

# Smart Time Table System -Implement

200412325 박동현

200412361 하희진

200611460 김정태

200611490 오준

200711452 유지호

200813530 품이핑

# Introduction

1. Introduce Used UML Tool
  2. Sequence of GUI
  3. Trace the 'Smart'
4. Difference of Class between each Development phase

# 1. Introduce Used UML Tool



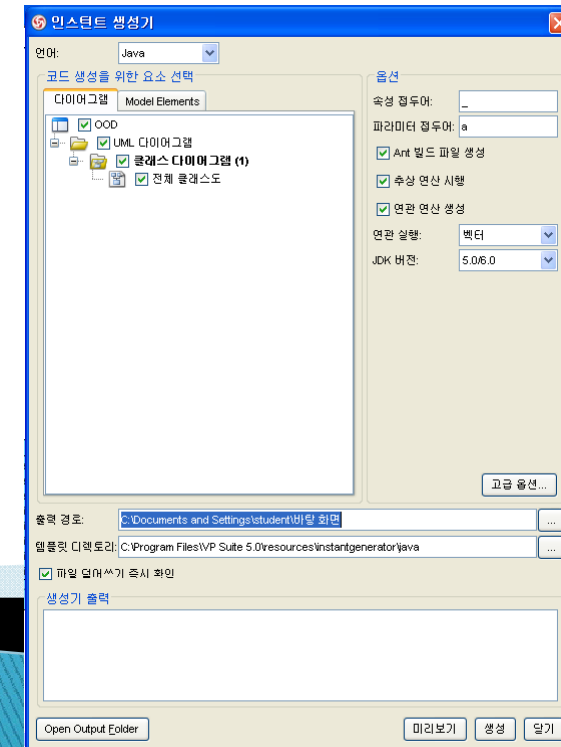
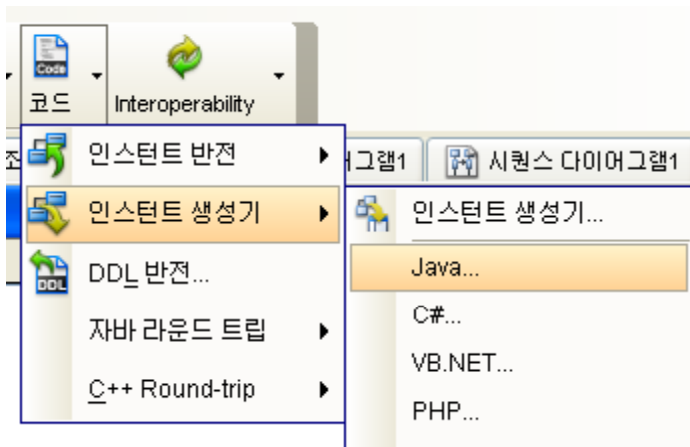
Visual Paradigm :

무료 UML Tool로써 DB Architecture 지원/자료 정리 용  
이 등의 장점이 있음.



