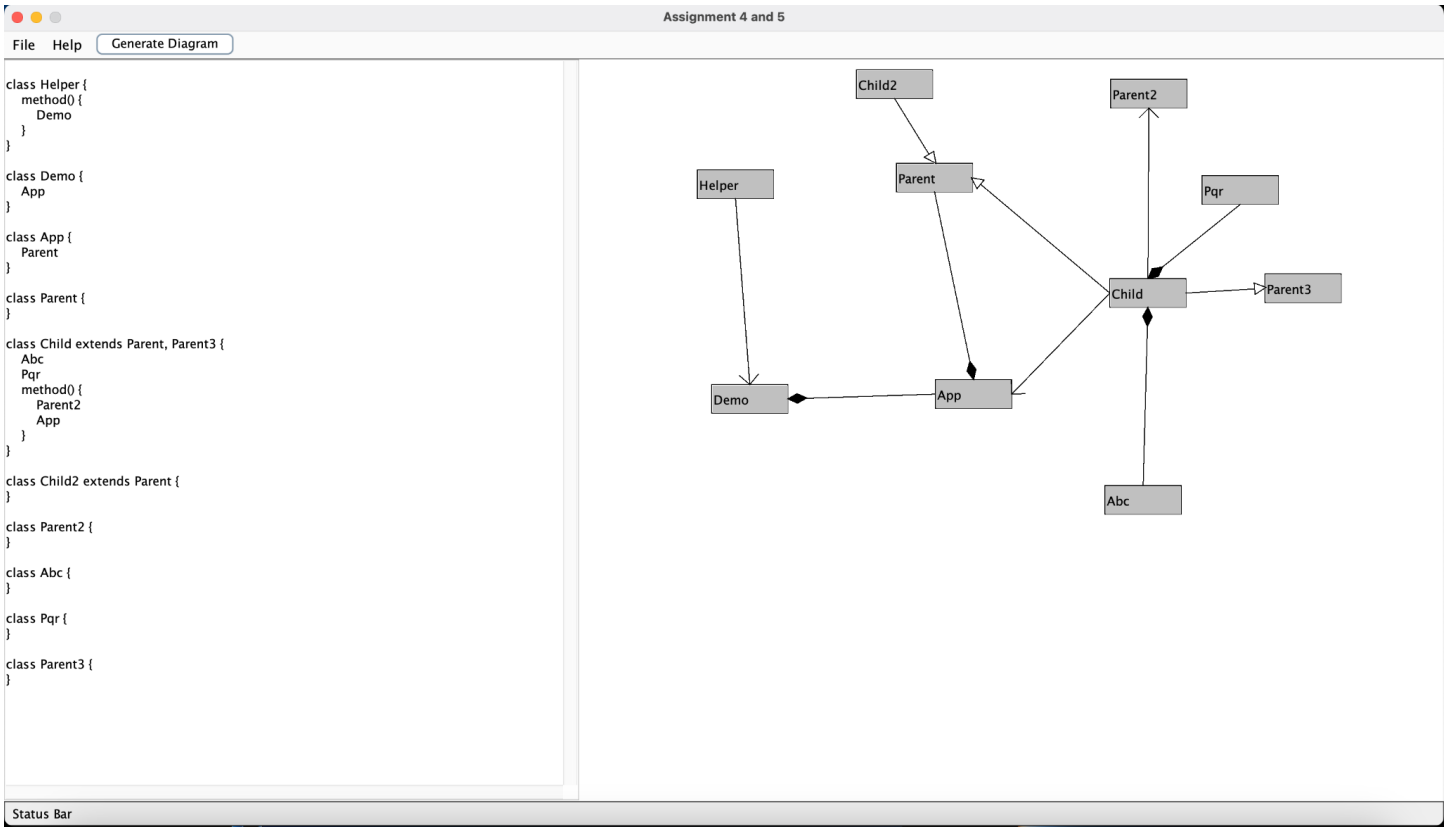

Save
Save As: ClassDiagram1
sanketkapse
Name
Date Modified
abc
Thursday, December 1, 2022, 7:33 PM
Applications
Thursday, August 18, 2022, 2:18 PM
Class Diagram4.asta
Monday, September 12, 2022, 7:14 PM
ClassDiagram1
Friday, December 2, 2022, 4:00 PM
Coding
Sunday, September 4, 2022, 8:15 PM
Desktop
Thursday, December 1, 2022, 11:15 PM
Documents
Sunday, October 9, 2022, 1:46 PM
Downloads
Friday, December 2, 2022, 7:05 PM
Fall 22
Wednesday, November 30, 2022, 9:15 AM
get-pip.py
Monday, August 15, 2022, 12:37 PM
File Format: Text Files
New Folder
Cancel
Save
Assignment 4 and 5
App
}
}
class Child2 extends Parent {
}
class Parent2 {
}
class Abc {
}
class Pqr {
}
class Parent3 {
}
Child2
Parent2
Helper
Parent
Pqr
Child
Parent3
Demo
App
Abc
Status Bar

