



BÀI TẬP

CHUYÊN ĐỀ

LẬP TRÌNH PYTHON NÂNG CAO



BÀI 1: Lập trình hướng đối tượng



Mục tiêu chính: trang bị các kiến thức và kỹ thuật:

- Lập trình hướng đối tượng cơ bản: class, object
- Kế thừa - inheritance
- Lớp trừu tượng – Abstract base class

1.1. Giải phương trình bậc nhất

✓ **Yêu cầu: Viết chương trình giải phương trình bậc nhất**

Giải PT bậc 1: $ax + b = 0$

Nhập a: 4

Nhập b: 2

Kết quả: -0.5

Giải PT bậc 1: $ax + b = 0$

Nhập a: 0

Nhập b: 3

Kết quả: Phương trình vô nghiệm

Giải PT bậc 1: $ax + b = 0$

Nhập a: 0

Nhập b: 0

Kết quả: Phương trình vô số nghiệm

✓ **Hướng dẫn sử dụng:**

- Nhập a, b => chương trình sẽ "Tìm nghiệm" -> hiển thị kết quả

✓ **Tóm tắt yêu cầu**

▪ **Nhập:**

- a, b

▪ **Xuất:**

- Nghiệm phương trình

▪ **Qui tắc xử lý :**

- $ax+b=0$
- $x=-b/a$

✓ **Hướng dẫn**

- Tạo project **Python_nang_cao**
- Trong project Python_nang_cao, tạo **package Bai1**

- Trong package Bai1, tạo module có tên là **giai_ptb1.py**
- Trong module giai_ptb1, tạo một class có tên là **PhuongTrinhBacNhat**, gồm:
 - Các thuộc tính: a, b
 - Phương thức khởi tạo có tham số truyền vào là a, b
 - Phương thức tinh_nghiem() trả về kết quả phương trình bậc 1
 - Phương thức in_nghiem() in kết quả như trên

```
class PhuongTrinhBacNhat(object):
    """
    classdocs: Giải phương trình bậc 1
    """

    def __init__(self, a, b):
        """
        Constructor
        """
        self.a = a
        self.b = b

    def tinh_nghiem(self):
        if self.a == 0 and b != 0:
            return "Phương trình vô nghiệm"
        elif a == 0 and b == 0:
            return "Phương trình vô số nghiệm"
        else:
            return -self.b/self.a

    def in_nghiem(self):
        print("Kết quả:", self.tinh_nghiem())
```

- Gọi sử dụng class PhuongTrinhBacNhat vừa xây dựng

1.2. Tính toán hai số

- ✓ **Yêu cầu:** Xây dựng ứng dụng tính tổng, hiệu, tích, thương của hai số

```
Tính toán hai số:
Nhập Số thứ nhất:      5
Nhập Số thứ hai:      10
Tổng = 15
Hiệu = -5
Tích = 50
Thương = 0.5
```

- ✓ **Hướng dẫn sử dụng:**



- Nhập số thứ nhất, số thứ hai => chương trình sẽ thực hiện việc tính toán tổng, hiệu, tích, thương -> hiển thị kết quả

✓ **Tóm tắt yêu cầu**

▪ **Nhập:**

- Số thứ nhất
- Số thứ hai

▪ **Xuất:**

- Kết quả: Tổng/ hiệu/ tích/ thương

✓ **Hướng dẫn**

- Trong package Bai1, tạo module có tên là **phep_tinh.py**
- Trong module phep_tinh, tạo một lớp PhepTinh gồm có
 - Các thuộc tính: so_thu_nhat, so_thu_hai
 - Phương thức khởi tạo có tham số truyền vào là số thứ nhất, số thứ hai
 - Các phương thức tính tổng/ hiệu/ tích/ thương.

```
class PhepTinh(object):
    """
    classdocs: Phép tính: cộng, trừ, nhân, chia
    """

    def __init__(self, so_thu_nhat, so_thu_hai):
        """
        Constructor
        """
        self.so_thu_nhat = so_thu_nhat
        self.so_thu_hai = so_thu_hai

    def tong(self):
        return self.so_thu_nhat + self.so_thu_hai

    def hieu(self):
        return self.so_thu_nhat - self.so_thu_hai

    def tich(self):
        return self.so_thu_nhat * self.so_thu_hai

    def thuong(self):
        return self.so_thu_nhat/self.so_thu_hai
```

- Gọi sử dụng class PhepTinh
- Tính giá trị cho các phép toán bằng cách gọi hàm được xây dựng trong class PhepTinh.

