

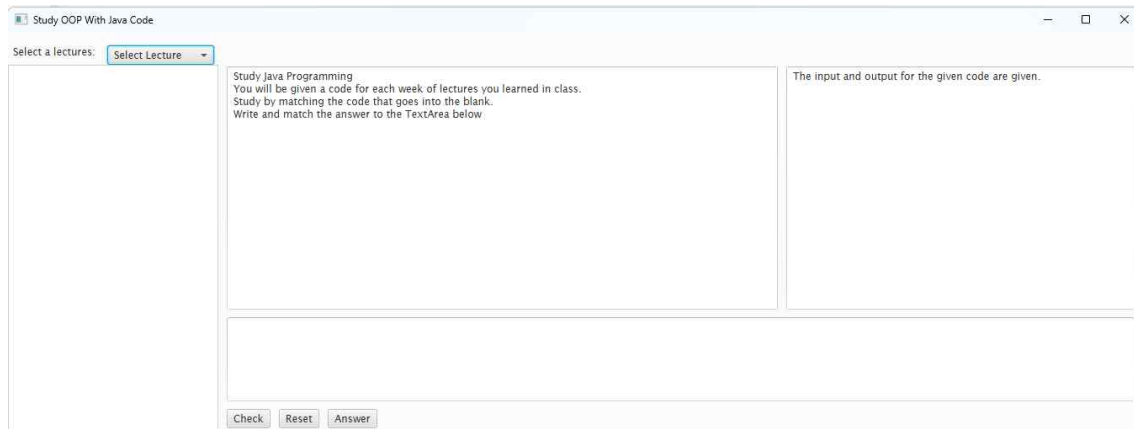
# OOP Term Project

## 1. The purpose of an application

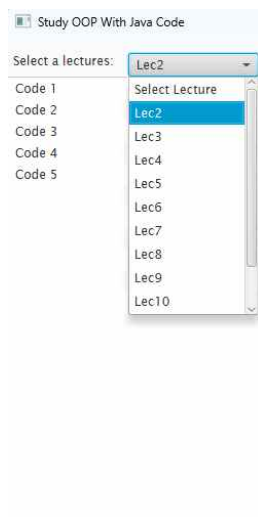
자바 언어를 공부하면서, 단순히 문법만 아는 것이 아니라 직접 코딩을 해보면서 흐름을 이해하고 적용할 수 있어야지 실력이 는다는 것을 느꼈다. 그래서 수업 때 다뤘던 강의자료에 있는 예제 코드들을 가져와서 코드 사이사이에 뚫려있는 빈칸에 어떤 코드가 들어가야 할지 생각해 보면서 정답을 맞추는 application을 만들었다. 이 application을 사용하면서 확실히 문법을 이해했는 지, 프로그램의 흐름을 알 수 있는지 스스로 확인하며 실력 또한 발전할 수 있을 것이다.

## 2. Use case, and result when the application is running.

처음 실행화면은 다음과 같다.