b. New

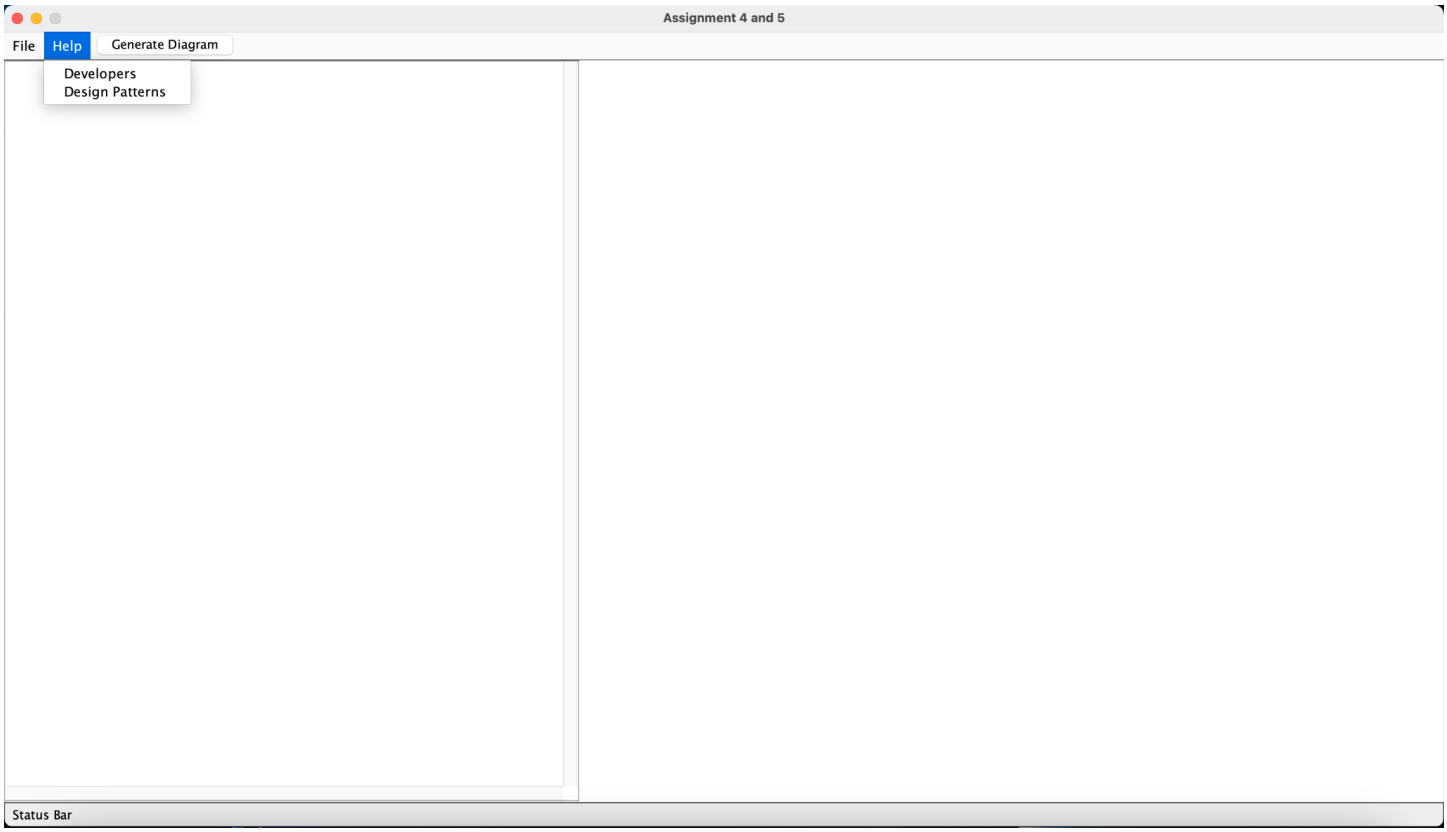

Assignment 4 and 5
File
Help
Generate Diagram
Status Bar

c. Load

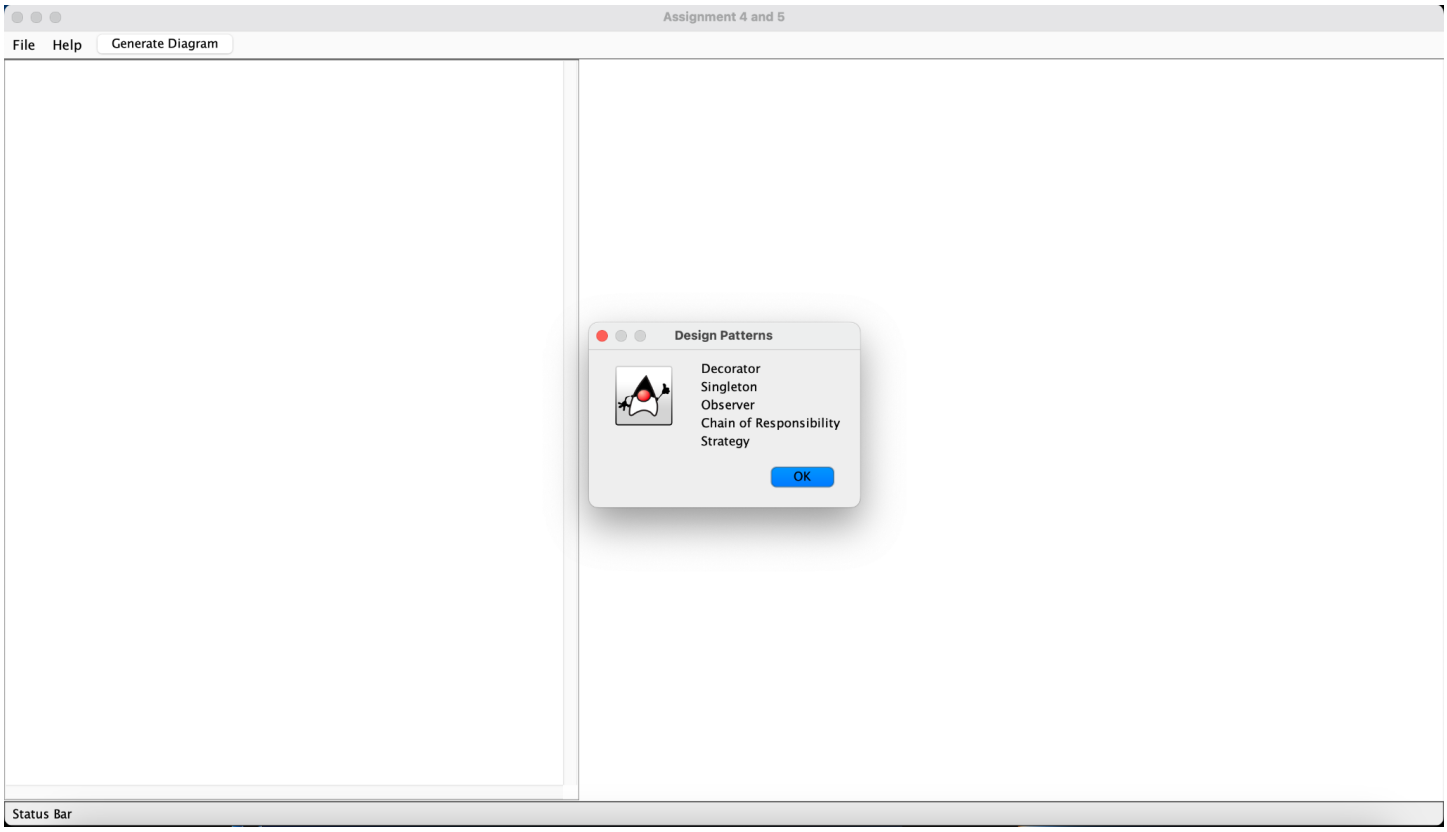

Open
Assignment 4 and 5
sanketkapse
Name
Date Modified
abc
Thursday, December 1, 2022, 7:33 PM
Applications
Thursday, August 18, 2022, 2:18 PM
Class Diagram4.asta
Monday, September 12, 2022, 7:14 PM
ClassDiagram1
Friday, December 2, 2022, 4:00 PM
Coding
Sunday, September 4, 2022, 8:15 PM
Desktop
Thursday, December 1, 2022, 11:15 PM
Documents
Sunday, October 9, 2022, 1:46 PM
Downloads
Friday, December 2, 2022, 7:05 PM
Fall 22
Wednesday, November 30, 2022, 9:15 AM
get-pip.py
Monday, August 15, 2022, 12:37 PM
hardik.py
Tuesday, November 15, 2022, 5:03 PM
Library
Monday, November 28, 2022, 11:50 PM
Movies
Tuesday, August 16, 2022, 3:33 PM
File Format:
All Files
Cancel
Open
Status Bar