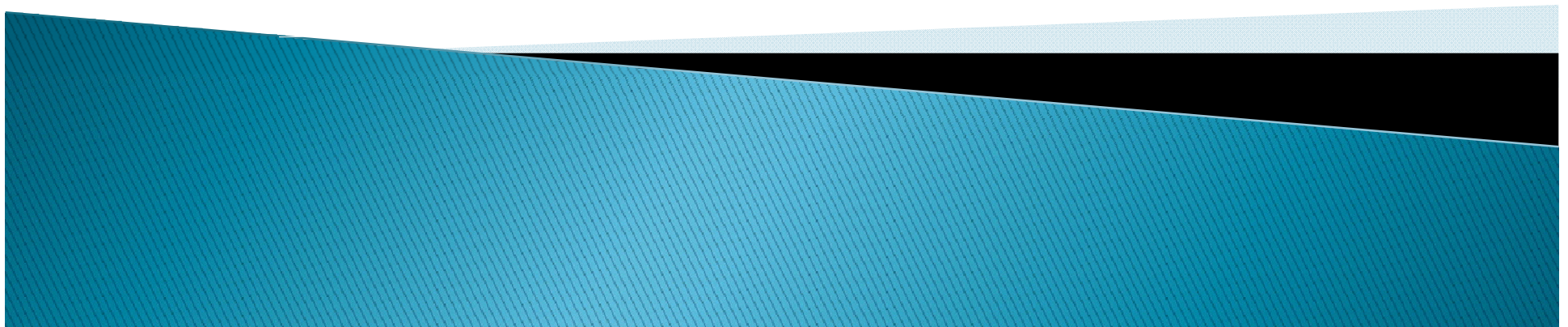
# 1. Introduce Used UML Tool

Visual Paradigm은 Code Generator를 제공하며 이를 기반으로 하여 Source Code 구현 실시



## 2. Sequence of GUI

Demonstration!

