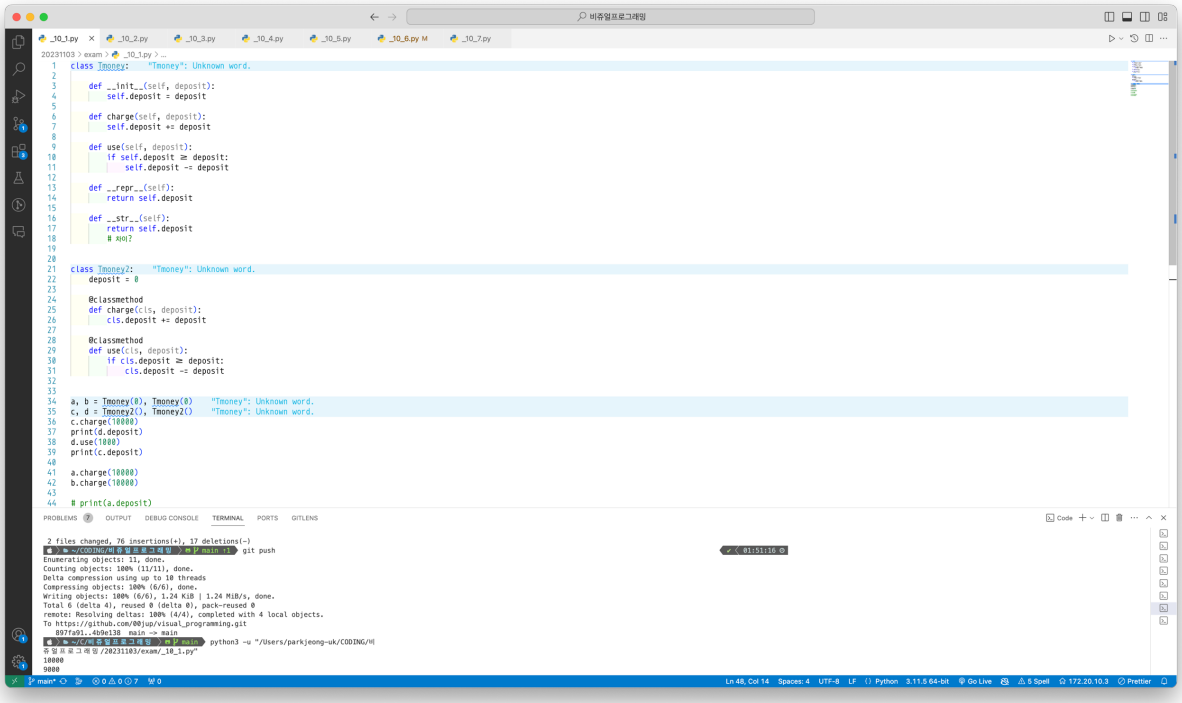
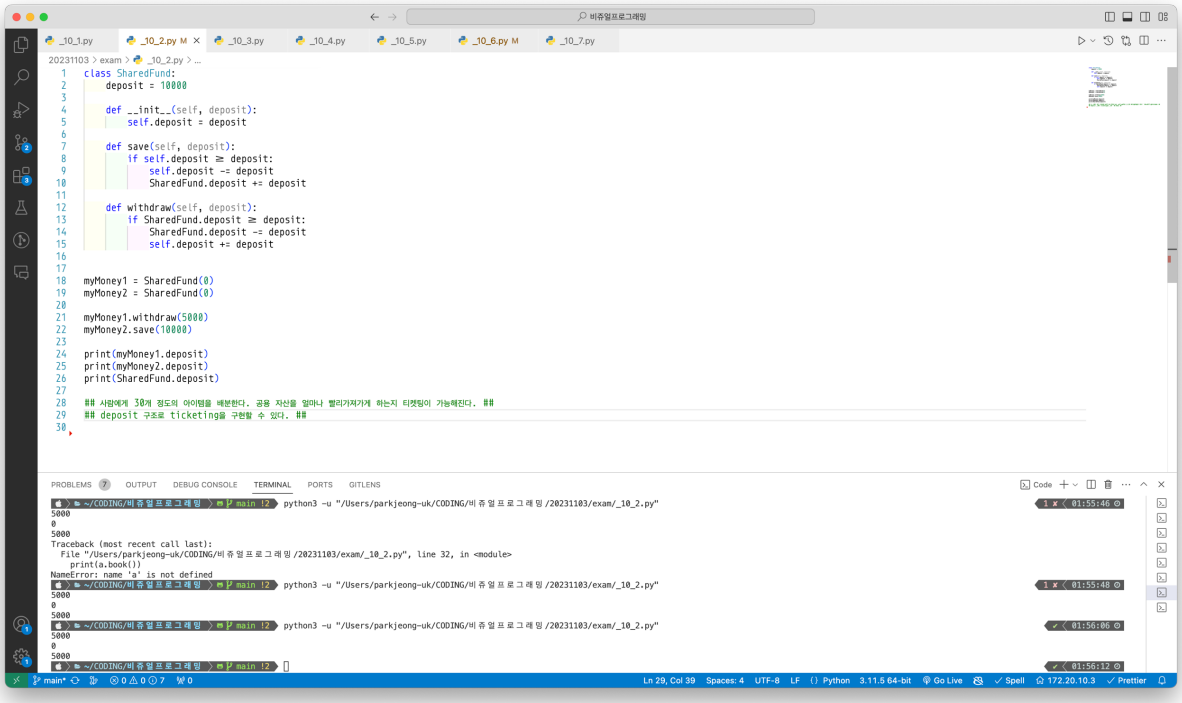
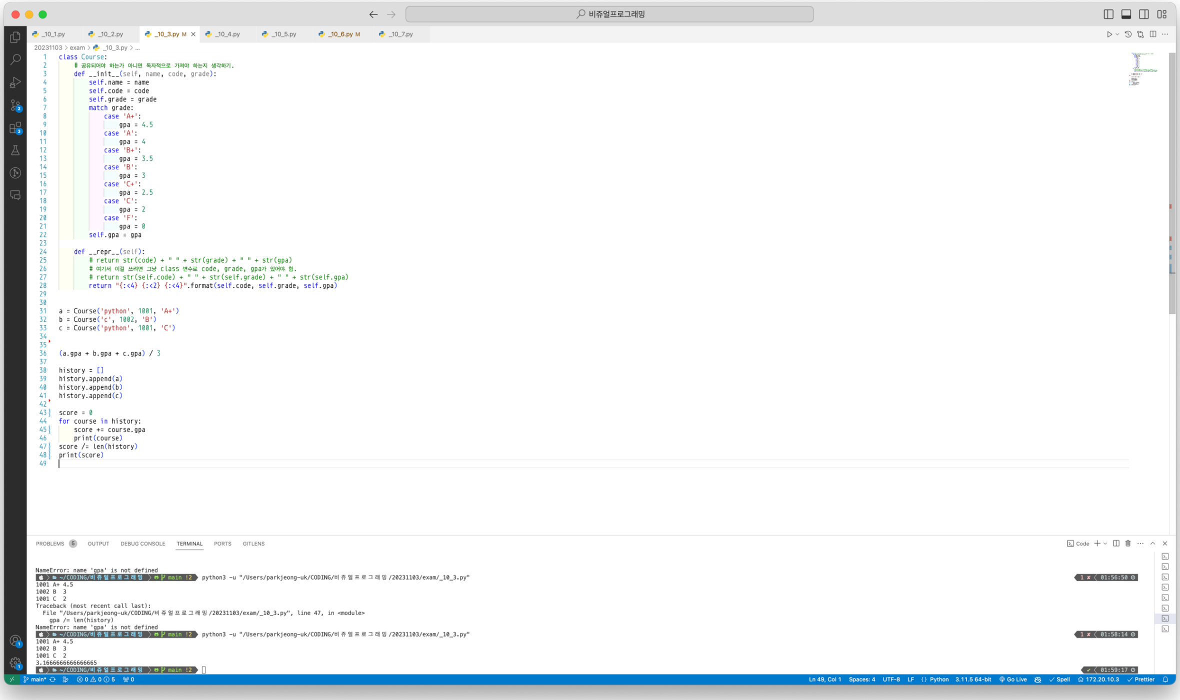