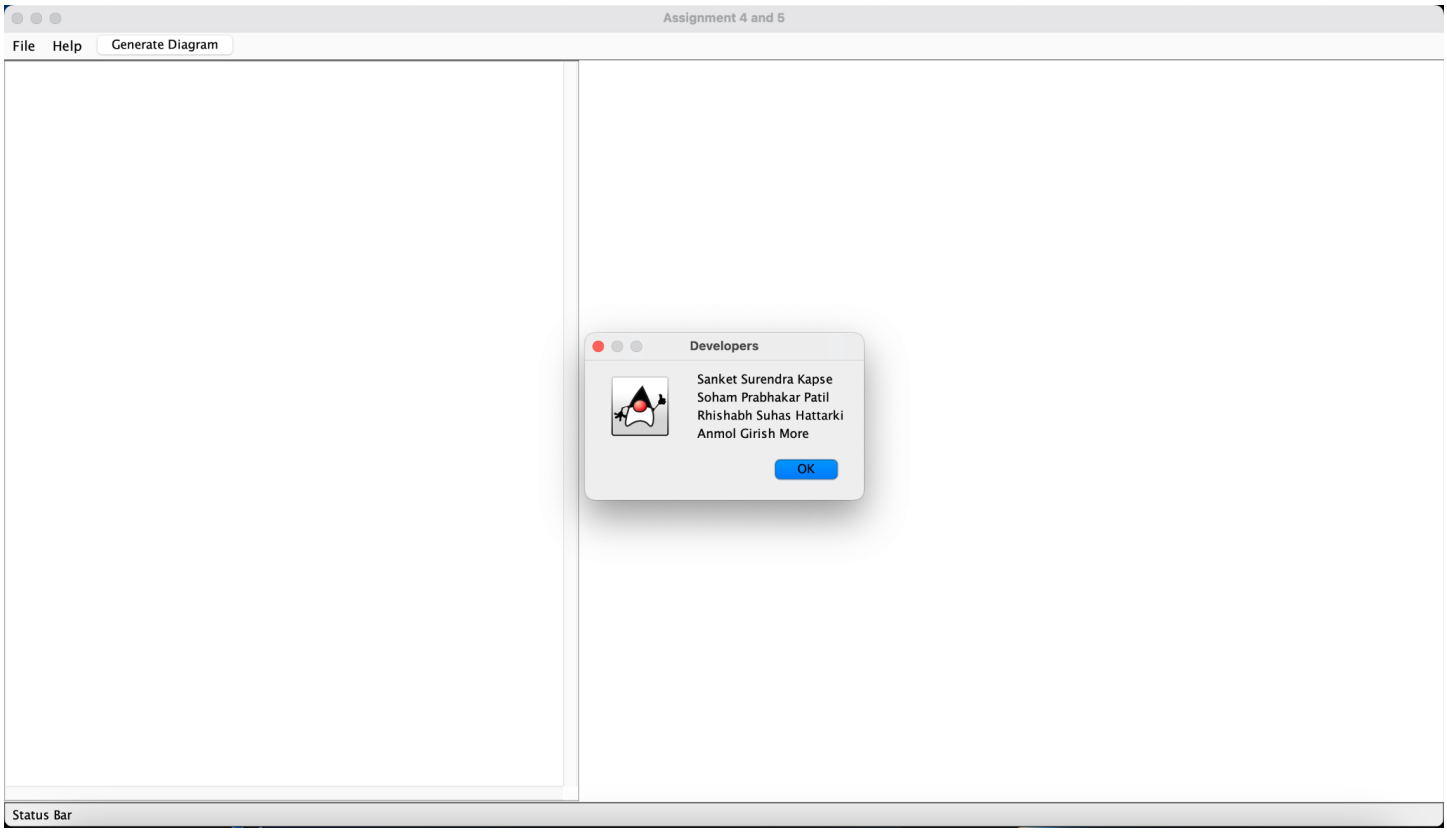
6. Help Menu



a. Design patterns used



b. **Developers**



7. **Drag and Drop (Refer Screenshots for reverse code generation)**
8. **Reverse Code Generation: Code text to Class Diagram**

Assignment 4 and 5

File Help Generate Diagram

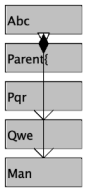
```
class Abc extends Parent{ {
    Pqr
    method() {
        Qwe
        Man
    }
}
```

Class dragged to other location

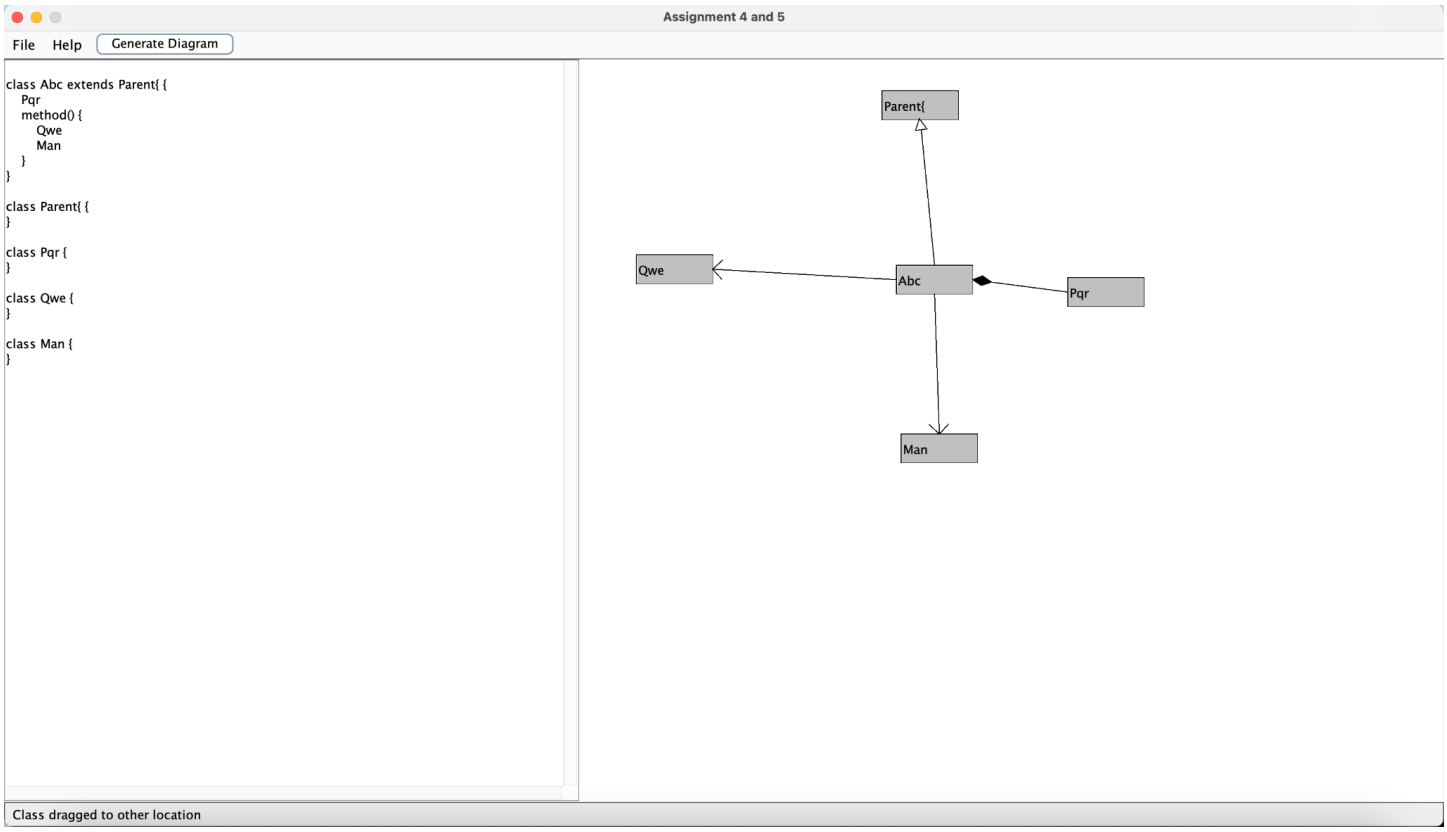
Assignment 4 and 5

File Help Generate Diagram

```
class Abc extends Parent{ {
    Pqr
    method() {
        Qwe
        Man
    }
}

class Parent{ {
}

class Pqr {
}

class Qwe {
}

class Man {
}
```



