자료 구조 실습 과제

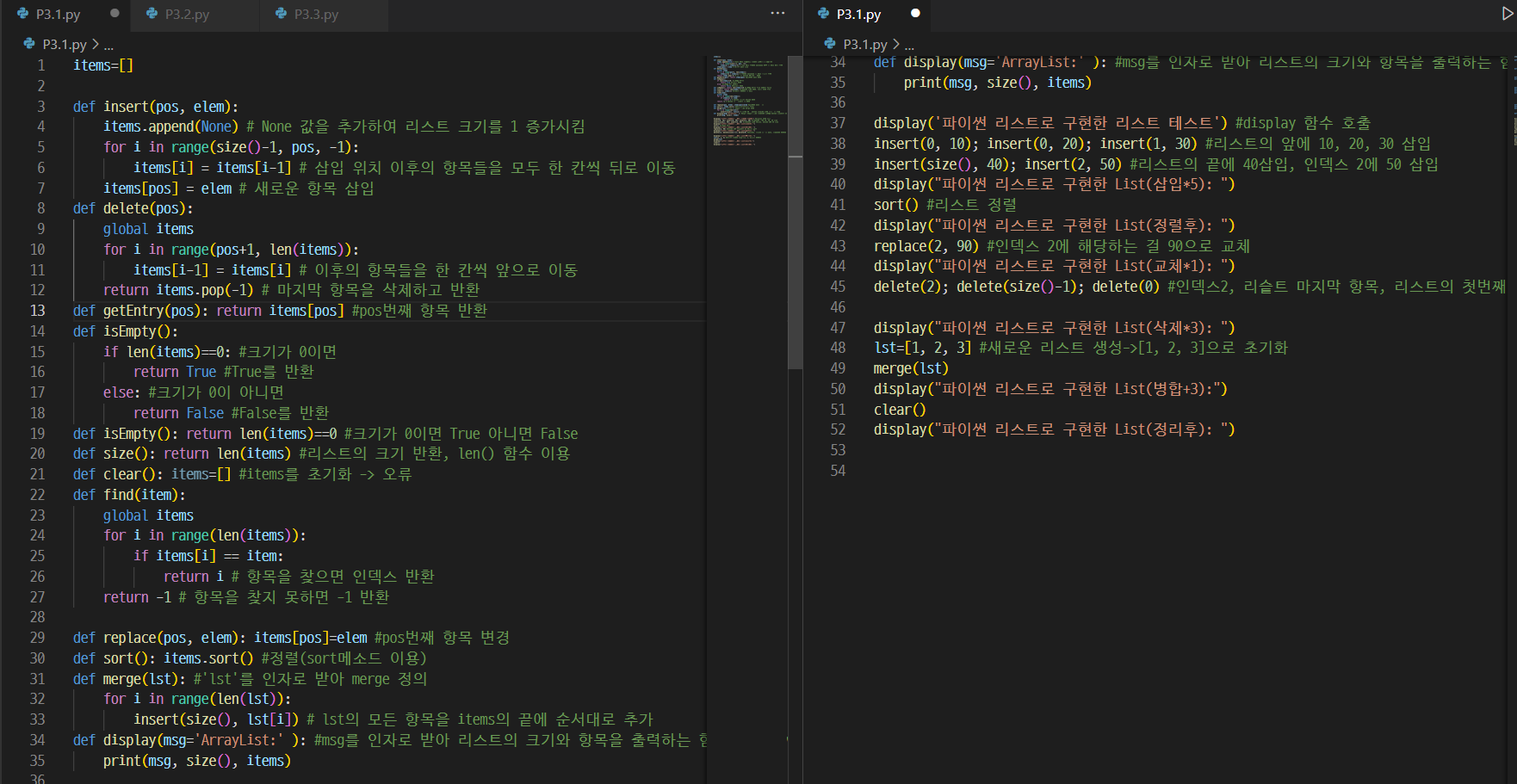
분반: 503분반

학번: C293066

학과: 캠퍼스 자율 전공

이름: 김채민

P3.1



items=[]

def insert(pos, elem):

