BÀI TẬP THỰC HÀNH SỐ 2

LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG VỚI PYTHON

I. Nội dung lý thuyết cần học

- Lớp, đối tượng
- Thuộc tính của lớp và thuộc tính của thể hiện (class attribute và instance attribute
- Phạm vi truy cập (access modifiers)
- Phương thức của lớp, phương thức tĩnh
- Ghi đè toán tử, ghi đè hàm
- Kế thừa, đa hình

II. Bài tập

Bài 1: Cài đặt lớp Sinh viên và danh sách sinh viên như sau và hoàn thành các hàm còn trống (chưa có nội dung)

```python
class SinhVien:
    # biến của lớp, chung cho tất cả đối tượng thuộc lớp
    truong = "Đại học Đà Lạt"

    # Hàm khởi tạo, hàm tạo lập: khởi gán các thuộc tính của đối tượng
    def __init__(self, maSo: int, hoTen: str, ngaySinh: datetime) -> None:
        self.__maSo = maSo # thuộc tính private
        self.__hoTen = hoTen # thuộc tính private
        self.__ngaySinh = ngaySinh # thuộc tính private


    #cho phép truy xuất tới thuộc tính từ bên ngoài thông qua trường maSo
    @property
    def maSo(self):
        return self.__maSo

    # cho phép thay đổi giá trị thuộc tính maSo
    @maSo.setter
    def maSo(self, maso):
        if self.laMaSoHopLe(maso):
            self.__maSo = maso
```

```python
    # phương thức tĩnh: các phương thức không truy xuất gì đến thuộc tính, hành vi của lớp
    # những phương thức này không cần truyền tham số mặc định self
    # đây không phải là một hành vi (phương thức) của 1 đối tượng thuộc lớp
    @staticmethod
    def laMaSoHopLe(maso: int):
        return len(str(maso)) == 7

    # phương thức của lớp, chỉ truy xuất tới các biến thành viên của lớp
    # không truy xuất được các thuộc tính riêng của đối tượng
    @classmethod
    def doiTenTruong(self, tenmoi):
        self.truong = tenmoi


    # tương tự ghi đè phương thức toString()
    def __str__(self) -> str:
        return f"{self.__maSo}\t{self.__hoTen}\t{self.__ngaySinh}"

    # hành vi của đối tượng sinh viên
    def xuat(self):
        print(f"{self.__maSo}\t{self.__hoTen}\t{self.__ngaySinh}")
```

```python
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)

    # tìm sinh viên theo mssv, nếu có trả về sinh viên
    def timSvTheoMssv(self, mssv: int):
        return [ sv for sv in self.dssv if sv.mssv == mssv]

    # tìm sinh viên theo mssv, nếu có trả về vị trí của sinh viên trong danh sách
    def timVTSvTheoMssv(self, mssv: int):
        for i in range(len(self.dssv)):
            if self.dssv[i].mssv == mssv:
                return i
        return -1
```

```python
    # xóa sinh viên có mã số mssv, thông báo xóa dc hoặc ko
    def xoaSvTheoMssv(self, maSo: int) -> bool:
        vt = self.timVTSvTheoMssv(maSo)
        if vt != -1:
            del self.dssv[vt]
            return True
        else:
            return False

    # tìm sinh viên tên "Nam"
    def timSvTheoTen(self, ten: str):
        pass

    # tìm những sinh viên sinh trước 15/6/2000
    def timSvSinhTruocNgay(self, ngay: datetime):
        pass
```

Bài 2: Bổ sung phương thức:

1. Đọc danh sách sinh viên từ tập tin (txt/csv) (xem: https://www.w3schools.com/python/python_file_open.asp)
2. Sắp xếp danh sách sinh viên tăng/giảm theo họ tên

Bài 3: Cài đặt lớp phân số có phương thức rút gọn phân số, ghi đè toán tử +, -, *, / như sau:

```python
class PhanSo:
    def __init__(self) -> None:
        pass

    def rutGon(self):
        pass

    def __add__(self, other):
        pass

    def __sub__(self, other):
        pass

    def __mul__(self, other):
        pass
```

```python
    def __truediv__(self, other):
        pass


a = PhanSo()
a.tu= 2
a.mau = 3
b = PhanSo(3,5)
print(f"{a} + {b} = {a+b}") # 1/6 + 4/12 = 1/2
print(f"{a} - {b} = {a-b}")
print(f"{a} * {b} = {a*b}")
print(f"{a} / {b} = {a/b}")
```