비주얼프로그래밍
_10_1.py
_10_2.py
_10_3.py
_10_4.py
_10_5.py
_10_6.py M
_10_7.py
20231103 > exam > _10_1.py > ...
class Tmoney: "Tmoney": Unknown word.
def __init__(self, deposit):
self.deposit = deposit
def charge(self, deposit):
self.deposit += deposit
def use(self, deposit):
if self.deposit >= deposit:
self.deposit -= deposit
def __repr__(self):
return self.deposit
def __str__(self):
return self.deposit
차이?
class Tmoney2:   "Tmoney": Unknown word.
deposit = 0
@classmethod
def charge(cls, deposit):
cls.deposit += deposit
@classmethod
def use(cls, deposit):
if cls.deposit >= deposit:
cls.deposit -= deposit
a, b = Tmoney(0), Tmoney(0)   "Tmoney": Unknown word.
c, d = Tmoney2(), Tmoney2()   "Tmoney": Unknown word.
c.charge(10000)
print(d.deposit)
d.use(1000)
print(c.deposit)
a.charge(10000)
b.charge(10000)
# print(a.deposit)
PROBLEMS 7 OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS
Code
2 files changed, 76 insertions(+), 17 deletions(-)
~/CODING/Ml 유일프로그래밍 main +1 git push
01:51:16
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 10 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 1.24 KiB | 1.24 MiB/s, done.
Total 6 (delta 4), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (4/4), completed with 4 local objects.
To https://github.com/00jup/visual_programming.git
897fa91..4b9e138 main -> main
python3 -u "/Users/parkjeong-uk/CODING/Ml
유일프로그래밍/20231103/exam/_10_1.py"
10000
9000
main*
0 0 7
0
Ln 48, Col 14 Spaces: 4 UTF-8 LF Python 3.11.5 64-bit Go Live 5 Spell 172.20.10.3 Prettier