    items.append(None) # None 값을 추가하여 리스트 크기를 1 증가시킴

    for i in range(size()-1, pos, -1):

        items[i] = items[i-1] # 삽입 위치 이후의 항목들을 모두 한 칸씩 뒤로 이동

    items[pos] = elem # 새로운 항목 삽입

def delete(pos):

    global items

    for i in range(pos+1, len(items)):

        items[i-1] = items[i] # 이후의 항목들을 한 칸씩 앞으로 이동

    return items.pop(-1) # 마지막 항목을 삭제하고 반환

def getEntry(pos): return items[pos] #pos번째 항목 반환

def isEmpty():

    if len(items)==0: #크기가 0이면

        return True #True를 반환

    else: #크기가 0이 아니면

        return False #False를 반환

def isEmpty(): return len(items)==0 #크기가 0이면 True 아니면 False

def size(): return len(items) #리스트의 크기 반환, len() 함수 이용

def clear(): items=[] #items를 초기화 -> 오류

def find(item):

    global items

    for i in range(len(items)):

        if items[i] == item:

            return i # 항목을 찾으면 인덱스 반환

    return -1 # 항목을 찾지 못하면 -1 반환

def replace(pos, elem): items[pos]=elem #pos번째 항목 변경

def sort(): items.sort() #정렬(sort메소드 이용)

def merge(lst): #'lst'를 인자로 받아 merge 정의

    for i in range(len(lst)):

        insert(size(), lst[i]) # lst의 모든 항목을 items의 끝에 순서대로 추가

def display(msg='ArrayList:' ): #msg를 인자로 받아 리스트의 크기와 항목을 출력하는 함수 display 정의

    print(msg, size(), items)

display('파이썬 리스트로 구현한 리스트 테스트') #display 함수 호출

insert(0, 10); insert(0, 20); insert(1, 30) #리스트의 앞에 10, 20, 30 삽입

insert(size(), 40); insert(2, 50) #리스트의 끝에 40삽입, 인덱스 2에 50 삽입

display("파이썬 리스트로 구현한 List(삽입\*5): ")

sort() #리스트 정렬

display("파이썬 리스트로 구현한 List(정렬후): ")

replace(2, 90) #인덱스 2에 해당하는 걸 90으로 교체

display("파이썬 리스트로 구현한 List(교체\*1): ")

delete(2); delete(size()-1); delete(0) #인덱스2, 리슽트 마지막 항목, 리스트의 첫번째 항목 삭제

display("파이썬 리스트로 구현한 List(삭제\*3): ")