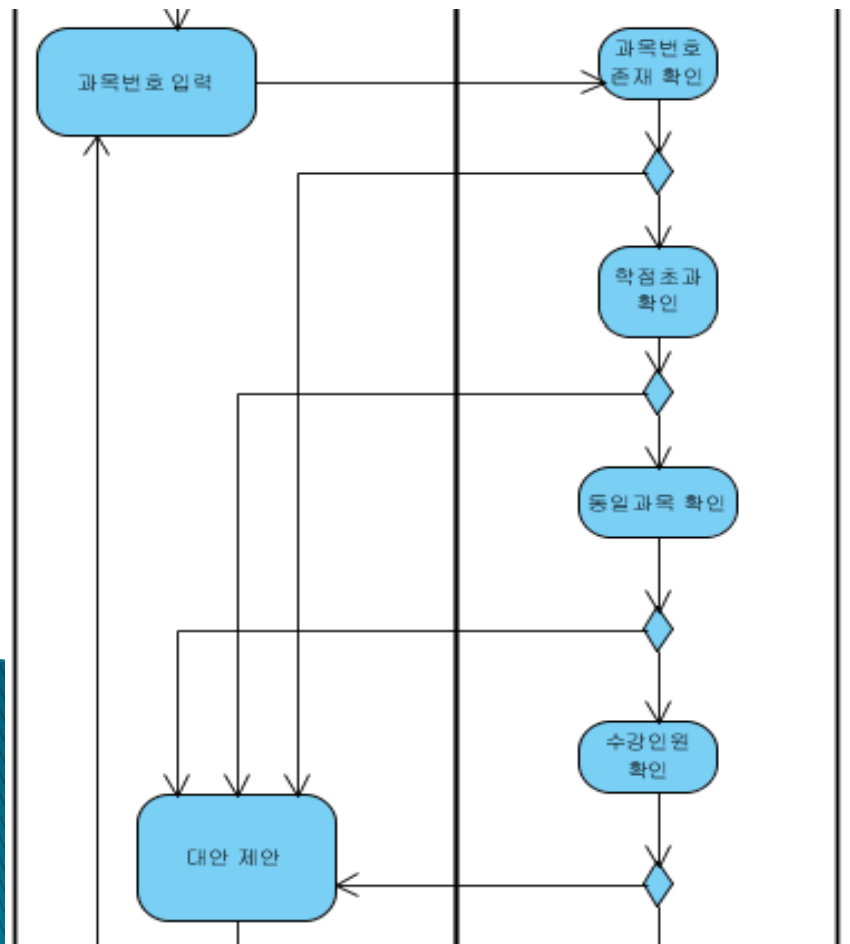


# 3. Trace the 'Smart'

## <4.2.2. 수강신청>

in SRS from Analysis step

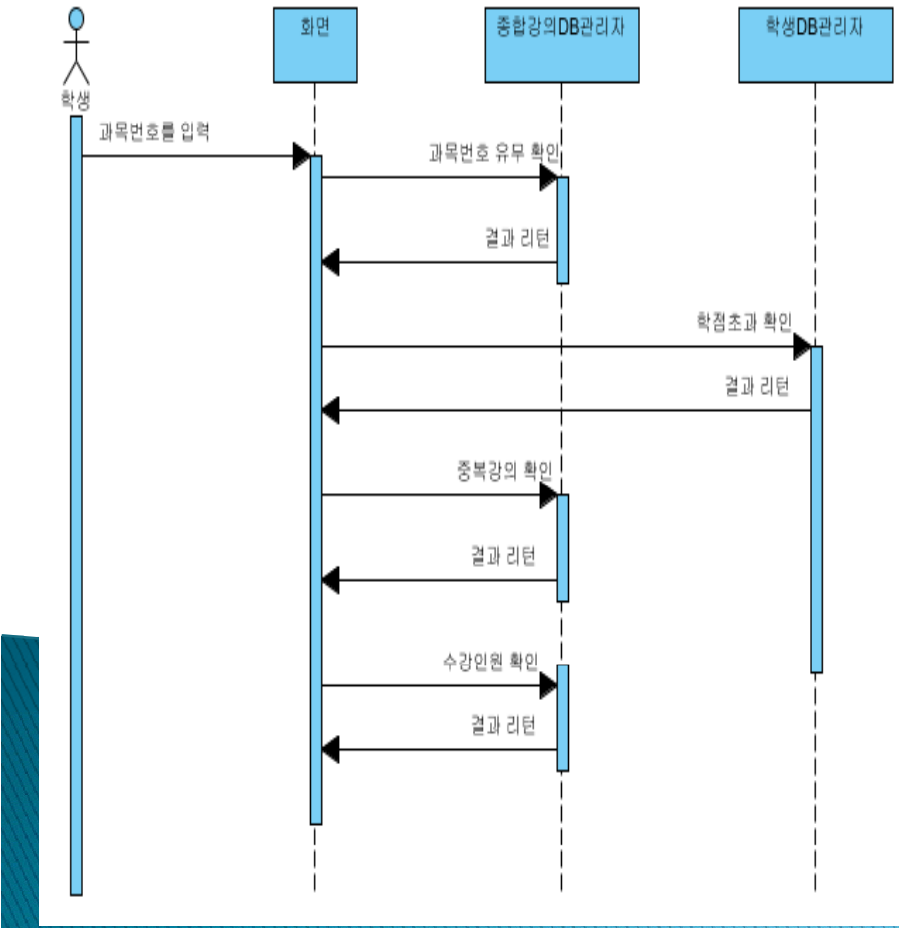
사용자의 입력을 거부하는 데  
대한 대안을 제시하는 AI의  
개념 제시



# 3. Trace the 'Smart'

<4.3 Sequence Diagram>  
in SRS from Analysis step

AI의 진입 요건에 대한 분석





# 3. Trace the 'Smart'

Subject
-string professor_name
-string room_number
-string subject_name
-int start_time
-int lecture_time
-int lecture_day
-int max_student_num
-int cur_student_num
+Subject(int subject_num, String subje...

SubjectDBM
+Hashtable getAllSubject(void)
+Hashtable get2Subject(void)
+Hashtable getSameNameSub(int subNum)
+Hashtable getSameTimeSub(int subNum)

<4.2 Design entity attributes>  
in SDD from Design step

Student
