**JAVA 학생관리프로그램**

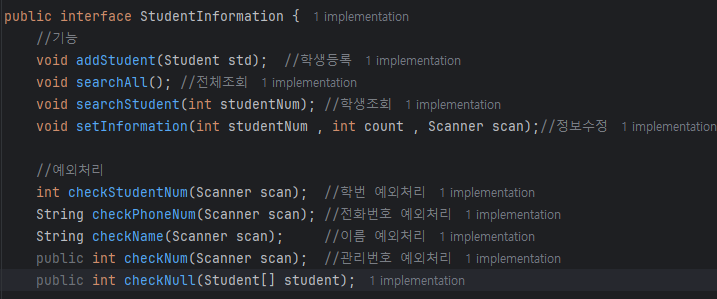
**백앤드 개발자 부트캠프(스프링) 27주차**

**제출일자 : 2024년 8월 22일 (목)**

**김명찬**

**목차**

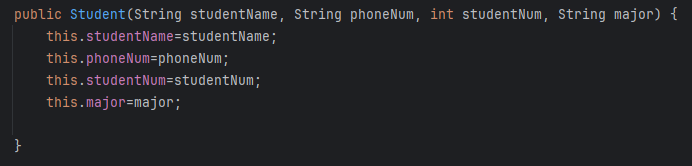
1. **구조**
2. [StudentInformation --------------------------------------------------------------------------(2)](#페이지2)
3. [Student ---------------------------------------------------------------------------------------(3)](#페이지3)
4. [StudentInformationSystem ------------------------------------------------------------------(5)](#페이지5)
5. [StudentInformationMain --------------------------------------------------------------------(7)](#페이지7)
6. **기능**
7. [학생 등록 --------------------------------------------------------------------------------------(8](#페이지8))
8. [전체 조회 --------------------------------------------------------------------------------------(9)](#페이지9)
9. [학생 조회 -------------------------------------------------------------------------------------(10)](#페이지10)
10. [정보 수정 -------------------------------------------------------------------------------------(11)](#페이지11)
11. [프로그램 종료 --------------------------------------------------------------------------------(12)](#페이지12)
12. [예외처리 ---------------------------------------------------------------------------------(13~17)](#페이지13)
13. **구조**



**StudentInformation 인터페이스**

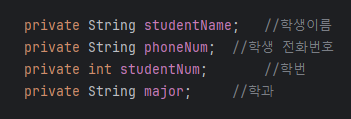
코드의 간결함과 가독성 그리고 특정 기능을 수정하거나 확장해야 할 경우 구현한 클래스의 해당 메소드만 확인하면 되기에 유지보수가 쉬워 인터페이스를 만들어 해당 프로그램에서 필수로 구현해야 하는 메소드를 정의했습니다.

**Student 클래스**



Student 클래스 생성자

학생이름(studentName) , 전화번호(phoneNum) , 학번(studentNum) , 학과(major)을 생성자를 통해 받아 생성된 Student 객체의 필드에 각각 저장합니다.

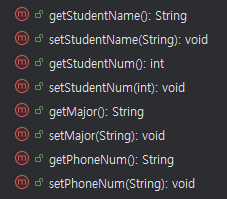


studentName : 학생이름을 담기위한 필드 (String 타입)

phoneNum : 학생 전화번호를 담기 위한 필드 (String 타입)

studentNum : 학번을 담기위한 필드 (int 타입)

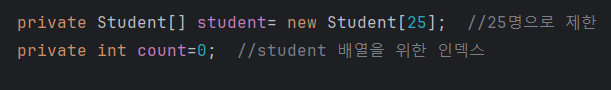
major : 학생의 학과를 담기위한 필드 (String 타입)



필드들을 private으로 지정했기 때문에 외부에서 객체를 수정(set)하기 위한 메소드들과 필드를 얻기 위한 (get)메소드들을 구현했습니다.