1.3. Quản lý CD

- ✓ **Viết chương trình xây dựng class CD và quản lý các CD như sau:**

```

Thông tin CD:
Nhập tên CD:      Happy New Year
Nhập tên ca sỹ:   ABBA
Nhập số bài hát:      7
Nhập giá thành: 185000
--- Danh sách CD: ---
# Happy New Year - ABBA - 7 - 185000
Tổng giá thành: 185000
Tiếp tục nhập: 1: Có, 0: Không  1
Nhập tên CD:      Chat với Mozart
Nhập tên ca sỹ:   Mỹ Linh
Nhập số bài hát:      8
Nhập giá thành: 245000
--- Danh sách CD: ---
# Happy New Year - ABBA - 7 - 185000
# Chat với Mozart - Mỹ Linh - 8 - 245000
Tổng giá thành: 430000
Tiếp tục nhập: 1: Có, 0: Không  0
    
```

- Người dùng lần lượt nhập thông tin của các CD, chương trình sẽ quản lý thông tin các CD này và in ra danh sách các CD kèm theo tổng số tiền của các CD.

- ✓ **Tóm tắt yêu cầu**

- **Nhập:**

- Tên CD: chuỗi ký tự
- Ca sỹ: chuỗi ký tự
- Số bài hát: số nguyên
- Giá thành : số thực

- **Xuất:**

- List các CD
- Tổng giá thành của các CD

- ✓ **Hướng dẫn**

- Xây dựng lớp CD như sau:

- Các thuộc tính gồm có: ten, ca_sy, so_bai_hat, gia_thanh
- Xây dựng phương thức khởi tạo cho lớp CD với các tham số truyền vào là ten, ca_sy, so_bai_hat, gia_thanh
- Xây dựng phương thức in thông tin của một CD

```
class CD(object):
    """
    classdocs: class CD
    """
    tong_tien = 0
    def __init__(self, ten, ca_sy, so_bai_hat, gia_thanh):
        """
        constructor
        """
        self.ten = ten
        self.ca_sy = ca_sy
        self.so_bai_hat = so_bai_hat
        self.gia_thanh = gia_thanh
        CD.tong_tien += self.gia_thanh

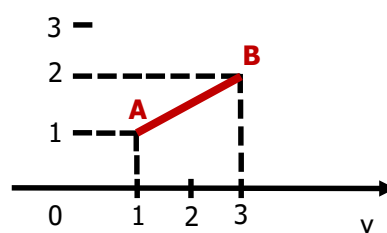
    def thong_tin_cd(self):
        return("# " + self.ten + " - " + self.ca_sy + " - "\
            + str(self.so_bai_hat) + " - " + str(self.gia_thanh))
```

Trong phần gọi sử dụng lớp CD để xử lý yêu cầu:

- Cho người dùng nhập vào thông tin 1 CD => thêm vào list các CD
- Xuất list các CD kèm tổng số tiền của các CD
- Hỏi người dùng có muốn thêm CD mới không. Nếu người dùng chọn **1: có** thì cho người dùng nhập CD mới và lặp lại các bước như trên. Nếu người dùng chọn **0: không** thì dừng chương trình.

1.4. Tính khoảng cách giữa 2 điểm

- ✓ **Yêu cầu:** Viết chương trình tính khoảng cách giữa hai điểm trong mặt phẳng tọa độ Oxy.
 - Sử dụng tiếp cận hướng đối tượng, xây dựng chương trình tính khoảng cách giữa 2 điểm trong mặt phẳng tọa độ 2 chiều Oxy



- Với khoảng cách được tính theo công thức:



- Khoảng cách = $\text{Math.sqrt}((A.x-B.x)^2 + (A.y-B.y)^2)$

Ví dụ:

- Nhập điểm A(x,y) = (1,3)
- Nhập điểm B(x, y) = (2,4)

=> khoảng cách giữa A và B = 1.4142

```
Tính khoảng cách giữa A và B
Nhập Ax:          1
Nhập Ay:          3
Nhập Bx:          2
Nhập By:          4
Khoảng cách giữa A & B= 1.4142
```

✓ Tóm tắt yêu cầu

▪ Nhập:

- Điểm A
- Điểm B

▪ Xuất:

- Khoảng cách giữa hai điểm

▪ Quy tắc xử lý :

- Xử lý cho chức năng “Tính khoảng cách”
 - Khai báo biến diemA, diemB kiểu Diem
 - Gán giá trị cho các biến diemA, diemB (từ giá trị người dùng nhập trên shell)
 - Gọi phương thức tinh_khoang_cach(self, diemB) của đối tượng diemA với giá trị truyền cho tham số là diemB
 - Xuất kết quả tính được