-int restrict_point
-int cur_point
-string student_name
-string student_number
-linked_list TimeTable
+linked_list getTimeTable()
+Student(string ID, String name, int maxnum, int cumum, linkedlist timetabl)
+bool delSubject(int subject_num)
+bool setSubject(int subject_num)

Subject Class, SubjectDBM  
Class, Student Class와 AI  
Class 간의 Data Flow 정의



# 3. Trace the 'Smart'

AI
-SubjectDBM sub -Student stu
+Hashtable sixTeen(linked_list result, Hashtable result) +Hashtable overStu(Hashtable result) +Hashtable sameSub(Hashtable result) +Hashtable sameTimeSub(Hashtable result) +int controlState(int state)

## <4.2.7> AI Class

in SDD from Design step

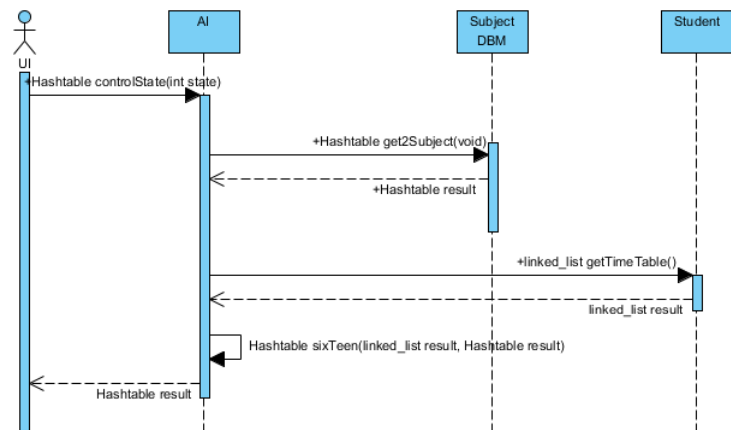
AI Class의 설계. Attribute  
및 Operator 정의.