Bài 4: Cài đặt lớp danh sách phân số, bổ sung các chức năng và kiểm tra kết quả:

1. Đếm số phân số âm trong mảng
2. Tìm phân số dương nhỏ nhất
3. Tìm tất cả vị trí của phân số x trong mảng
4. Tổng tất cả các phân số âm trong mảng

5. Xóa phân số x trong mảng
6. Xóa tất cả phân số có tử là x
7. Sắp xếp phân số theo chiều tăng, giảm; tăng theo mẫu, tử, giảm theo mẫu tử.

Bài 5: Sử dụng kế thừa

Tạo lớp ***Sinh viên*** (trong file ***sinh_vien.py***) và các lớp ***SinhVienChinhQuy*** (file ***sinh_vien_chinh_quy.py***) và lớp ***SinhVienPhiCQ*** (file ***sv_phi_chinh_quy.py***) kế thừa từ lớp SinhVien như sau:

```python
from datetime import datetime


class SinhVien:
    truong = "Đại học Đà Lạt"
    def __init__(self, maSo: int, hoTen: str, ngaySinh: datetime) -> None:
        self._maSo = maSo # khai báo kiểu truy xuất là protected
        self._hoTen = hoTen
        self._ngaySinh = ngaySinh

    @property
    def hoTen(self):
        return self._hoTen

    @hoTen.setter
    def hoTen(self, hoTen: str):
        self._hoTen = hoTen
```

```python
    @property
    def mssv(self):
        return self._maSo

    @mssv.setter
    def mssv(self, ms: int):
        if self.ktMsHopLe(ms):
            self._maSo = ms

    @staticmethod
    def ktMsHopLe(mssv: int):
        return len(str(mssv)) == 7


    def __str__(self) -> str:
        return f"{self._maSo}\t{self._hoTen}\t{self._ngaySinh}"
```

File ***sinh_vien_chinh_quy.py***

OOP > ViDuKeThua > sinh_vien_chinh_quy.py > ...

```python
from sinh_vien import SinhVien
from datetime import datetime

class SinhVienChinhQuy(SinhVien):
    def __init__(self, maSo: int, hoTen: str, ngaySinh: datetime, diemRL: int) -> None:
        super().__init__(maSo, hoTen, ngaySinh)
        self.diemRL = diemRL

    def __str__(self) -> str:
        return super().__str__() + f"\t{self.diemRL}"
```

File ***sv_phi_chinh_quy.py***

OOP > ViDuKeThua > sv_phi_chinh_quy.py > ...

```python
from sinh_vien import SinhVien
from datetime import datetime

class SinhVienPhiCQ(SinhVien):
    def __init__(self, maSo: int, hoTen: str, ngaySinh: datetime, trinhdo: str, tgdt: int) -> None:
        super().__init__(maSo, hoTen, ngaySinh)
        self.thoiGianDaoTao = tgdt
        self.trinhDo = trinhdo

    def __str__(self) -> str:
        return super().__str__() + f"\t{self.trinhDo}\t{self.thoiGianDaoTao}"
```

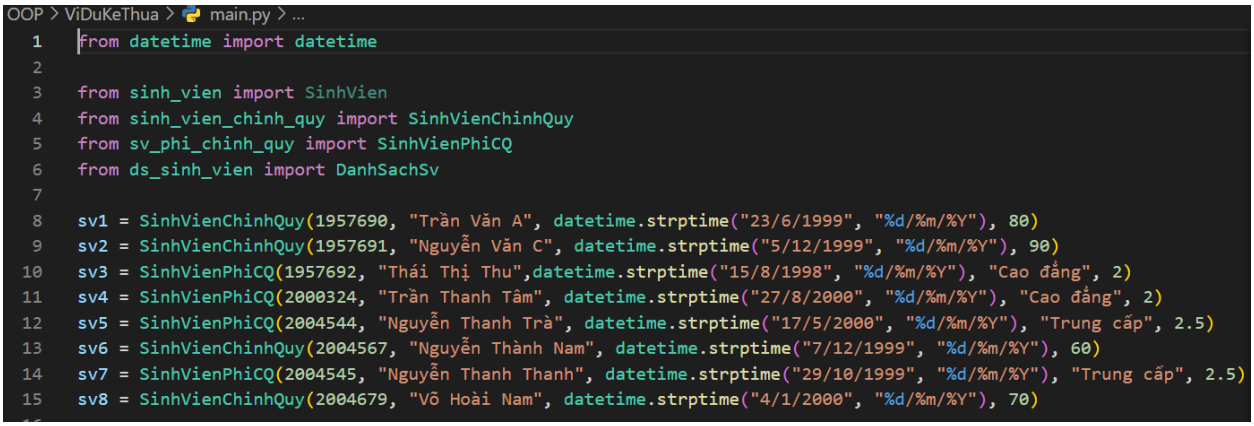
Tạo lớp danh sách sinh viên như sau:

OOP > ViDuKeThua > ds_sinh_vien.py > ...

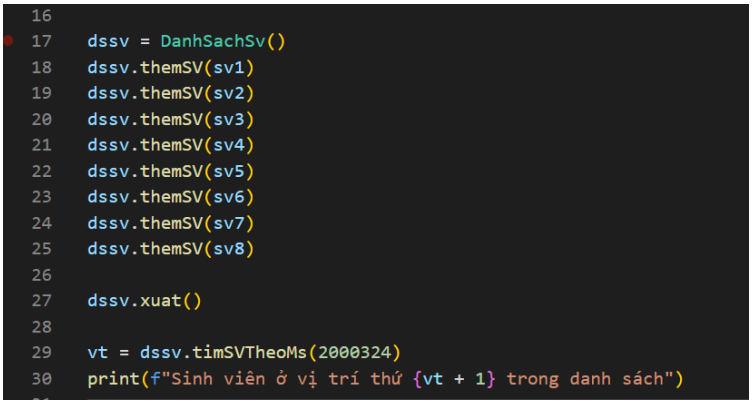
```python
from sinh_vien_chinh_quy import SinhVienChinhQuy
from sv_phi_chinh_quy import SinhVienPhiCQ
from sinh_vien import SinhVien


class DanhSachSv:
    def __init__(self) -> None:
        self.dssv = []

    def themSV(self, sv: SinhVien):
        self.dssv.append(sv)

    def xuat(self):
        for sv in self.dssv:
            print(sv)
```

```python
    def timSVTheoMs(self, ms: str):
        for i in range(len(self.dssv)):
            if self.dssv[i].mssv == ms:
                return i
        else:
            return -1

    def timSvTheoLoai(self, loai: str):
        if loai == "chinhquy":
            return [sv for sv in self.dssv if isinstance(sv, SinhVienChinhQuy)]
        return [sv for sv in self.dssv if isinstance(sv, SinhVienPhiCQ)]

    # tìm sinh viên có điểm rèn luyện từ 80 trở lên

    # tìm sinh viên có trình độ cao đẳng sinh trước 15/8/1999
```