**StudentInformationSystem 클래스**

 클래스의 기본 생성자

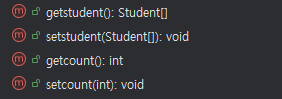


학생들의 정보를 담기 위한 Student 타입 배열

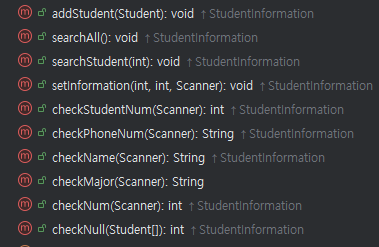
Student 타입 배열의 인덱스를 위한 count 필드



입력을 위한 Scanner 객체생성



배열과 필드를 private로 지정했기 때문에 외부에서 접근하기 위한 메소드



요구하는 기능들 (1.학생등록 , 2.전체조회 , 3.학생조회 , 4.정보수정)에 대한 메소드

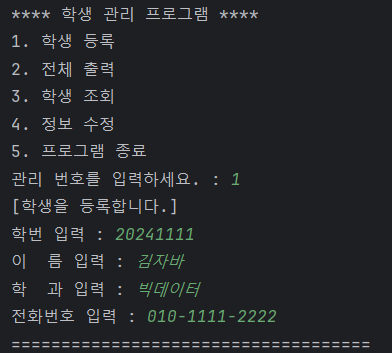
예외처리를 위한 메소드(5.학번 예외처리 , 6.전화번호 예외처리 , 7.이름 예외처리 , 8.학과 예외처리 , 9. 관리번호 예외처리 10. 등록된 학생이 없을 때 예외처리)

**StudentInformationMain 클래스**



구현했던 기능들을 통합하여 실행하는 메인 클래스

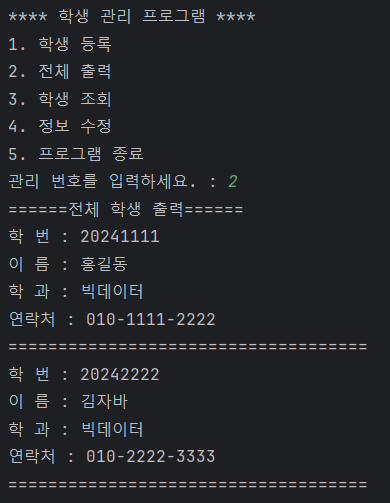
1. **기능**
2. **학생등록**



public void addStudent(Student student)

학번 , 이름 , 학과 , 전화번호를 입력받아 Student 객체의 생성자 매개변수로 받아 Student 객체로 만든 후 Student 타입의 배열에 객체 저장

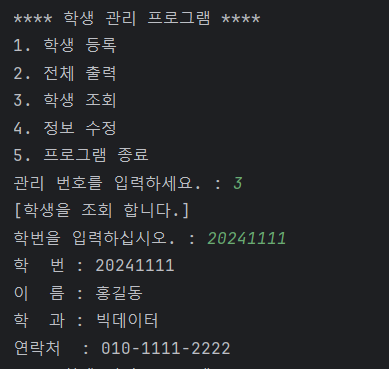
1. **전체 조회**



public void searchAll()

Student 타입의 배열이 null 이 아니면 향상된 반복문을 통해 배열에 있는 모든 학생들의 정보들을 전부 출력

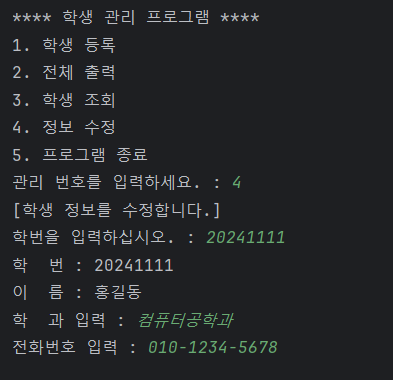
1. **학생 조회**



public void searchStudent(int studentNum)

학번을 매개변수로 받아 향상된 반복문을 통해서 입력받은 학번과 배열 내부에 있는 Student 객체의 학번이 일치하면 해당 학생의 정보 전부 출력

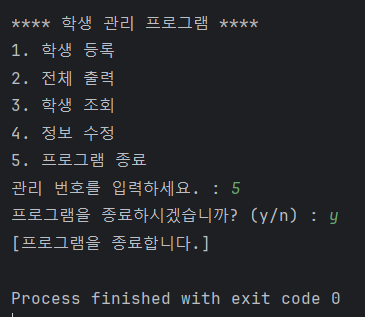
1. **정보 수정**

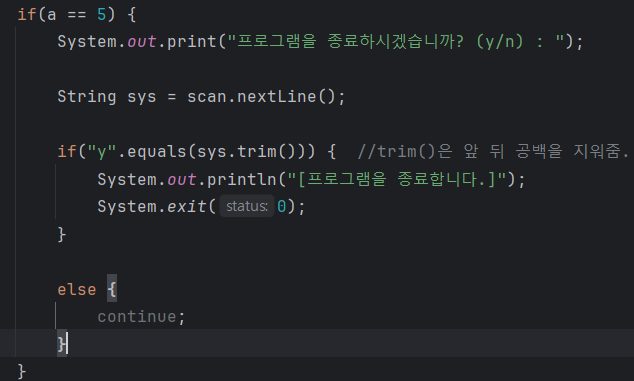


setInformation(int studentNum , int count , Scanner scan)