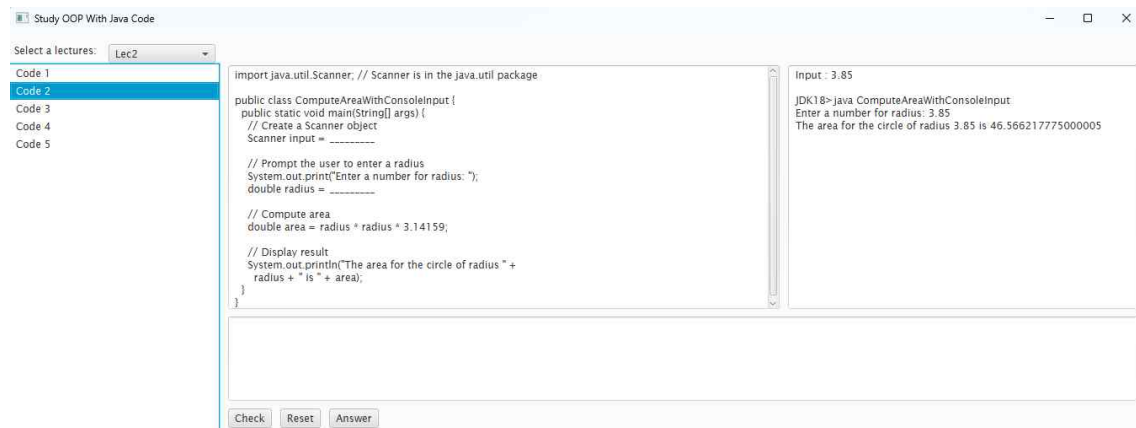


왼쪽의 ComboBox와 ListView를 통해 강의 주차와 그 강의 속 코드를 선택할 수 있다.

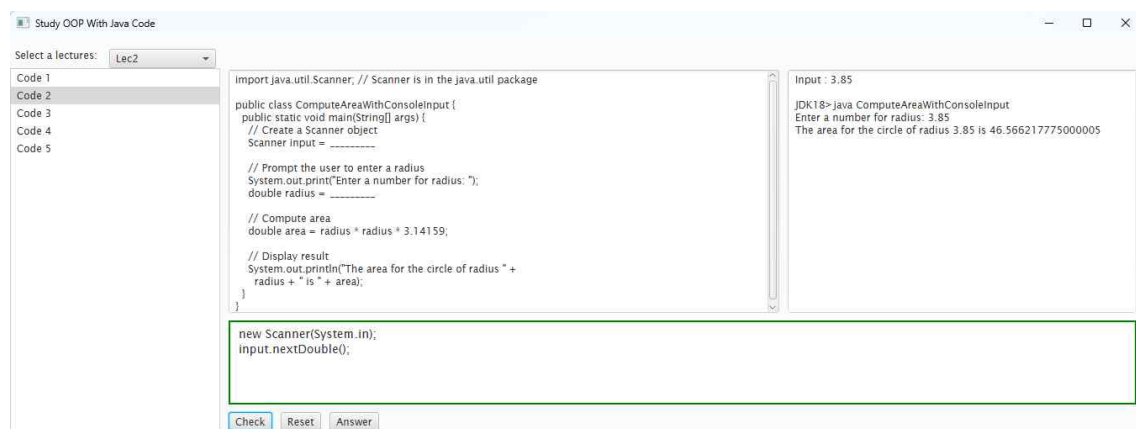
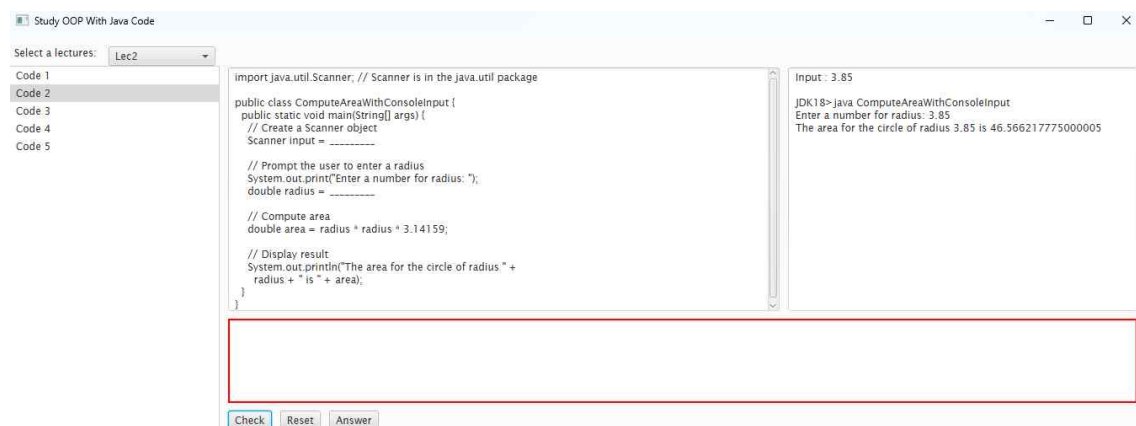


선택 후에 왼쪽 TextArea에 빈칸이 풀려있는 코드와 오른쪽 TextArea에 코드의 Input 값 (Input이 있다면)과, Output 값을 보여준다.

