Bài 1,2

from datetime import datetime

class SinhVien:

    truong = "Dai hoc Da Lat"

    def \_\_init\_\_(self, maSo: int, hoTen: str, ngaySinh: datetime) :

        self.\_maSo=maSo

        self.\_hoTen=hoTen

        self.\_ngaySinh=ngaySinh

    @property

    def hoTen(self):

        return self.\_hoTen

    @hoTen.setter

    def hoTen(self, hoTen: str):

        self.\_hoTen = hoTen

    @property

    def maSo(self):

        return self.\_maSo

    @maSo.setter

    def maSo(self, maso):

        if self.laMaSoHopLe(maso):

            self.\_maSo=maso

    @staticmethod

    def laMaSoHopLe(maso: int):

        return len(str(maso))==7

    @classmethod

    def doiTenTruong(self, tenmoi):

        self.truong=tenmoi

    def \_\_str\_\_(self) -> str:

        return f"{self.\_maSo}\t{self.\_hoTen}\t{self.\_ngaySinh}"

    def xuat(self):

        print(f"{self.\_maSo}\t{self.\_hoTen}\t{self.\_ngaySinh}")

class DanhSachSv:

    def \_\_init\_\_(self) -> None:

        self.dssv = []

    def themSinhVien(self, sv: SinhVien):

        self.dssv.append(sv)

    def xuat(self):

        for sv in self.dssv:

            print(sv)

    def timSvTheoMSSV(self, maSo: int):

        return [sv for sv in self.dssv if sv.maSo == maSo]

    def timVTSvTheoMSSV(self, maSo:int):

        for i in range(len(self.dssv)):

            if self.dssv[i].maSo == maSo:

                return i

        return -1

    def xoaSvTheoMSSV(self, maSo: int)->bool:

        vt = self.timVTSvTheoMSSV(maSo)

        if vt != -1:

            del self.dssv[vt]

            return True

        else:

            return False

    def timSvTheoTen(self, ten:str):

        return [sv for sv in self.dssv if (ten.upper() in sv.\_hoTen.upper())]

    def timSvSinhTruocNgay(self, ngay: datetime):

        return [sv for sv in self.dssv if sv.ngaySinh < ngay]

sv1 = SinhVien(2111113, "Nguyen Van A", datetime.strptime("01/01/2000","%d/%m/%Y"))

sv2 = SinhVien(2111114, "Nguyen Van B", datetime.strptime("11/11/2001","%d/%m/%Y"))

sv3 = SinhVien(2111115, "Nguyen Van C", datetime.strptime("12/12/2002","%d/%m/%Y"))

sv4 = SinhVien(2111116, "Nguyen Van D", datetime.strptime("10/10/2003","%d/%m/%Y"))

dssv = DanhSachSv()

dssv.themSinhVien(sv1)

dssv.themSinhVien(sv2)

dssv.themSinhVien(sv3)

dssv.themSinhVien(sv4)

dssv.xuat()

a = int(input('Nhập vào mã số muốn tìm kiếm: '))

kq = dssv.timSvTheoMSSV(a)

if dssv.timVTSvTheoMSSV(a) != -1:

    print(f'Đã tìm thấy sinh viên có mã số {a}')

    for sv in kq:

        sv.xuat()

else:

    print(f'Không tìm thấy sinh viên có mã số {a}')

b = int(input('Nhập vào mã số muốn xóa: '))

kq = dssv.xoaSvTheoMSSV(b)

print(f'\nĐã xóa {b} khỏi danh sách\n')

dssv.xuat()

c = str(input('Nhập vào tên sinh viên cần tìm: '))

kq = dssv.timSvTheoTen(c)

print(f'\nĐã tìm thấy tên {c}')

for sv in kq:

    sv.xuat()

d = input('Nhập vào ngày sinh cần tìm:')

ngay = datetime.strptime(d, '%d/%m/%Y')

kqNgay = dssv.timSvSinhTruocNgay(ngay)

print(f'\nĐã tìm thấy ngày sih trước {d}:')

for sv in kqNgay:

    sv.xuat()

Bài 3

import math

class PhanSo:

    def \_\_init\_\_(self, tu\_so=1, mau\_so=1):

        self.tu = tu\_so

        self.mau = mau\_so if mau\_so > 0 else 1

    def \_\_str\_\_(self) -> str:

        return f"{self.tu}/{self.mau}"

    def rutGon(self):

        ucln = math.gcd(self.tu, self.mau)

        self.tu //= ucln

        self.mau //= ucln

    def \_\_add\_\_(self, other):

        kq = PhanSo()

        if not isinstance(other,PhanSo):

            other = PhanSo(other)

        kq.tu = self.tu \* other.mau + self.mau \* other.tu

        kq.mau = self.mau \* other.mau

        kq.rutGon()

        return kq

    def \_\_sub\_\_(self, other):

        kq = PhanSo()

        if not isinstance(other,PhanSo):

            other = PhanSo(other)

        kq.tu = self.tu \* other.mau - self.mau \* other.tu

        kq.mau = self.mau \* other.mau

        kq.rutGon()

        return kq

    def \_\_mul\_\_(self, other):

        kq = PhanSo()

        if not isinstance(other,PhanSo):

            other = PhanSo(other)

        kq.tu = self.tu \* other.tu

        kq.mau = self.mau \* other.mau

        kq.rutGon()

        return kq

    def \_\_truediv\_\_(self, other):

        kq = PhanSo()

        if not isinstance(other,PhanSo):

            other = PhanSo(other)

        kq.tu = self.tu \* other.mau

        kq.mau = self.mau \* other.tu

        kq.rutGon()

        return kq

a = PhanSo()

a.tu = 2

a.mau = 3

b = PhanSo(3, 5)

print(f"{a} + {b} = {a + b}")

print(f"{a} - {b} = {a - b}")

print(f"{a} \* {b} = {a \* b}")

print(f"{a} / {b} = {a / b}")