lst=[1, 2, 3] #새로운 리스트 생성->[1, 2, 3]으로 초기화

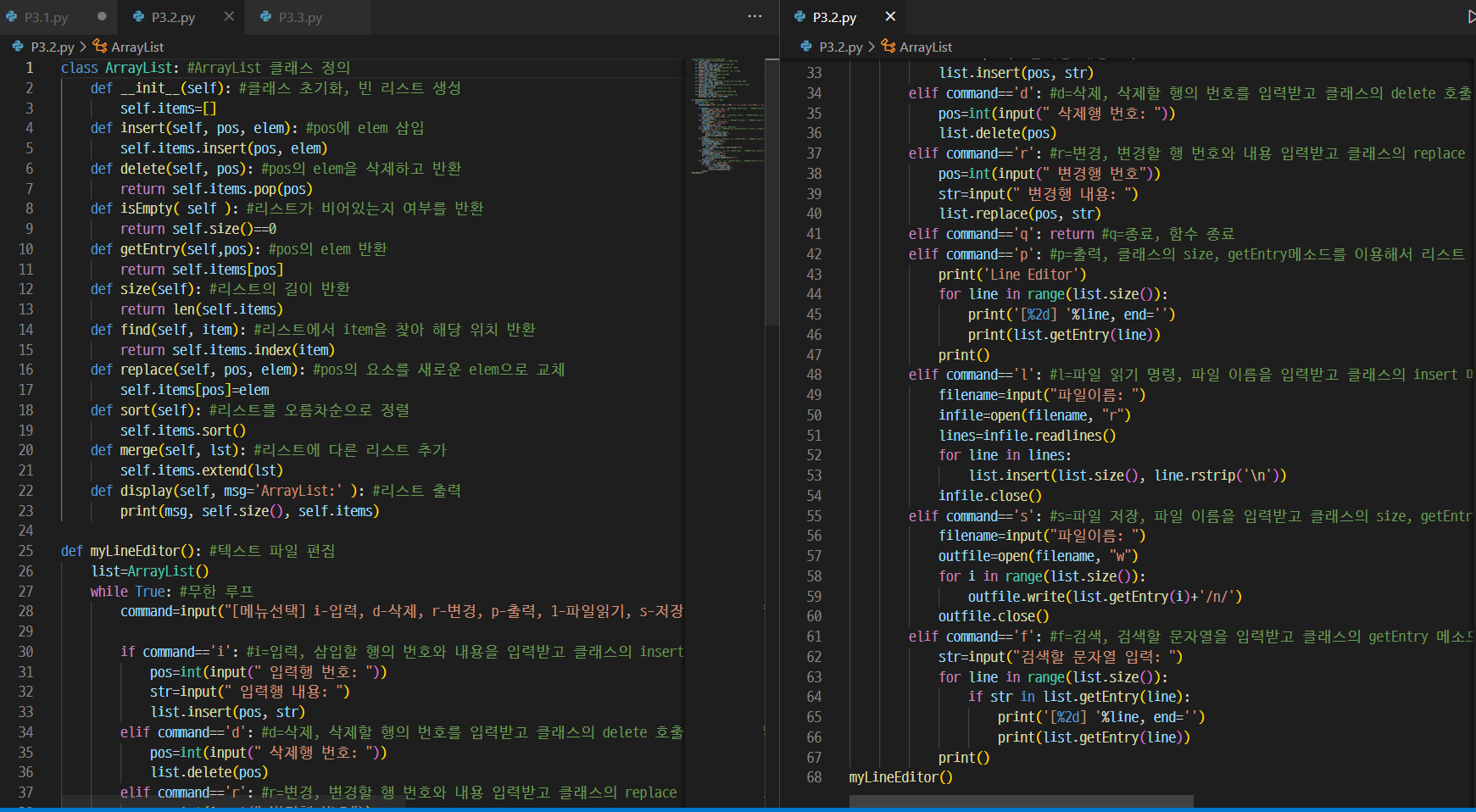
merge(lst)

display("파이썬 리스트로 구현한 List(병합+3):")

clear()

display("파이썬 리스트로 구현한 List(정리후): ")

P3.2



class ArrayList: #ArrayList 클래스 정의

    def \_\_init\_\_(self): #클래스 초기화, 빈 리스트 생성

        self.items=[]

    def insert(self, pos, elem): #pos에 elem 삽입

        self.items.insert(pos, elem)

    def delete(self, pos): #pos의 elem을 삭제하고 반환

        return self.items.pop(pos)

    def isEmpty( self ): #리스트가 비어있는지 여부를 반환

        return self.size()==0

    def getEntry(self,pos): #pos의 elem 반환

        return self.items[pos]

    def size(self): #리스트의 길이 반환

        return len(self.items)

    def find(self, item): #리스트에서 item을 찾아 해당 위치 반환

        return self.items.index(item)

    def replace(self, pos, elem): #pos의 요소를 새로운 elem으로 교체

        self.items[pos]=elem

    def sort(self): #리스트를 오름차순으로 정렬

        self.items.sort()

    def merge(self, lst): #리스트에 다른 리스트 추가

        self.items.extend(lst)

    def display(self, msg='ArrayList:' ): #리스트 출력

        print(msg, self.size(), self.items)

def myLineEditor(): #텍스트 파일 편집

    list=ArrayList()

    while True: #무한 루프

        command=input("[메뉴선택] i-입력, d-삭제, r-변경, p-출력, 1-파일읽기, s-저장, q-종료, f-문자열을 포함하고 있는 라인들만 출력=> ") #명령어 입력

        if command=='i': #i=입력, 삽입할 행의 번호와 내용을 입력받고 클래스의 insert호출해서ㅓ 내용 삽입

            pos=int(input(" 입력행 번호: "))