✓ Hướng dẫn:

- Trong package Bai1, tạo module tinh_kc_giua_hai_diem.py. Trong đó:
- Xây dựng lớp Diem như sau:
 - Các thuộc tính gồm có: x,y
 - Xây dựng phương thức khởi tạo cho lớp Diem với hai tham số truyền vào là x, y
 - Xây dựng phương thức để tính toán khoảng cách giữa điểm hiện hành A và điểm B.



```
import math
class Diem(object):
    """
    classdocs: Phép tính: cộng, trừ, nhân, chia
    """

    def __init__(self, x, y):
        """
        Constructor
        """
        self.x = x
        self.y = y

    def tinh_khoang_cach(self, B):
        return math.sqrt((self.x-B.x)*( self.x-B.x)+ (self.y-B.y)*( self.y-B.y))
```

- Gọi sử dụng lớp Diem để tính toán và in ra khoảng cách.

1.5. Tính lương (bài làm thêm)

✓ Yêu cầu:

Xây dựng chương trình tính thuế thu nhập cá nhân và thực lĩnh hàng tháng của một nhân viên. Thông tin của một nhân viên gồm có: họ tên, hệ số lương, số người giảm trừ gia cảnh, phụ cấp trong tháng của nhân viên.

Với lương cơ bản = 1.260.000

Thu nhập = hệ số lương * lương cơ bản + phụ cấp

Thu nhập chịu thuế = thu nhập – 9.000.000 – số người giảm trừ gia cảnh * 3.600.000

Thuế TNCN = tính theo biểu thuế dựa vào thu nhập chịu thuế (bảng phía dưới)

Thực lĩnh = thu nhập – thuế TNCN

Biểu thuế TNCN

Bậc thuế	Thu nhập chịu thuế/tháng (triệu đồng)	Thuế suất (%)	Số tiền thuế tối đa của bậc thuế
1	Thu nhập chịu thuế < 5	5	250000
2	5<= thu nhập chịu thuế <10	10	500000
3	10<= thu nhập chịu thuế <18	15	1200000
4	18<= thu nhập chịu thuế <32	20	2800000
5	32<= thu nhập chịu thuế <52	25	5000000
6	52<= thu nhập chịu thuế <80	30	8400000



7	Thu nhập chịu thuế > 80	35	
---	-------------------------	----	--

Ví dụ: Nhân viên:

- Họ tên: Nguyễn Văn An
- Hệ số lương: 2.67
- Số người giảm trừ gia cảnh: 1
- Phụ cấp: 12.000.000

⇒ In kết quả:

- Thu nhập = $2.67 * 1.260.000 + 12.000.000 = 15.364.200$
- Thu nhập chịu thuế = $15.364.200 - 9.000.000 - 1 * 3.600.000 = 2.764.200$
- Thuế thu nhập cá nhân = $2.764.200 * 5/100 = 138.210$
- Thực lĩnh = $15.364.200 - 138.210 = 15.225.990$

✓ Tóm tắt yêu cầu

▪ Nhập:

- Họ tên
- Hệ số lương
- Số người giảm trừ gia cảnh
- Phụ cấp

▪ Xuất:

- Thu nhập
- Thu nhập chịu thuế
- Thuế thu nhập cá nhân
- Thực lĩnh

✓ Hướng dẫn

- Trong package Bai1, tạo module **quan_ly_nhan_vien.py**. Trong đó:
- Xây dựng lớp NhanVien
 - Với các thuộc tính như đã nêu trong yêu cầu
 - Xây dựng phương thức khởi tạo cho lớp nhân viên với các tham số truyền vào: họ tên, hệ số lương, số người giảm trừ gia cảnh, phụ cấp
 - Xây dựng các phương thức cần thiết cho lớp nhân viên
- Trong phần gọi sử dụng lớp nhân viên để xử lý yêu cầu:
 - Cho người dùng nhập vào thông tin 1 nhân viên.
 - Tính toán và hiển thị các kết quả như yêu cầu.

Mở rộng: Có thể tạo một danh sách nhân viên. Thực hiện lặp lại việc tạo nhân viên, mỗi lần tạo một nhân viên thì thêm vào danh sách. => Hiển thị danh sách nhân viên.

