

1주차 김경민님 코드리뷰

1. Student 클래스의 getter 메서드명

```
class Student:
    def __init__(self, id, name, age, subject, grade):
        self.__id = id
        self.__name = name
        self.__age = age
        self.__subject = subject
        self.__grade = grade

    def getter_id(self):
        return self.__id

    def getter_name(self):
        return self.__name

    def getter_age(self):
        return self.__age

    def getter_subject(self):
        return self.__subject

    def getter_grade(self):
        return self.__grade
```

- 일반적으로 getter 메서드명은 get_xxx로 사용함
- 참고 : <https://thrillfighter.tistory.com/738>

2. StudentManagerRepo 클래스와 StudentManagerService 클래스의 메서드 매개변수명 변경

```
class StudentManagerRepo:
    @abstractmethod
    def add_student(self, student): # 학생 추가
        pass

    @abstractmethod
    def list_student(self): # 전체 학생 조회
        pass

    @abstractmethod
    def search_student(self, name): # 학생 조회
        pass

    @abstractmethod
    def delete_student(self, name): # 학생 제거
        pass

    @abstractmethod
    def update_student(self, name, student): # 학생 수정
        pass
```

```
class StudentManagerService:
    def __init__(self):
        self.__student_repo = StudentManagerImpl()

    def add_student(self, student): # 학생 추가
        self.__student_repo.add_student(student)

    def list_student(self): # 전체 학생 조회
        return self.__student_repo.list_student()

    def search_student(self, name): # 학생 조회
        return self.__student_repo.search_student(name)

    def delete_student(self, name): # 학생 제거
        self.__student_repo.delete_student(name)

    def update_student(self, name, student): # 학생 수정
        self.__student_repo.update_student(name, student)
```

- 경민님의 **StudentManagerImpl**의 `search_student`와 `delete_student`, `update_student`에서 학번(id)을 통해 메서드의 기능을 실행하므로 **StudentManagerRepo** 클래스와 **StudentManagerService** 클래스의 메서드의 매개변수도 `name` → `id`로 변경하면 좋을 것 같음

```
def search_student(self, id):
    if id in self.__students:
        return self.__students[id]
    else:
        return None

def delete_student(self, id):
    if id in self.__students:
        del self.__students[id]
    else:
        raise ValueError("해당 학번의 학생을 찾을 수 없습니다.")

def update_student(self, id, updated_student):
    if id in self.__students:
        self.__students[id] = updated_student
    else:
        raise ValueError("해당 학번의 학생을 찾을 수 없습니다.")
```

3. `print_student` 함수 대신 다음에는 `__string__` 메서드를 오버라이딩 해보셔도 좋을 듯

```
# 학생 정보 출력
def print_student(student):
    print(f"학번: {student.getter_id()}, 이름: {student.getter_name()}, 나이: {student.getter_age()}, 전공: {student.getter_subject()}, 학점: {student.g
```

4. **StudentManagerImpl** 클래스의 `update_student` 메서드

```
def update_student(self, id, updated_student):
    if id in self.__students:
        self.__students[id] = updated_student
    else:
        raise ValueError("해당 학번의 학생을 찾을 수 없습니다.")
```

- a. dictionary key는 기존 학번 그대로지만, value는 새로 업데이트된 학번일 것 같음
⇒ search 시 문제가 생길 것 같은데?
⇒ update 시 이미 key가 존재하는 지 확인(있을 경우 변경 불가) 후, key를 변경하는 방법이 좋을 것 같음