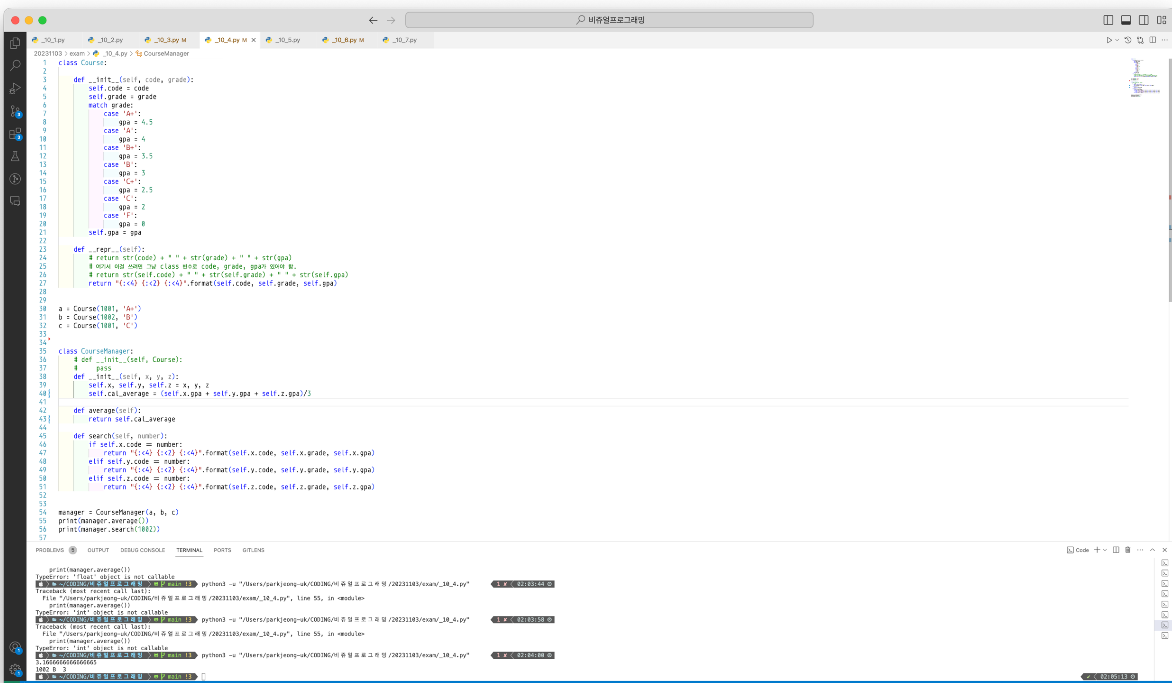

비쥬얼프로그래밍
_10_1.py
_10_2.py M
_10_3.py
_10_4.py
_10_5.py
_10_6.py M
_10_7.py
20231103 > exam > _10_2.py > ...
class SharedFund:
deposit = 10000
def __init__(self, deposit):
self.deposit = deposit
def save(self, deposit):
if self.deposit ≥ deposit:
self.deposit -= deposit
SharedFund.deposit += deposit
def withdraw(self, deposit):
if SharedFund.deposit ≥ deposit:
SharedFund.deposit -= deposit
self.deposit += deposit
myMoney1 = SharedFund(0)
myMoney2 = SharedFund(0)
myMoney1.withdraw(5000)
myMoney2.save(10000)
print(myMoney1.deposit)
print(myMoney2.deposit)
print(SharedFund.deposit)
사람에게 30개 정도의 아이템을 배분한다. 공용 자산을 얼마나 빌리거져거게 하는지 티켓팅이 가능해진다.
deposit 구조로 ticketing을 구현할 수 있다.
PROBLEMS 7
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
GITLENS
Code
~/CODING/비쥬얼프로그래밍 main !2 python3 -u "/Users/parkjeong-uk/CODING/비쥬얼프로그래밍/20231103/exam/_10_2.py"
5000
0
5000
Traceback (most recent call last):
File "/Users/parkjeong-uk/CODING/비쥬얼프로그래밍/20231103/exam/_10_2.py", line 32, in <module>
print(a.book())
NameError: name 'a' is not defined
~/CODING/비쥬얼프로그래밍 main !2 python3 -u "/Users/parkjeong-uk/CODING/비쥬얼프로그래밍/20231103/exam/_10_2.py"
5000
0
5000
~/CODING/비쥬얼프로그래밍 main !2 python3 -u "/Users/parkjeong-uk/CODING/비쥬얼프로그래밍/20231103/exam/_10_2.py"
5000
0
5000
~/CODING/비쥬얼프로그래밍 main !2
main*
Ln 29, Col 39
Spaces: 4
UTF-8
LF
Python
3.11.5 64-bit
Go Live
Spell
172.20.10.3
Prettier



_10_3.py

```python
class Course:
    # 공유되어야 하는가 아니면 독자적으로 가져야 하는지 생각하기.
    def __init__(self, name, code, grade):
        self.name = name