# 3. Trace the 'Smart'

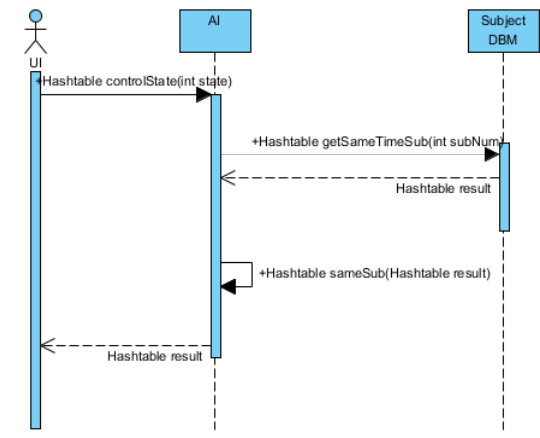
## <5.1.2.6 AI의 Sequence Diagram> in SDD from Design step

AI가 가능하게 되는 네 가지 예외 상황에 대한 Sequence Diagram 언급

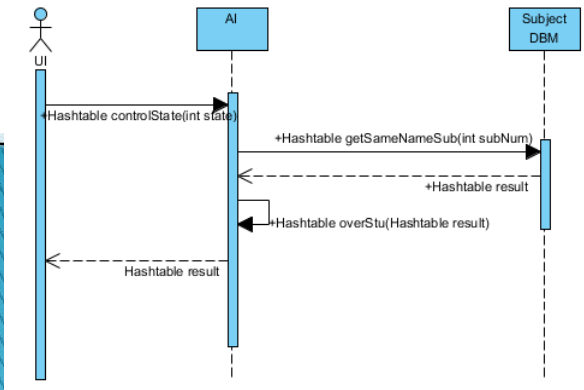
sd AL1\_16/18 3학점 신청시



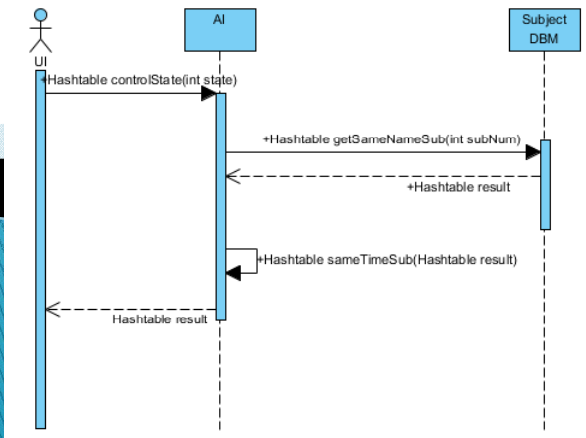
sd 같은 과목 중복 신청



sd 수강인원 초과

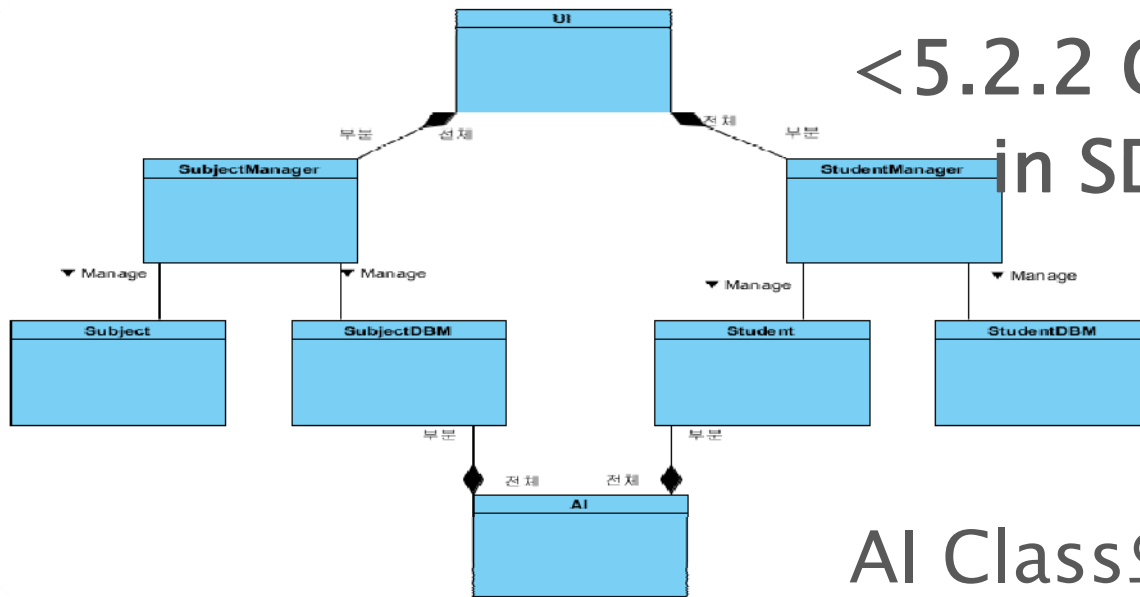


sd 시퀀스 다이어그램1



# 3. Trace the 'Smart'

<5.2.2 Class Dependency>  
in SDD from Design step

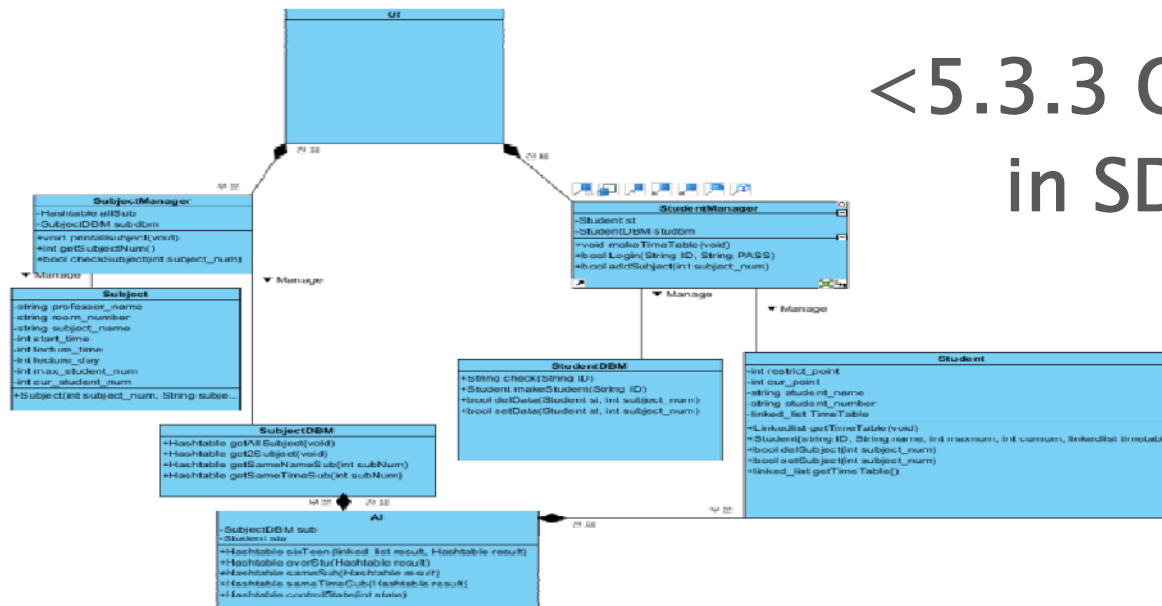


AI Class와 System 내의 다른  
Class 사이의 Dependency



# 3. Trace the 'Smart'

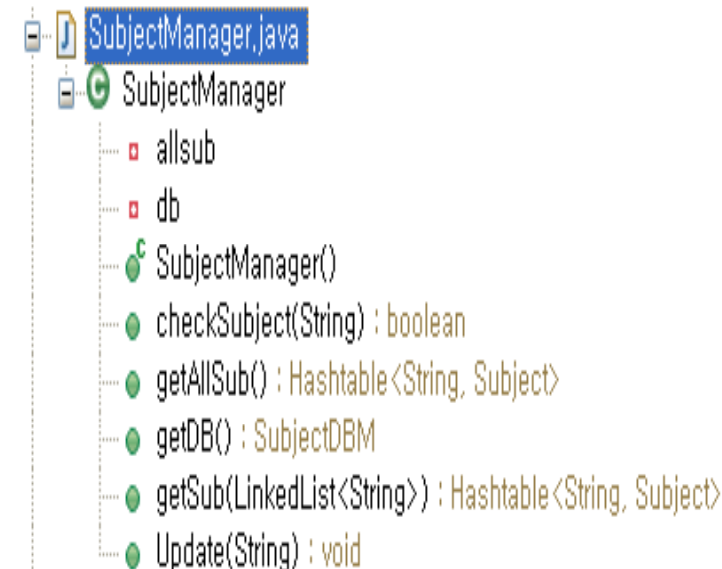
<5.3.3 Class Diagram>  
in SDD from Design step



# 4. Difference of Class between each Development phase

## <SubjectManager Class>

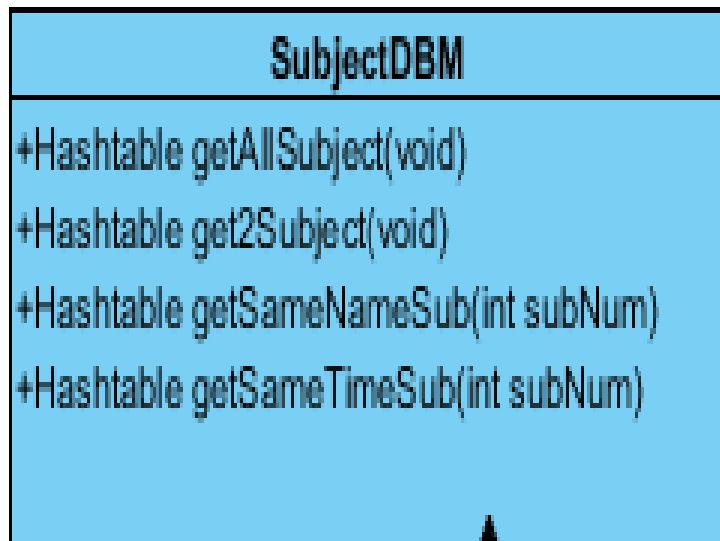
SubjectManager
-Hashtable allSub
-SubjectDBM subdbm
+void printallsubject(void)
+int getSubjectNum()
+bool checkSubject(int subject_num)



```
SubjectManager.java
├── SubjectManager
│   ├── allsub
│   ├── db
│   ├── SubjectManager()
│   ├── checkSubject(String) : boolean