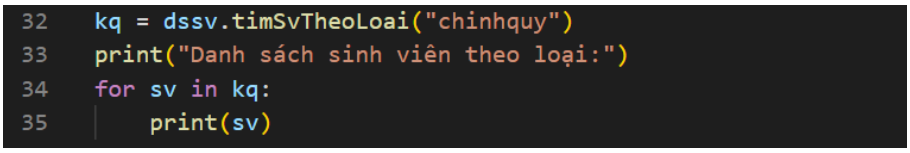
Tạo file ***main.py*** để tạo danh sách sinh viên và kiểm tra kết quả chạy:



```
OOP > ViDuKeThua > main.py > ...
1  from datetime import datetime
2
3  from sinh_vien import SinhVien
4  from sinh_vien_chinh_quy import SinhVienChinhQuy
5  from sv_phi_chinh_quy import SinhVienPhiCQ
6  from ds_sinh_vien import DanhSachSv
7
8  sv1 = SinhVienChinhQuy(1957690, "Trần Văn A", datetime.strptime("23/6/1999", "%d/%m/%Y"), 80)
9  sv2 = SinhVienChinhQuy(1957691, "Nguyễn Văn C", datetime.strptime("5/12/1999", "%d/%m/%Y"), 90)
10 sv3 = SinhVienPhiCQ(1957692, "Thái Thị Thu",datetime.strptime("15/8/1998", "%d/%m/%Y"), "Cao đẳng", 2)
11 sv4 = SinhVienPhiCQ(2000324, "Trần Thanh Tâm", datetime.strptime("27/8/2000", "%d/%m/%Y"), "Cao đẳng", 2)
12 sv5 = SinhVienPhiCQ(2004544, "Nguyễn Thanh Trà", datetime.strptime("17/5/2000", "%d/%m/%Y"), "Trung cấp", 2.5)
13 sv6 = SinhVienChinhQuy(2004567, "Nguyễn Thành Nam", datetime.strptime("7/12/1999", "%d/%m/%Y"), 60)
14 sv7 = SinhVienPhiCQ(2004545, "Nguyễn Thanh Thanh", datetime.strptime("29/10/1999", "%d/%m/%Y"), "Trung cấp", 2.5)
15 sv8 = SinhVienChinhQuy(2004679, "Võ Hoài Nam", datetime.strptime("4/1/2000", "%d/%m/%Y"), 70)
```



```
16
17 dssv = DanhSachSv()
18 dssv.themSV(sv1)
19 dssv.themSV(sv2)
20 dssv.themSV(sv3)
21 dssv.themSV(sv4)
22 dssv.themSV(sv5)
23 dssv.themSV(sv6)
24 dssv.themSV(sv7)
25 dssv.themSV(sv8)
26
27 dssv.xuat()
28
29 vt = dssv.timSVTheoMs(2000324)
30 print(f"Sinh viên ở vị trí thứ {vt + 1} trong danh sách")
```



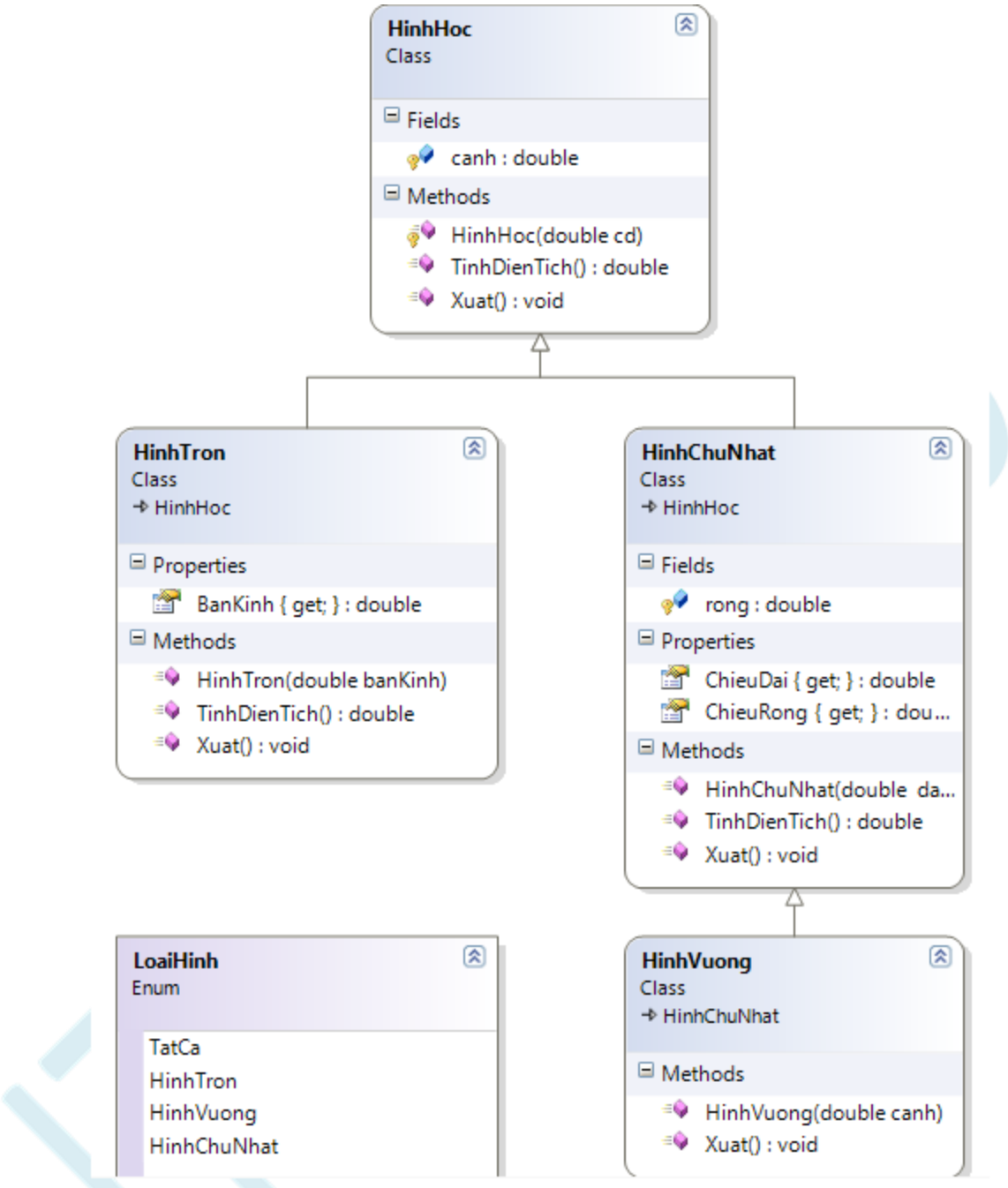
```
32 kq = dssv.timSvTheoLoai("chinhquy")
33 print("Danh sách sinh viên theo loại:")
34 for sv in kq:
35     print(sv)
```