            str=input(" 입력행 내용: ")

            list.insert(pos, str)

        elif command=='d': #d=삭제, 삭제할 행의 번호를 입력받고 클래스의 delete 호출해서 내용 삭제

            pos=int(input(" 삭제행 번호: "))

            list.delete(pos)

        elif command=='r': #r=변경, 변경할 행 번호와 내용 입력받고 클래스의 replace 메소드를 호출해서 새로운 값으로 변경

            pos=int(input(" 변경행 번호"))

            str=input(" 변경행 내용: ")

            list.replace(pos, str)

        elif command=='q': return #q=종료, 함수 종료

        elif command=='p': #p=출력, 클래스의 size, getEntry메소드를 이용해서 리스트 전체 출력

            print('Line Editor')

            for line in range(list.size()):

                print('[%2d] '%line, end='')

                print(list.getEntry(line))

            print()

        elif command=='l': #l=파일 읽기 명령, 파일 이름을 입력받고 클래스의 insert 메소드를 이용해 리스트에 추가

            filename=input("파일이름: ")

            infile=open(filename, "r")

            lines=infile.readlines()

            for line in lines:

                list.insert(list.size(), line.rstrip('\n'))

            infile.close()

        elif command=='s': #s=파일 저장, 파일 이름을 입력받고 클래스의 size, getEntry 메소드를 이용해 리스트 전체를 파일에 저장

            filename=input("파일이름: ")

            outfile=open(filename, "w")

            for i in range(list.size()):

                outfile.write(list.getEntry(i)+'/n/')

            outfile.close()

        elif command=='f': #f=검색, 검색할 문자열을 입력받고 클래스의 getEntry 메소드를 이용해 해당 문자열을 포함하는 모든 행 출력

            str=input("검색할 문자열 입력: ")

            for line in range(list.size()):

                if str in list.getEntry(line):

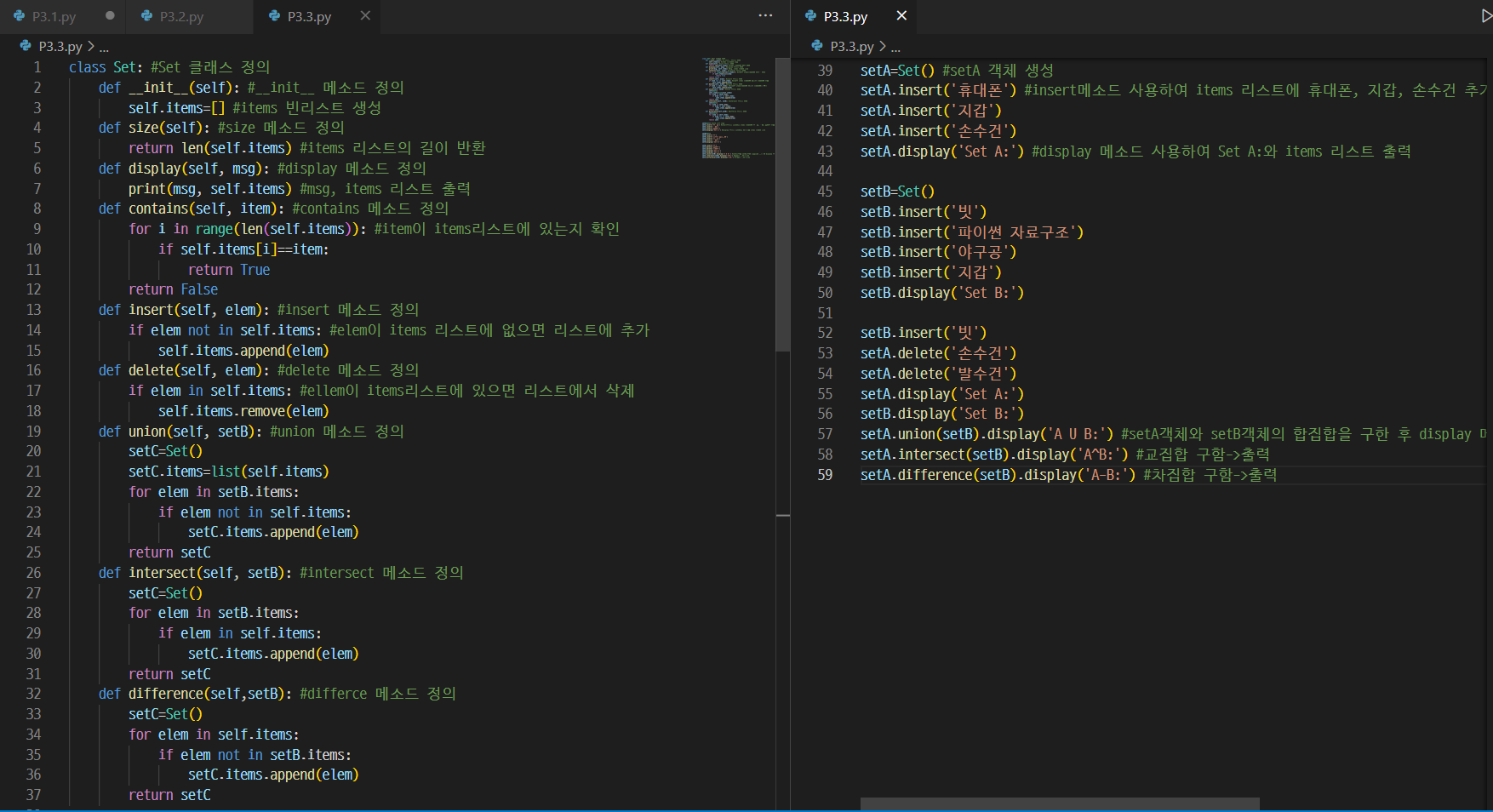
                    print('[%2d] '%line, end='')