│   ├── getAllSub() : Hashtable<String, Subject>
│   ├── getDB() : SubjectDBM
│   ├── getSub(LinkedList<String>) : Hashtable<String, Subject>
│   └── Update(String) : void
```

# 4. Difference of Class between each Development phase

<SubjectDBM Class>



```
SubjectDBM.java
├── SubjectDBM
│   ├── con
│   ├── driver
│   ├── rs1
│   ├── sql1
│   ├── stmt
│   ├── Sub
│   ├── url
│   ├── SubjectDBM()
│   ├── get2Subject() : Hashtable<String, Subject>
│   ├── getSameNameSub(String) : Hashtable<String, Subject>
│   ├── getSameTimeSub(String) : Hashtable<String, Subject>
│   ├── getSub() : Hashtable<String, Subject>
│   ├── setAllSub() : void
│   └── UpdateSubject(Subject) : void
```

The screenshot shows the file structure of SubjectDBM.java. The SubjectDBM class is highlighted with a green circle. The methods listed are: SubjectDBM(), get2Subject() : Hashtable<String, Subject>, getSameNameSub(String) : Hashtable<String, Subject>, getSameTimeSub(String) : Hashtable<String, Subject>, getSub() : Hashtable<String, Subject>, setAllSub() : void, and UpdateSubject(Subject) : void.