        self.code = code
        self.grade = grade
        match grade:
            case 'A+':
                gpa = 4.5
            case 'A':
                gpa = 4
            case 'B+':
                gpa = 3.5
            case 'B':
                gpa = 3
            case 'C+':
                gpa = 2.5
            case 'C':
                gpa = 2
            case 'F':
                gpa = 0
        self.gpa = gpa

    def __repr__(self):
        # return str(code) + " " + str(grade) + " " + str(gpa)
        # 여기서 이걸 쓰려면 그냥 class 변수로 code, grade, gpa가 있어야 함.
        # return str(self.code) + " " + str(self.grade) + " " + str(self.gpa)
        return "{:<4} {:<2} {:<4}".format(self.code, self.grade, self.gpa)


a = Course('python', 1001, 'A+')
b = Course('c', 1002, 'B')
c = Course('python', 1001, 'C')


(a.gpa + b.gpa + c.gpa) / 3

history = []
history.append(a)
history.append(b)
history.append(c)

score = 0
for course in history:
    score += course.gpa
    print(course)
score /= len(history)
print(score)
```

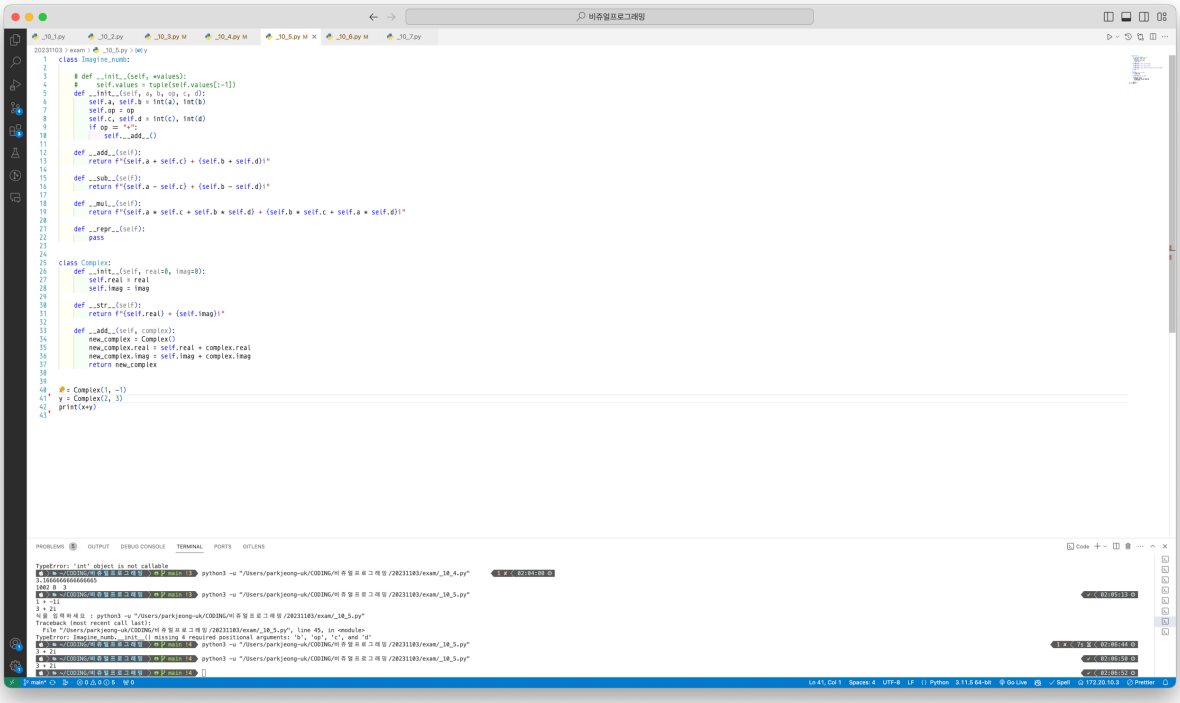
```
NameError: name 'gpa' is not defined
~/CODING/비쥬얼프로그래밍 main !2 python3 -u "/Users/parkjeong-uk/CODING/비쥬얼프로그래밍/20231103/exam/_10_3.py"
1001 A+ 4.5
1002 B  3
1001 C  2
Traceback (most recent call last):
  File "/Users/parkjeong-uk/CODING/비쥬얼프로그래밍/20231103/exam/_10_3.py", line 47, in <module>
    gpa /= len(history)
NameError: name 'gpa' is not defined
~/CODING/비쥬얼프로그래밍 main !2 python3 -u "/Users/parkjeong-uk/CODING/비쥬얼프로그래밍/20231103/exam/_10_3.py"
1001 A+ 4.5
1002 B  3
1001 C  2
3.1666666666666665
```