Class dragged to other location



Architecture Patterns used in Project

MVC:

Model

Class: Arrow: Decorate line with arrow.
Class: ClassBoxEncoder: Encodes ClassBox data into string.
Class: ClassSource: Blackboard data source stores classes, relationships and generated code.
Class CodeGenerator: Generate code text from class diagram.
Abstract Class: Connection: abstract class for relations.
Class: ConnectionEncoder: Encodes relationship data into string.
Enum: ConnectionTypes: Represents line, arrow, triangle, diamond
Class: Diamond: Decorate line with diamond.
Interface: Encoder: Interface for strategy pattern.
Class: FileHandler: Performs load and save file operations.
Class: HelpMenuHandler: generates the data needed for Help Menu.
Class: Line: Concrete line class.
Abstract Class: LineDecorator: Decorator abstract class for relations.
Class: LinePositions: Line data structure
Class: LoadHandlerArrow: Load association relation in blackboard.
Class: LoadHandlerDiamond: Load aggregation relation in blackboard.
Class: LoadHandlerTriangle: Load inheritance relation in blackboard.
Interface: RelationLoadHandler: Handler interface for loading relations into blackboard.
Class: ReverseCodeGenerator: Reverse code generator.
Class: Triangle: Decorate line with triangle.

View
Class: App: Parent frame.
Class: AppPanel: Contains CodePanel and ClassPanel.
Class: ClassBox: class box UI using Graphics.
Class: ClassPanel: Contains connections and classes. Observes ClassSource. Calls Chain of responsibility.
Class: CodePanel: Contains textual code representation. Observes ClassSource.
Interface: DrawConnection: Parent interface for drawing relationships.
Class: DrawLine: Draw line between 2 classes.
Class: DrawDiamond: Draw filled diamond.
Class: DrawArrow: Draw arrow.
Class: DrawTriangle: Draw empty triangle.
Class: FileMenu: Displays file menu in menu bar and displays save, load, new.
Class: GenerateDiagramButton: Button to generate class diagram.
Class: HelpMenu: Displays help menu in menu bar and displays developers and design patterns options.
Class: MessageDialog: Shows informational dialog box.
Class: StatusBar: Singleton JLabel to show status of actions.

Control
Class: FileController: Listens for File menu actions and calls methods in FileHandler class in model.
HelpMenuController: Monitors help menu actions and calls methods in HelpMenuHandler class in model.
Class: MouseController: Observes mouse actions and performs actions like drag and drop, class and relation creation, and vicinity checking.
Class: CodePanelController: Listens to changes in text code and adds, removes, and updates the class diagram.
Class: GenerateDiagramButtonController: On ‘Generate Diagram’ button click, creates class diagram after changes in text code.

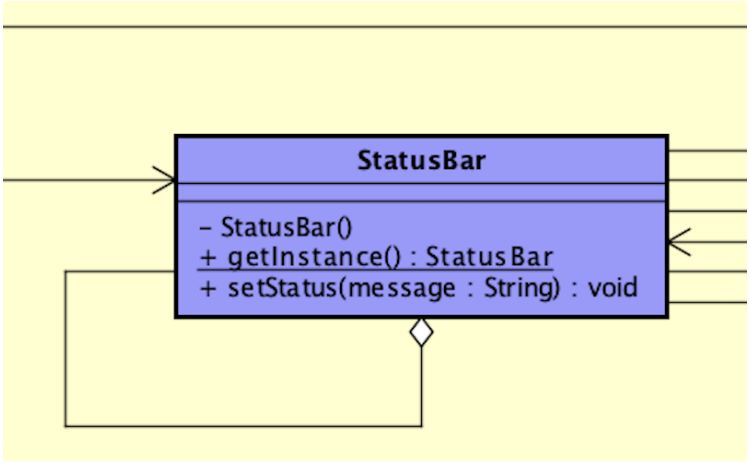
Blackboard Pattern:

Blackboard: ClassSource
KnowledgeSource: ReverseCodeGenerator
KnowledgeSource: CodePanelController
KnowledgeSource: CodeGenerator
KnowledgeSource: MouseController
KnowledgeSource: FileHandler
KnowledgeSource: ClassPanel
KnowledgeSource: CodePanel

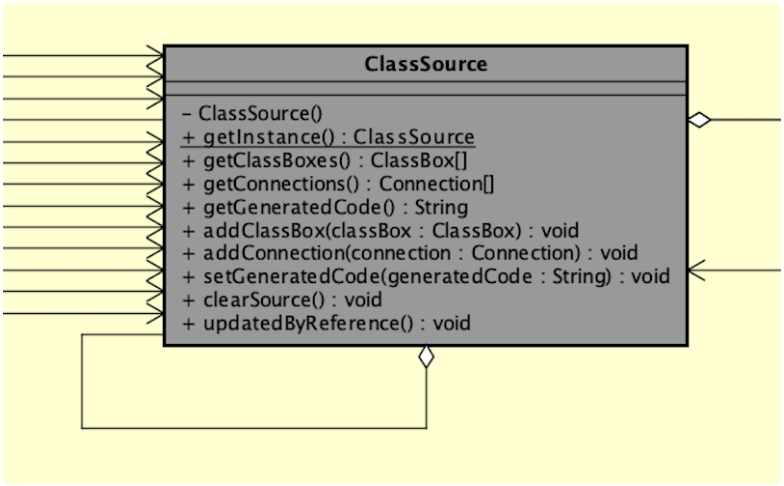
Design Patterns used in Project

Singleton Pattern:

Class: StatusBar

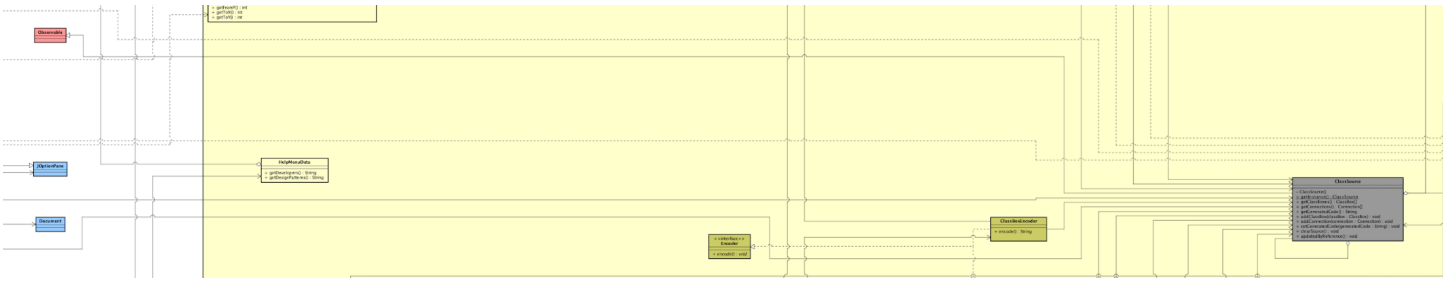


Class: ClassSource

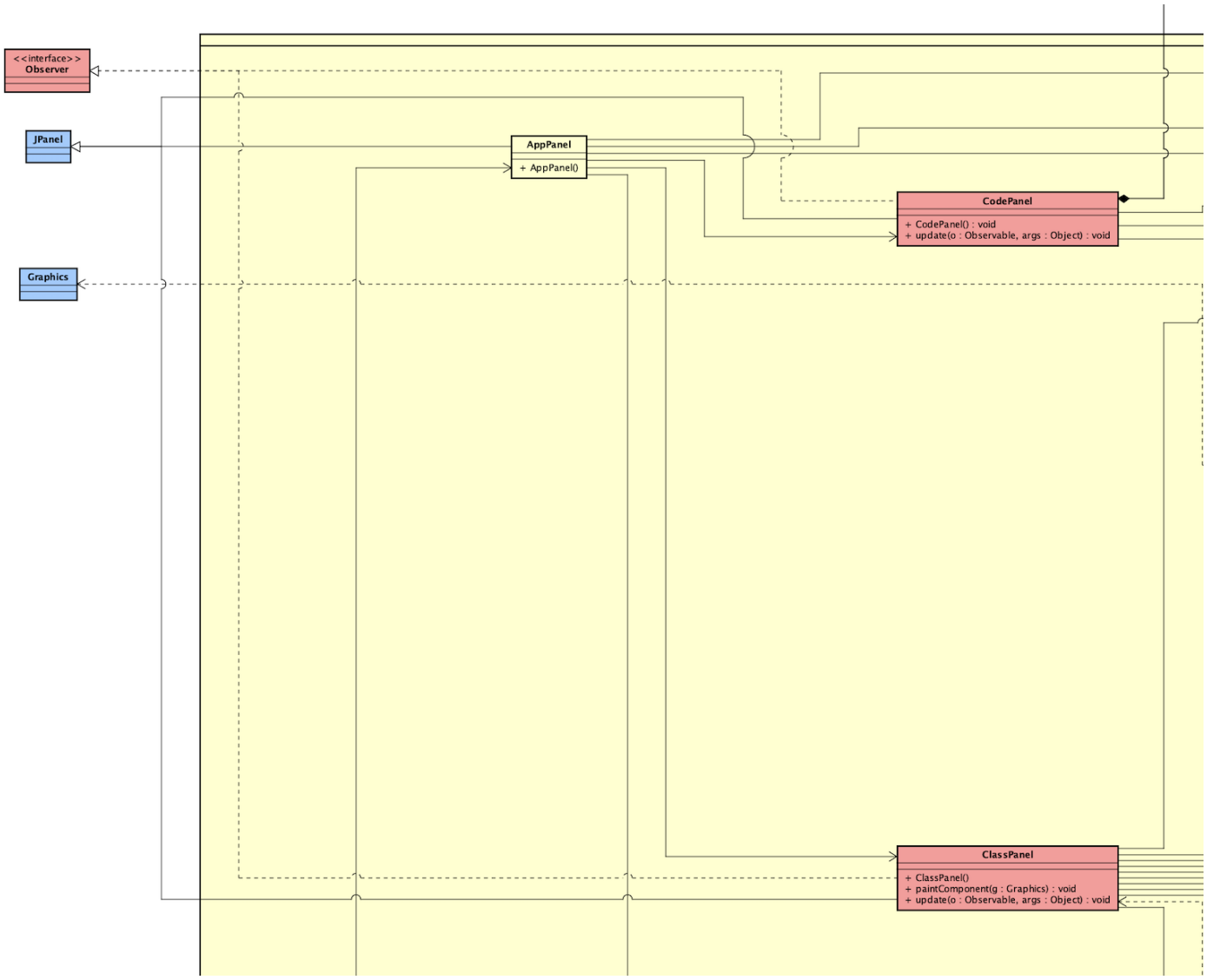


Observer Pattern:

Class: ClassSource -> Observable.