학번(studentNum) , 수정하려고 하는 학생이 저장되어 있는 배열의 인덱스(count) , 입력받기 위한 Scanner 객체)를 매개변수로 받아 해당 학생의 학과 전화번호 수정(setMajor , setPhoneNum 사용

1. **프로그램 종료**





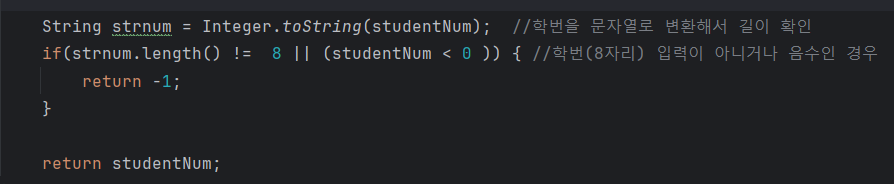
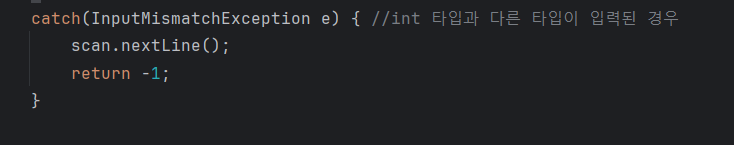
입력받은 값이 문자열 y 또는 Y와 일치했을 때 프로그램 종료 문구와 함께 종료

n 입력 시 프로그램 그대로 실행

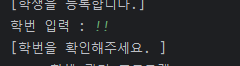
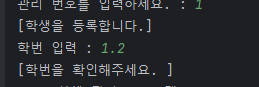
1. **예외처리**

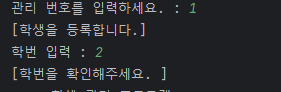
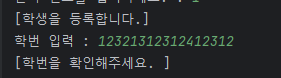
**학번 예외처리**

public int checkStudentNum(Scanner scan)

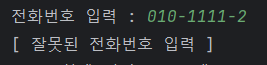
 

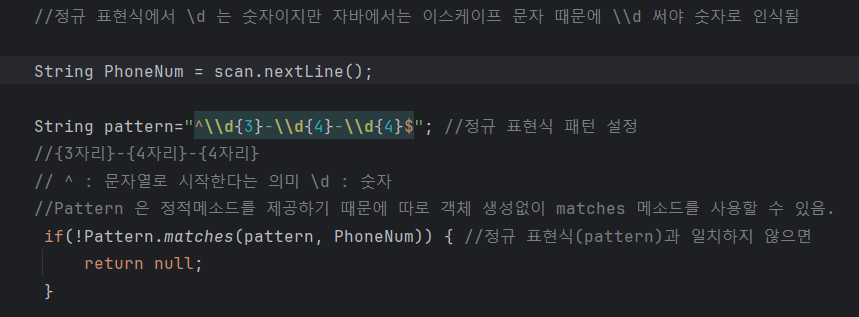
 

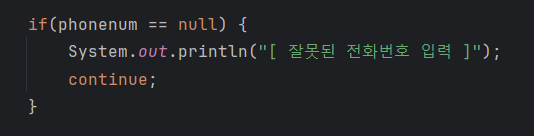
학번 8자리 입력이 되지 않았을 때와 음수가 입력되었을 때 학번 타입(int) 과 다른 타입이 입력이 되었을 때 예외처리