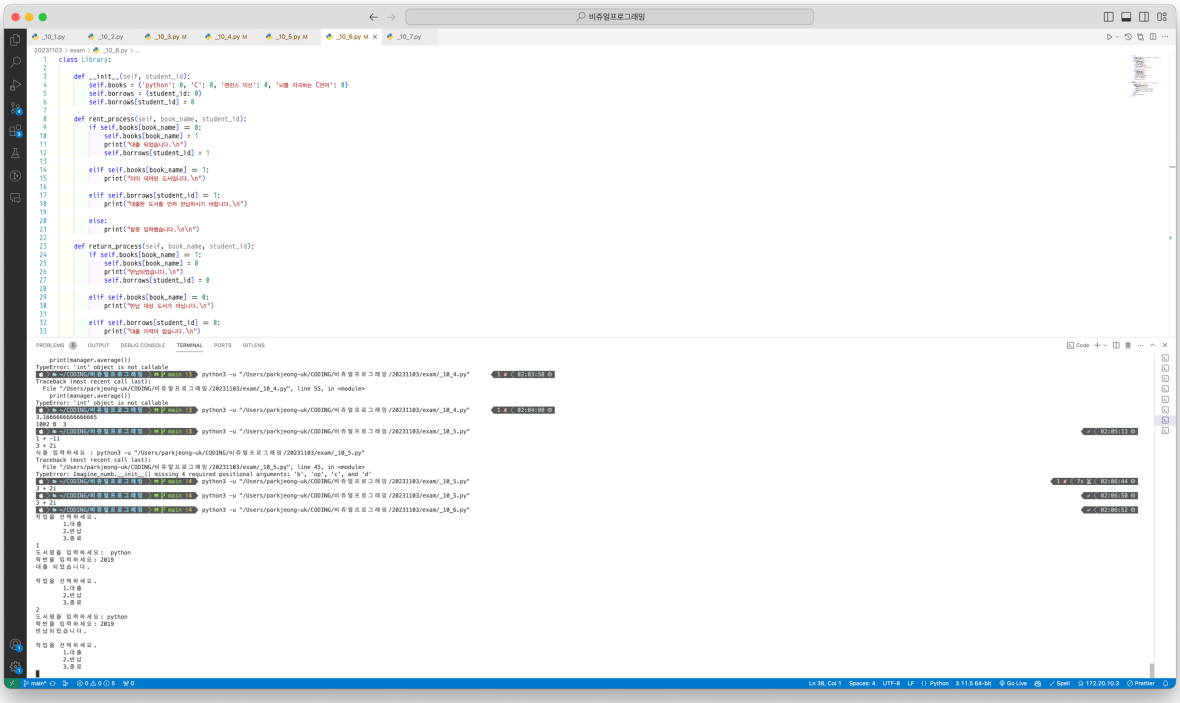


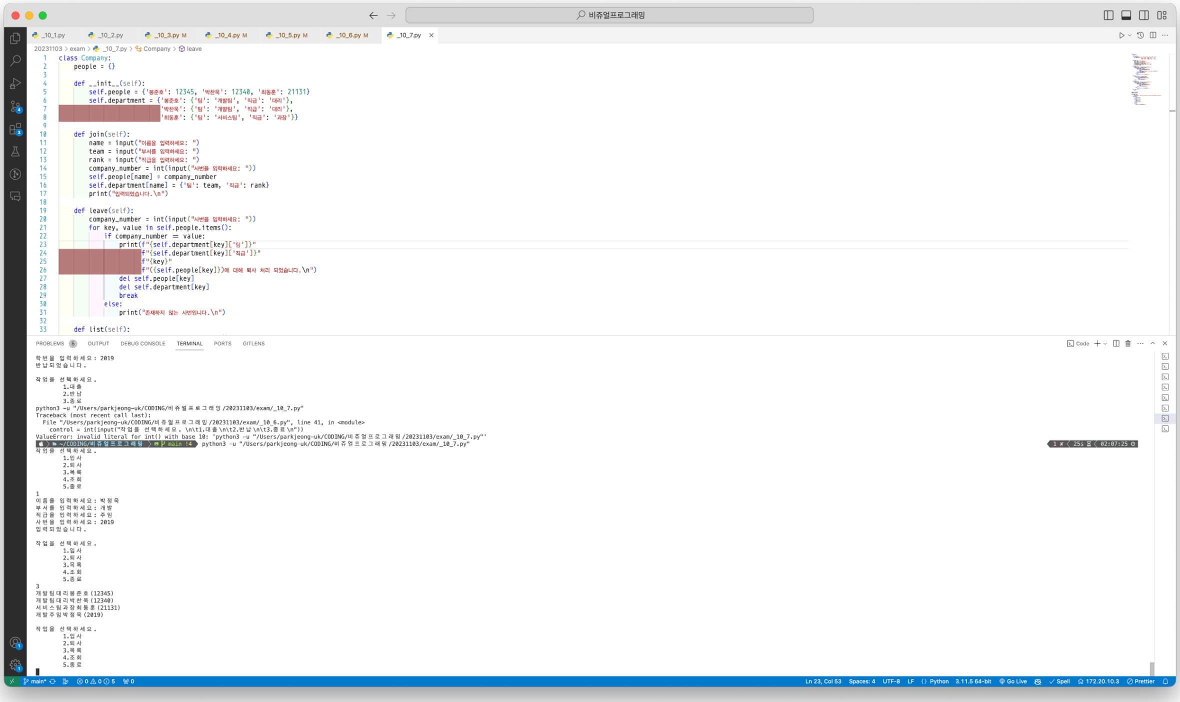
_10_4.py

```python
class Course:

    def __init__(self, code, grade):
        self.code = code
        self.grade = grade
        match grade:
            case 'A+':
                gpa = 4.5
            case 'A':
                gpa = 4
            case 'B+':
                gpa = 3.5
            case 'B':
                gpa = 3
            case 'C+':
                gpa = 2.5
            case 'C':
                gpa = 2
            case 'F':
                gpa = 0
        self.gpa = gpa

    def __repr__(self):
        # return str(code) + " " + str(grade) + " " + str(gpa)
        # 여기서 이걸 쓰려면 그냥 class 변수로 code, grade, gpa가 있어야 함.
        # return str(self.code) + " " + str(self.grade) + " " + str(self.gpa)
        return "{:<4} {:<2} {:<4}".format(self.code, self.grade, self.gpa)


a = Course(1001, 'A+')
b = Course(1002, 'B')
c = Course(1001, 'C')


class CourseManager:
    # def __init__(self, Course):
    #     pass
    def __init__(self, x, y, z):
        self.x, self.y, self.z = x, y, z
        self.cal_average = (self.x.gpa + self.y.gpa + self.z.gpa)/3

    def average(self):
        return self.cal_average

    def search(self, number):
        if self.x.code == number:
            return "{:<4} {:<2} {:<4}".format(self.x.code, self.x.grade, self.x.gpa)
        elif self.y.code == number:
            return "{:<4} {:<2} {:<4}".format(self.y.code, self.y.grade, self.y.gpa)
        elif self.z.code == number:
            return "{:<4} {:<2} {:<4}".format(self.z.code, self.z.grade, self.z.gpa)


manager = CourseManager(a, b, c)
print(manager.average())
print(manager.search(1002))
```

```
    print(manager.average())
TypeError: 'float' object is not callable
~/CODING/비쥬얼프로그래밍 main !3 python3 -u "/Users/parkjeong-uk/CODING/비쥬얼프로그래밍/20231103/exam/_10_4.py"   1 ✘ 02:03:44
Traceback (most recent call last):
  File "/Users/parkjeong-uk/CODING/비쥬얼프로그래밍/20231103/exam/_10_4.py", line 55, in <module>
    print(manager.average())
TypeError: 'int' object is not callable
~/CODING/비쥬얼프로그래밍 main !3 python3 -u "/Users/parkjeong-uk/CODING/비쥬얼프로그래밍/20231103/exam/_10_4.py"   1 ✘ 02:03:58
Traceback (most recent call last):
  File "/Users/parkjeong-uk/CODING/비쥬얼프로그래밍/20231103/exam/_10_4.py", line 55, in <module>
    print(manager.average())
TypeError: 'int' object is not callable
~/CODING/비쥬얼프로그래밍 main !3 python3 -u "/Users/parkjeong-uk/CODING/비쥬얼프로그래밍/20231103/exam/_10_4.py"   1 ✘ 02:04:00
3.1666666666666665
1002 B  3
```