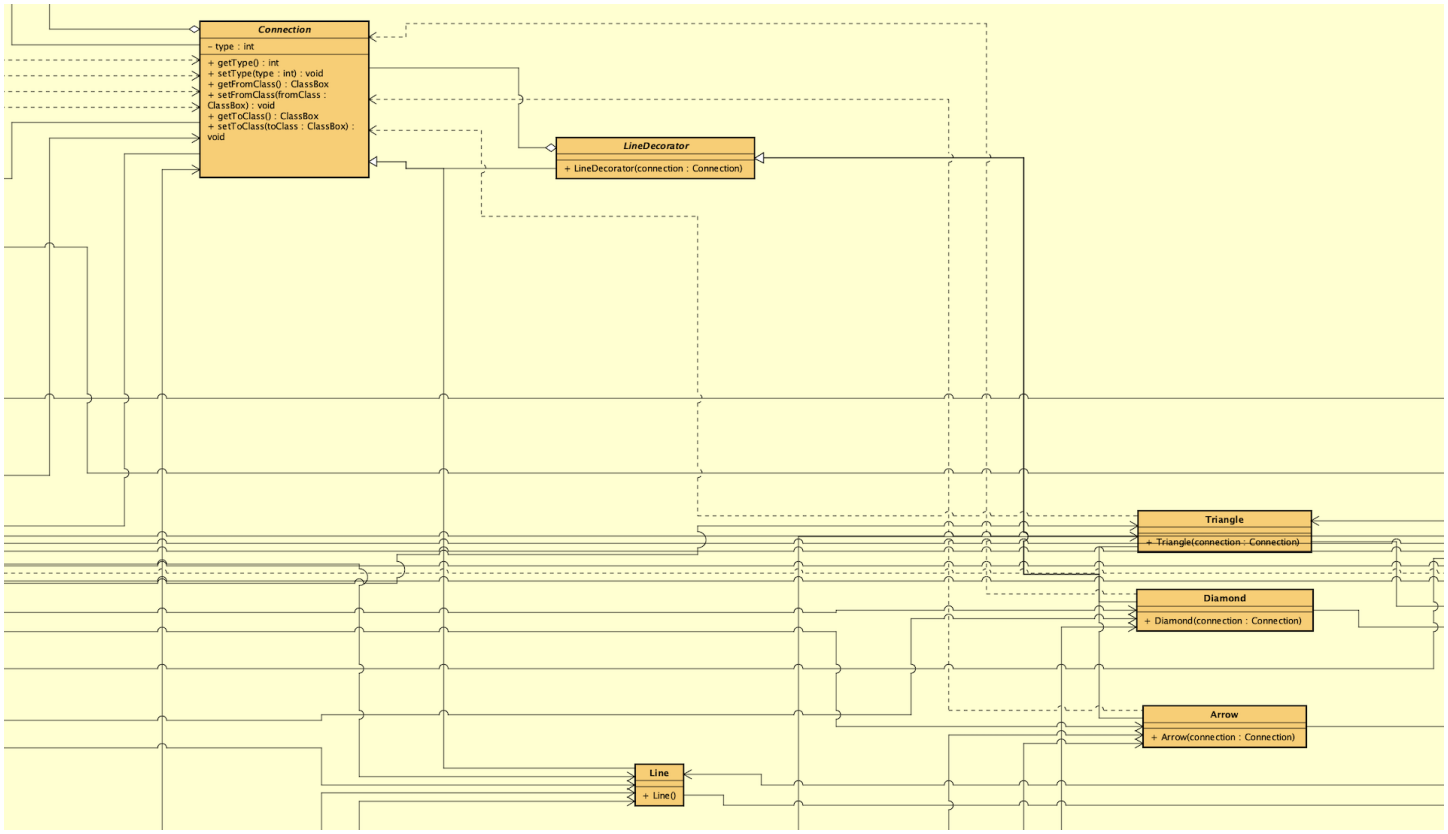


Class: ClassPanel -> Observer.
Class: CodePanel -> Observer.



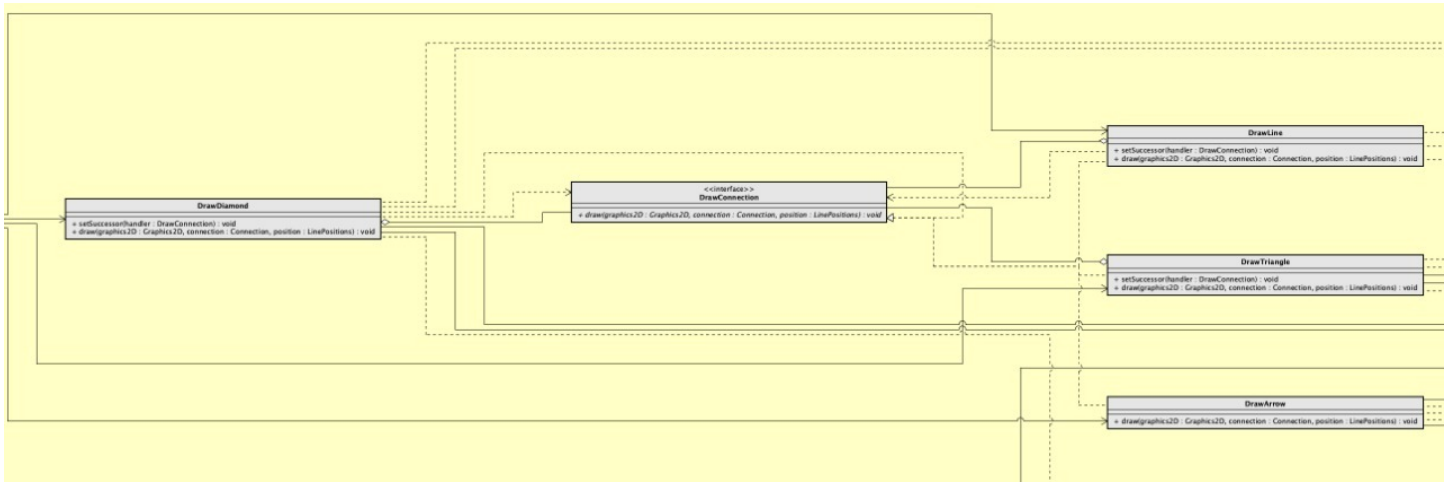
Decorator Pattern:

Abstract Class: Connection
Concrete Class: Line
Decorator Abstract Class: LineDecorator
Decorator Concrete Class: Triangle
Decorator Concrete Class: Diamond
Decorator Concrete Class: Arrow

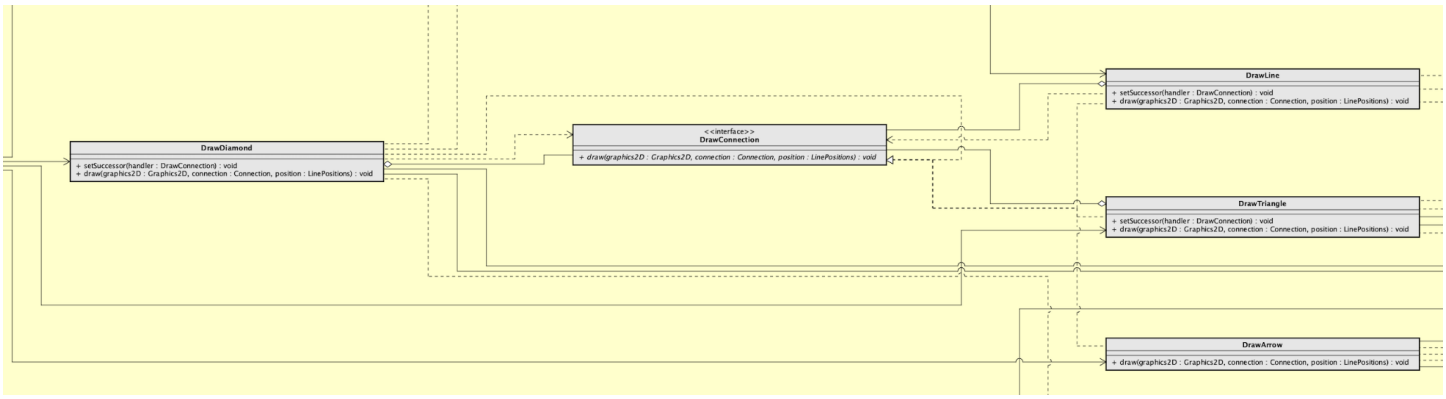


Chain of Responsibility Pattern:

Interface: DrawConnection
Class: DrawLine (Successor: DrawDiamond)
Class: DrawDiamond (Successor: DrawArrow)
Class: DrawArrow (Successor: DrawTriangle)
Class: DrawTriangle

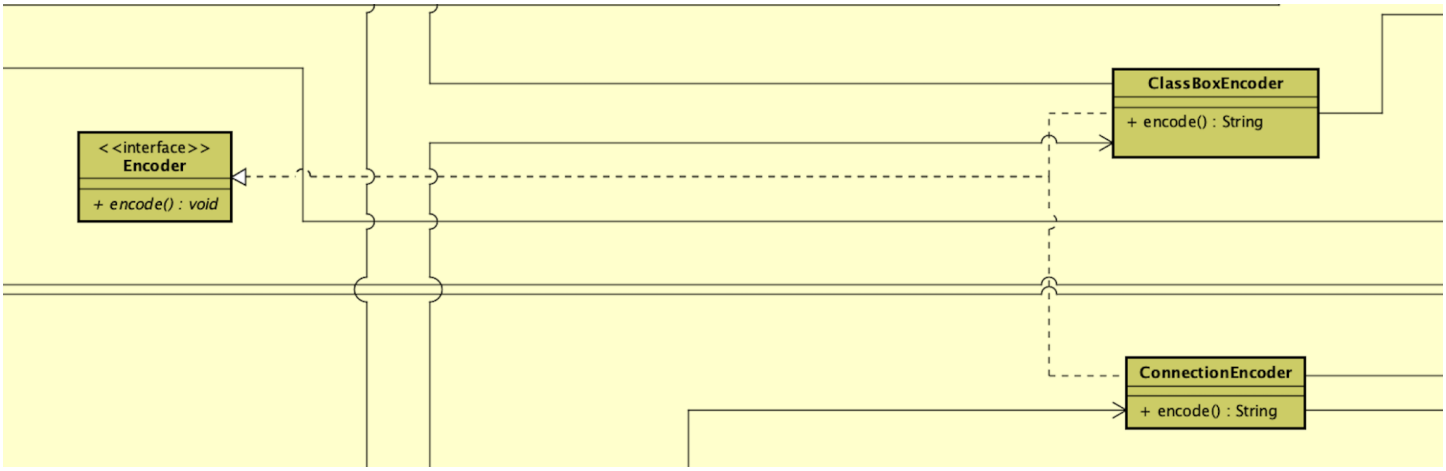


Interface: RelationLoadHandler
Class: LoadHandlerArrow (Successor: LoadHandlerTriangle)
Class: LoadHandlerTriangle (Successor: LoadHandlerDiamond)
Class: LoadHandlerDiamond



Strategy Pattern:

Interface: Encoder
Concrete Class: ClassBoxEncoder
Concrete Class: ConnectionEncoder



SOLID principles are implemented.