                    print(list.getEntry(line))

            print()

myLineEditor()

P3.3



class Set: #Set 클래스 정의

    def \_\_init\_\_(self): #\_\_init\_\_ 메소드 정의

        self.items=[] #items 빈리스트 생성

    def size(self): #size 메소드 정의

        return len(self.items) #items 리스트의 길이 반환

    def display(self, msg): #display 메소드 정의

        print(msg, self.items) #msg, items 리스트 출력

    def contains(self, item): #contains 메소드 정의

        for i in range(len(self.items)): #item이 items리스트에 있는지 확인

            if self.items[i]==item:

                return True

        return False

    def insert(self, elem): #insert 메소드 정의

        if elem not in self.items: #elem이 items 리스트에 없으면 리스트에 추가

            self.items.append(elem)

    def delete(self, elem): #delete 메소드 정의

        if elem in self.items: #ellem이 items리스트에 있으면 리스트에서 삭제

            self.items.remove(elem)

    def union(self, setB): #union 메소드 정의

        setC=Set()

        setC.items=list(self.items)

        for elem in setB.items:

            if elem not in self.items:

                setC.items.append(elem)

        return setC

    def intersect(self, setB): #intersect 메소드 정의

        setC=Set()

        for elem in setB.items:

            if elem in self.items:

                setC.items.append(elem)

        return setC

    def difference(self,setB): #differce 메소드 정의

        setC=Set()

        for elem in self.items:

            if elem not in setB.items:

                setC.items.append(elem)

        return setC

setA=Set() #setA 객체 생성

setA.insert('휴대폰') #insert메소드 사용하여 items 리스트에 휴대폰, 지갑, 손수건 추가

setA.insert('지갑')

setA.insert('손수건')

setA.display('Set A:') #display 메소드 사용하여 Set A:와 items 리스트 출력

setB=Set()

setB.insert('빗')

setB.insert('파이썬 자료구조')

setB.insert('야구공')

setB.insert('지갑')

setB.display('Set B:')

setB.insert('빗')

setA.delete('손수건')

setA.delete('발수건')

setA.display('Set A:')

setB.display('Set B:')

setA.union(setB).display('A U B:') #setA객체와 setB객체의 합집합을 구한 후 display 메소드를 사용하여 A U B:와 items 리스트를 출력

setA.intersect(setB).display('A^B:') #교집합 구함->출력

setA.difference(setB).display('A-B:') #차집합 구함->출력