비쥬얼프로그래밍
_10_1.py
_10_2.py
_10_3.py M
_10_4.py M
_10_5.py M
_10_6.py M
_10_7.py
20231103 > exam > _10_5.py > Imagine_numb
class Imagine_numb:
# def __init__(self, *values):
# self.values = tuple(self.values[:-1])
def __init__(self, a, b, op, c, d):
self.a, self.b = int(a), int(b)
self.op = op
self.c, self.d = int(c), int(d)
if op == "+":
self.__add__()
def __add__(self):
return f"{self.a + self.c} + {self.b + self.d}i"
def __sub__(self):
return f"{self.a - self.c} + {self.b - self.d}i"
def __mul__(self):
return f"{self.a * self.c + self.b * self.d} + {self.b * self.c + self.a * self.d}i"
def __repr__(self):
pass
class Complex:
def __init__(self, real=0, imag=0):
self.real = real
self.imag = imag
def __str__(self):
return f"{self.real} + {self.imag}i"
def __add__(self, complex):
new_complex = Complex()
new_complex.real = self.real + complex.real
new_complex.imag = self.imag + complex.imag
return new_complex
x = Complex(1, -1)
y = Complex(2, 3)
print(x+y)
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS
TypeError: 'int' object is not callable
3.1666666666666665
1002 0 3
1 + -11
3 + 21
식을 입력하세요 : python3 -u "/Users/parkjeong-uk/CODING/씨쥬얼프로그래밍/20231103/exam/_10_5.py"
Traceback (most recent call last):
File "/Users/parkjeong-uk/CODING/씨쥬얼프로그래밍/20231103/exam/_10_5.py", line 45, in <module>
TypeError: Imagine_numb.__init__() missing 4 required positional arguments: 'b', 'op', 'c', and 'd'
3 + 21
3 + 21
Ln 41, Col 1 Spaces: 4 UTF-8 LF Python 3.11.6 64-bit Go Live Spell 172.20.10.3 Prettier