**전화번호 예외처리**

**public String checkPhoneNum(Scanner scan)**







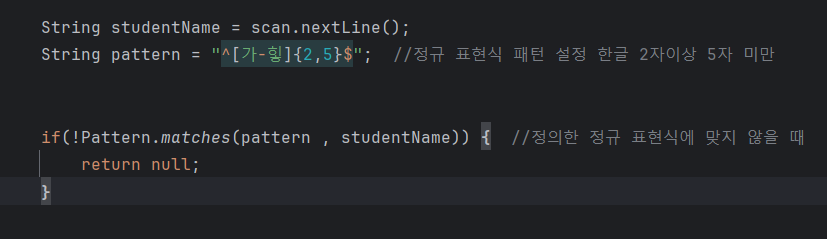
 

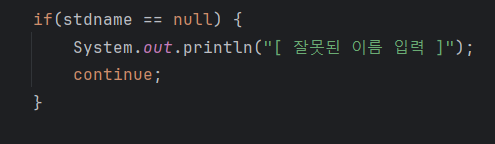
입력받은 값이 정규 표현식 패턴에 일치하지 않으면 null을 반환

**이름 예외처리**

**public String checkName(Scanner scan)**







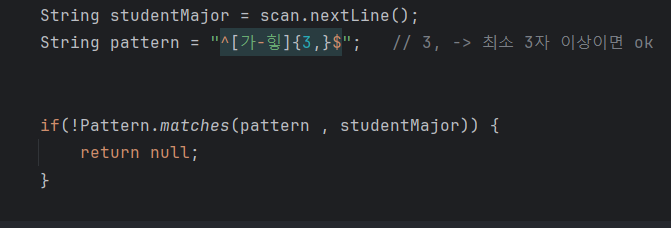


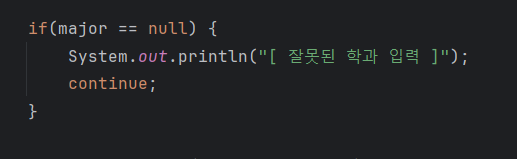
정규 표현식 패턴에 일치하지 않으면 null 반환

**학과 예외처리**

**public String checkMajor(Scanner scan)**







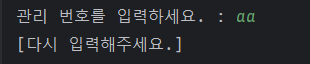
     

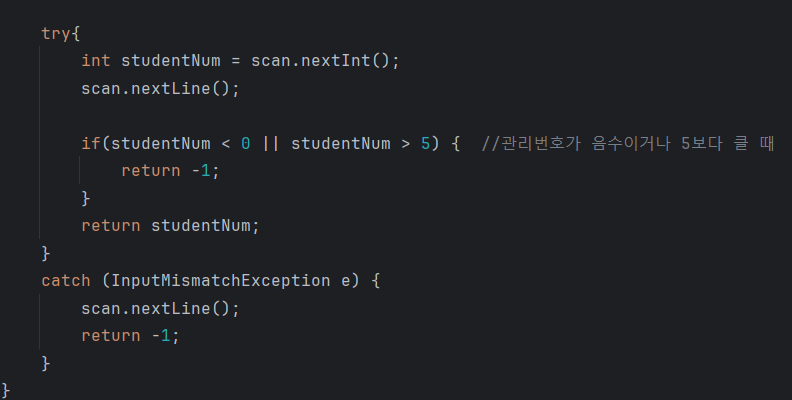
 

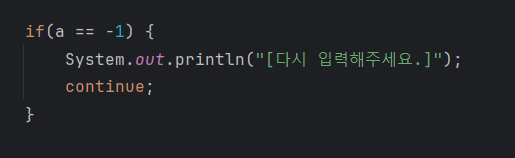
정규 표현식의 패턴과 일치하는 입력값이 아니면 null 반환

**관리번호 예외처리**

**public int checkNum(Scanner scan)**







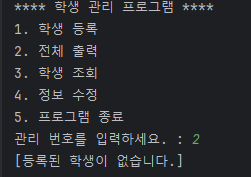
 

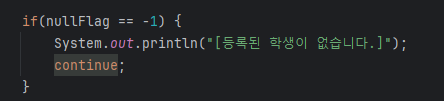
관리번호가 음수이거나 5보다 클 때 -1 을 반환하고

관리번호(int 타입)와 다른 타입이 입력되었을 때 -1 반환

**등록된 학생이 없을 때 예외처리**

**public int checkNull(Student[] student)**





(Student 타입) 학생 배열을 매개변수로 받습니다. 받은 배열을 advanced for문을 통해

전체 배열을 탐색합니다. 배열 안에 한 명이라도 저장되어 있다면 nullFlag에 1을 저장합니다.

최종적으로 return nullFlag 를 합니다.

**Reference**

교재 pdf

<https://moonong.tistory.com/31>

<https://inpa.tistory.com/entry/JAVA-%E2%98%95-%EC%A0%95%EA%B7%9C%EC%8B%9DRegular-Expression-%EC%82%AC%EC%9A%A9%EB%B2%95-%EC%A0%95%EB%A6%AC>