# 4. Difference of Class between each Development phase

## <Subject Class>

Subject
-string professor_name
-string room_number
-string subject_name
-int start_time
-int lecture_time
-int lecture_day
-int max_student_num
-int cur_student_num
+Subject(int subject_num, String subje...

```
Subject.java
├── Subject
└── TimeSub
    ├── end
    ├── room_number
    ├── start
    ├── week
    ├── getEnd() : int
    ├── getLec_room() : String
    ├── getStart() : int
    ├── getWeek() : int
    ├── setEnd(int) : void
    ├── setLec_room(String) : void
    ├── setStart(int) : void
    ├── setWeek(String) : void
    └── toString() : String
```

```
Subject.java
├── Subject
│   ├── cur_Student
│   ├── max_Student
│   ├── point
│   ├── professor
│   ├── sub_name
│   ├── sub_num
│   ├── time
│   ├── Subject()
│   ├── deac_Student() : void
│   ├── getCur_Student() : int
│   ├── getMax_Student() : int
│   ├── getPoint() : int
│   ├── getProfessor() : String
│   ├── getSub_name() : String
│   ├── getSub_num() : String
│   ├── getTime() : LinkedList<TimeSub>
│   ├── inc_Student() : void
│   ├── setCur_Student(int) : void
│   ├── setMax_Student(int) : void
│   ├── setPoint(int) : void
│   ├── setProfessor(String) : void
│   ├── setSub_name(String) : void
│   ├── setSub_num(String) : void
│   ├── setTime(LinkedList<TimeSub>) : v
│   └── toStringTime() : String
└── TimeSub
```

\*Subject Class와 TimeSub Class로 분화

# 4. Difference of Class between each Development phase

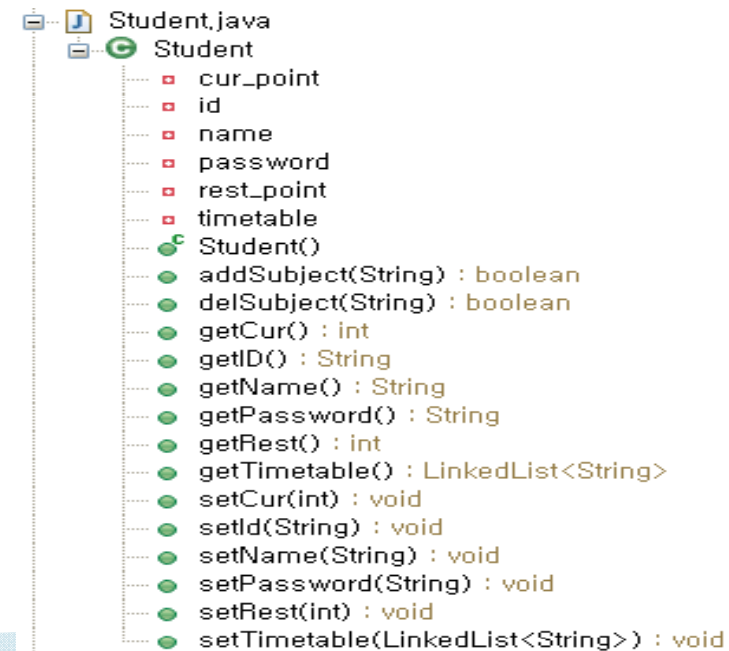
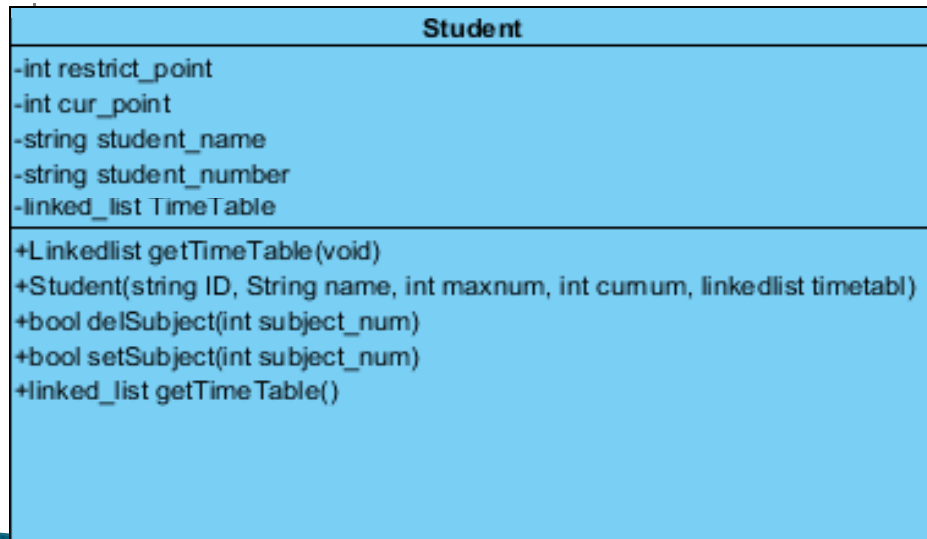
## <StudentManager Class>