비쥬얼프로그래밍
_10_1.py
_10_2.py
_10_3.py M
_10_4.py M
_10_5.py M
_10_6.py M
_10_7.py
20231103 > exam > _10_6.py > ...
class Library:
def __init__(self, student_id):
self.books = {'python': 0, 'C': 0, '멘탈스 미션': 0, '뇌를 자극하는 C언어': 0}
self.borrows = {student_id: 0}
self.borrows[student_id] = 0
def rent_process(self, book_name, student_id):
if self.books[book_name] == 0:
self.books[book_name] = 1
print("%s를 빌렸습니다.\n")
self.borrows[student_id] = 1
elif self.books[book_name] == 1:
print("이미 대여된 도서입니다.\n")
elif self.borrows[student_id] == 1:
print("%s을은 도서를 먼저 반납하시기 바랍니다.\n")
else:
print("잘못 입력했습니다.\n\n")
def return_process(self, book_name, student_id):
if self.books[book_name] == 1:
self.books[book_name] = 0
print("%s가반납되었습니다.\n")
self.borrows[student_id] = 0
elif self.books[book_name] == 0:
print("빈닙 대상 도서가 아닙니다.\n")
elif self.borrows[student_id] == 0:
print("%s를 이력이 없습니다.\n")
print(manager.average())
TypeError: 'int' object is not callable
Traceback (most recent call last):
File "/Users/parkjeong-uk/CODING/씨쥬얼프로그래밍/20231103/exam/_10_4.py", line 55, in <module>
print(manager.average())
TypeError: 'int' object is not callable
3.1666666666666665
1002 0 3
1 + -11
3 + 21
식을 입력하세요 : python3 -u "/Users/parkjeong-uk/CODING/씨쥬얼프로그래밍/20231103/exam/_10_5.py"
Traceback (most recent call last):
File "/Users/parkjeong-uk/CODING/씨쥬얼프로그래밍/20231103/exam/_10_5.py", line 45, in <module>
TypeError: Imagine_numb.__init__() missing 4 required positional arguments: 'b', 'op', 'c', and 'd'
3 + 21
3 + 21
3 + 21
작업을 선택하세요.
1.대출
2.반납
3.종료
1
도서명을 입력하세요 : python
작번을 입력하세요 : 2019
대출 되었습니다.
작업을 선택하세요.
1.대출
2.반납
3.종료
2
도서명을 입력하세요 : python
작번을 입력하세요 : 2019
반납되었습니다.
작업을 선택하세요.
1.대출
2.반납
3.종료
Ln 38, Col 1 Spaces: 4 UTF-8 LF Python 3.11.6 64-bit Go Live Spell 172.20.10.3 Prettier


비쥬얼프로그래밍
_10_1.py
_10_2.py
_10_3.py M
_10_4.py M
_10_5.py M
_10_6.py M
_10_7.py
20231103 > exam > _10_7.py > Company > leave
class Company:
    people = {}

    def __init__(self):
        self.people = {'봉준호': 12345, '박찬욱': 12340, '최동훈': 21131}
        self.department = {'봉준호': {'팀': '개발팀', '직급': '대리'},
                           '박찬욱': {'팀': '개발팀', '직급': '대리'},
                           '최동훈': {'팀': '서비스팀', '직급': '과장'}}

    def join(self):
        name = input("이름을 입력하세요: ")
        team = input("부서를 입력하세요: ")
        rank = input("직급을 입력하세요: ")
        company_number = int(input("사번을 입력하세요: "))
        self.people[name] = company_number
        self.department[name] = {'팀': team, '직급': rank}
        print("입력되었습니다.\n")

    def leave(self):
        company_number = int(input("사번을 입력하세요: "))
        for key, value in self.people.items():
            if company_number == value:
                print(f"{self.department[key]['팀']}"
                      f"{self.department[key]['직급']}"
                      f"{key}"
                      f"({self.people[key]})에 대해 퇴사 처리 되었습니다.\n")
                del self.people[key]
                del self.department[key]
                break
            else:
                print("존재하지 않는 사번입니다.\n")

    def list(self):
PROBLEMS 5
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
GITLENS
학번을 입력하세요: 2019
반납되었습니다.

작업을 선택하세요.
        1.대출
        2.반납
        3.종료
python3 -u "/Users/parkjeong-uk/CODING/비쥬얼프로그래밍/20231103/exam/_10_7.py"
Traceback (most recent call last):
  File "/Users/parkjeong-uk/CODING/비쥬얼프로그래밍/20231103/exam/_10_6.py", line 41, in <module>
    control = int(input("작업을 선택하세요. \n\t1.대출\n\t2.반납\n\t3.종료\n"))
ValueError: invalid literal for int() with base 10: 'python3 -u "/Users/parkjeong-uk/CODING/비쥬얼프로그래밍/20231103/exam/_10_7.py"'
~/CODING/비쥬얼프로그래밍 main !4 python3 -u "/Users/parkjeong-uk/CODING/비쥬얼프로그래밍/20231103/exam/_10_7.py"
작업을 선택하세요.
        1.입사
        2.퇴사
        3.목록
        4.조회
        5.종료
1
이름을 입력하세요: 박정욱
부서를 입력하세요: 개발
직급을 입력하세요: 주임
사번을 입력하세요: 2019
입력되었습니다.

작업을 선택하세요.
        1.입사
        2.퇴사
        3.목록
        4.조회
        5.종료
3
개발팀대리봉준호(12345)
개발팀대리박찬욱(12340)
서비스팀과장최동훈(21131)
개발주임박정욱(2019)

작업을 선택하세요.
        1.입사
        2.퇴사
        3.목록
        4.조회
        5.종료
main*
Ln 23, Col 53
Spaces: 4
UTF-8
LF
Python
3.11.5 64-bit
Go Live
Spell
Prettier