Cài đặt 2 hàm còn thiếu trong danh sách sinh viên:

1. Tìm sinh viên có điểm rèn luyện từ 80 trở lên
2. Tìm sinh viên có trình độ cao đẳng sinh trước 15/8/1999

Bài 6: Cài đặt chương trình quản lý hình học theo mô hình lớp sau dùng Python

Mỗi lớp tạo trong 1 file .py



Xây dựng lớp ***DanhSachHinhHoc*** trong file ***ds_hinh_hoc.py*** gồm các chức năng sau:

Phương thức	**Ý nghĩa**
themHinh(hh: HinhHoc)	Thêm hình vào danh sách
xuat ()	Xuất danh sách hình
timHinhCoDienTichLonNhat() -> DanhSachHinhHoc	Tìm hình có diện tích lớn nhất
TimHinhCoDienTichNhoNhat() -> DanhSachHinhHoc	Tìm hình có diện tích nhỏ nhất

TimHinhTronNhoNhat() -> DanhSachHinhHoc	Tìm hình tròn có diện tích lớn nhất
SapGiamTheoDienTich() -> void	Sắp các hình giảm dần theo diện tích
DemSoLuongHinh(kieu: LoaiHinh) -> int	Đếm số lượng hình theo loại
TinhTongDienTich() -> float	Tính tổng diện tích các hình
TimHinhCoDienTichLonNhat (kieu: LoaiHinh) -> DanhSachHinhHoc	Tìm hình có diện tích lớn nhất theo loại hình học cho trước
TimViTriCuaHinh(h: HinhHoc) -> int	Tìm vị trí của hình h trong danh sách
XoaTaiViTri(viTri: int) -> bool	Xóa một hình tại vị trí cho trước
TimHinhTheoDTich(dt: float) -> DanhSachHinhHoc	Tìm hình theo diện tích
XoaHinh(hh: HinhHoc) -> bool	Xóa một hình học khỏi danh sách
XoaHinhTheoLoai(kieu: LoaiHinh) -> void	Xóa tất cả các hình theo loại cho trước
XuatHinhTheoChieuTangGiam(kieu: LoaiHinh, tang: bool) -> void	Xuất danh sách hình theo loại cho trước và sắp tăng hoặc giảm
TinhTongDTTheoKieuHinh(kieu: LoaiHinh) -> float	Tính tổng diện tích các hình theo loại

Viết lời gọi hàm trong file ***main.py*** để kiểm tra các chức năng

III. Bài tập về nhà

1. Phân biệt ***class attribute*** và ***instante attribute***
2. So sánh cách thực thi "***access modifiers: public, private, protected***" giữa Python với C#/Java
3. Tìm hiểu thêm về y nghĩa và cách sử dụng các decorator : ***@property, @classmethod, @staticmethod*** trong Python
4. So sánh ***@classmethod*** và ***@staticmethod***
5. So sánh sự giống và khác biệt ở cách áp dụng kế thừa, đa hình giữa Python với C#/Java