StudentManager
-Student st -StudentDBM studbm
+void makeTimeTable(void) +bool Login(String ID, String PASS) +bool addSubject(int subject_num)

```
StudentManager.java
StudentManager
  st
  studbm
  StudentManager()
  addSubject(String) : int
  bool_addSubject(Object) : void
  compare(String, String) : boolean
  delSubject(String) : boolean
  getDBM() : StudentDBM
  getStudent() : Student
  Login(String, String) : boolean
  setStudent(String) : void
  void_makeTimeTable(Object) : void
```

# 4. Difference of Class between each Development phase

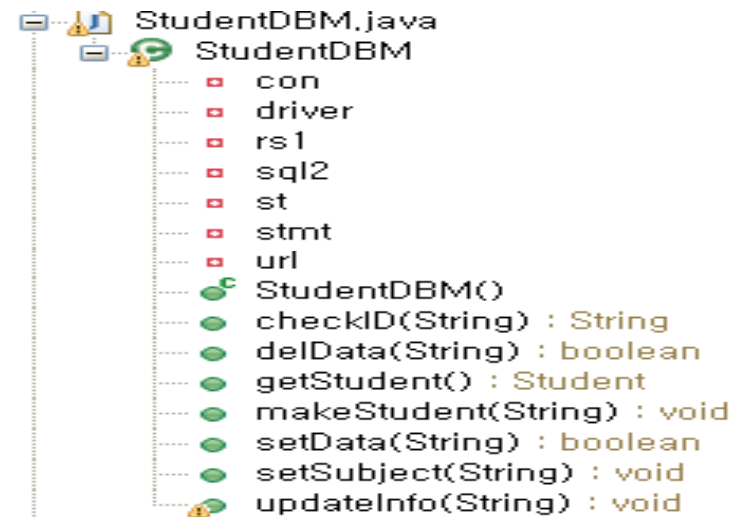
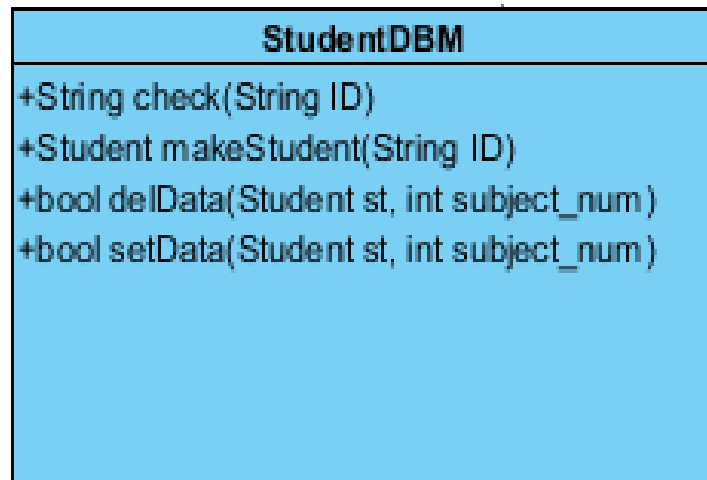
## <Student Class>





# 4. Difference of Class between each Development phase

<StudentDBM Class>



# Thank You!

1. ’