오른쪽 TextArea를 보면서 코드를 추측해 아래 TextArea에 답을 넣는다.



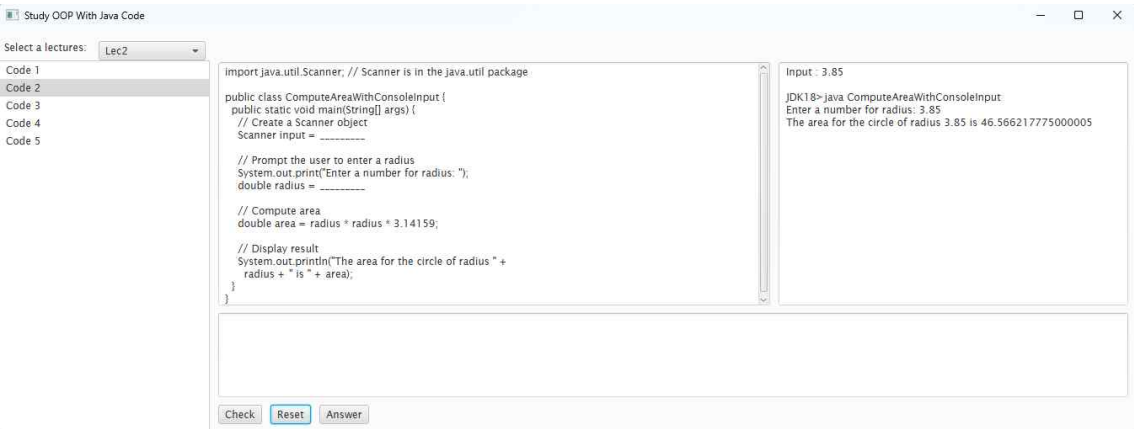
답을 안 넣었거나 틀린 답을 넣고 Check 버튼을 누르면 빨간색 테두리가, 옳은 코드를 넣고 누르면 초록색 테두리가 나온다.



답을 몰라 정답 코드를 확인하고 싶으면 Answer 버튼을 눌러 답을 확인한다. 이 때 Check 버튼은 비활성화된다.



Reset 버튼을 누르면 Check 버튼이 다시 활성화되고 정답란이 비워진다.



### 3. UML diagram

Main
<ul style="list-style-type: none"><li>- codeDescriptions: String[][]</li><li>- outputResults: String[][]</li><li>- correctAnswers: String[][]</li><li>- lectures: String[]</li><li>- codes: String[][]</li><li>- descriptionPane: DescriptionPane</li></ul>

- lectureSelectorPane: LectureSelectorPane
+ start(Stage): void
- loadDescriptions(): void
- loadAnswers(): void
- loadResults(): void
+ main(String[] args): void

LectureSelectorPane
- cbo: ComboBox<String>
- listView: ListView<String>
- lectures: String[]
- codes: String[][]
- descriptionPane: DescriptionPane
+ LectureSelectorPane(lectures: String[], codes: String[][], descriptionPane: DescriptionPane)
+ getComboBox(): ComboBox<String>
+ getListView(): ListView<String>
- updateListView(int): void
- updateDescription(int, int): void

DescriptionPane
- description: TextArea
- codeResult: TextArea
- userInput: TextArea
- btCheck: Button
- btReset: Button
- btAnswer: Button
- correctCode: String
+ DescriptionPane()
+ setDescription(String): void
+ setCodeResult(String): void
+ setCorrectCode(String): void
- checkAnswer(): void
- resetInput(): void
- showAnswer(): void