1.6. Quản lý giao dịch (kế thừa)



✓ **Yêu cầu: Xây dựng ứng dụng quản lý danh sách các giao dịch:**

Mô tả: Hệ thống quản lý 2 loại giao dịch:

- Giao dịch vàng: Mã giao dịch, ngày giao dịch (ngày/tháng/năm), đơn giá, số lượng, loại vàng có 3 loại 18k, 24k, 9999.
 - Thành tiền được tính như sau: thành tiền = số lượng * đơn giá.
- Giao dịch tiền tệ: Mã giao dịch, ngày giao dịch (ngày/tháng/năm), tỷ giá (cũng là đơn giá), số lượng, loại tiền tệ có 3 loại: USD, EUR, AUD, loại giao dịch mua/bán. Thành tiền được tính như sau:
 - Nếu loại giao dịch là "mua" thì: thành tiền = số lượng * tỷ giá
 - Nếu loại giao dịch là "bán" thì: thành tiền = (số lượng * tỷ giá) * 1.05

```

Quản lý giao dịch:
Nhập mã GD:      gd001
Nhập ngày GD:    13/03/2017
Nhập số lượng:   10
Chọn loại giao dịch: 1: Vàng, 2: Tiền Tệ:      1
Chọn loại: 18k / 24k / 9999:      18k
Nhập đơn giá:    2350000
gd001 - 13/03/2017 - 18k - 10 - 2350000 - Thành tiền = 23500000
Tổng số lượng: 10
Tổng số tiền: 23500000
Bạn muốn tiếp tục giao dịch? 1: Có, 0: Không      1
Nhập mã GD:      gd002
Nhập ngày GD:    14/03/2017
Nhập số lượng:   100
Chọn loại giao dịch: 1: Vàng, 2: Tiền Tệ:      2
Chọn loại: USD / EUR / AUD:      USD
Nhập tỷ giá:      23000
Bạn mua hay bán? 1: mua, 0: bán:      1
GD mua: gd002 - 14/03/2017 - USD - 100 - 23000 - Thành tiền = 2300000
Tổng số lượng: 100
Tổng số tiền: 2300000
Bạn muốn tiếp tục giao dịch? 1: Có, 0: Không      1
Nhập mã GD:      gd003
Nhập ngày GD:    16/03/2017
Nhập số lượng:   200
Chọn loại giao dịch: 1: Vàng, 2: Tiền Tệ:      2
Chọn loại: USD / EUR / AUD:      EUR
Nhập tỷ giá:      24500
Bạn mua hay bán? 1: mua, 0: bán:      0
GD mua: gd002 - 14/03/2017 - USD - 100 - 23000 - Thành tiền = 2300000
GD bán: gd003 - 16/03/2017 - EUR - 200 - 24500 - Thành tiền = 5145000.0
Tổng số lượng: 300
Tổng số tiền: 7445000.0
Bạn muốn tiếp tục giao dịch? 1: Có, 0: Không      0
    
```



Dựa vào mô tả trên, hãy:

- Tạo lớp **GiaoDich** với các thuộc tính và phương thức chung (giao dịch vàng cũng là giao dịch).
- Tạo lớp **GiaoDichTienTe** kế thừa từ lớp GiaoDich với các thuộc tính riêng và phương thức cần thiết.
- Nhập xuất danh sách các giao dịch.
- Tính tổng số lượng cho từng loại.
- Tính tổng thành tiền cho từng loại.

✓ **Hướng dẫn sử dụng:**

- Nhập mã giao dịch, ngày, số lượng
- Chọn loại giao dịch (1: Giao dịch Vàng, 2: Giao dịch Tiền tệ):
- Nếu chọn 1:
 - Chọn loại, nhập đơn giá => thêm vào list giao dịch vàng => hiển thị danh sách các giao dịch vàng: thông tin giao dịch, tổng số lượng, tổng số tiền
- Nếu chọn 2:
 - Chọn loại, nhập tỷ giá, nhập loại mua/bán => thêm vào list tương ứng => hiển thị thông tin các giao dịch, tổng số lượng, tổng thành tiền
- Sau mỗi lần thêm, chương trình sẽ hỏi người dùng có thêm tiếp hay không? Nếu chọn 1: có tiếp tục thêm, nếu chọn 0: không dừng chương trình.

✓ **Tóm tắt yêu cầu**

▪ **Nhập:**

- Loại vàng/ Loại tiền tệ
- Mã
- Ngày
- Số lượng
- Đơn giá/ Tỷ giá
- Loại giao dịch

▪ **Xuất:**

- Danh sách giao dịch
- Tổng số lượng
- Tổng thành tiền

✓ **Hướng dẫn**

- Trong package Bai1, tạo module **quan_ly_giao_dich.py**. Trong đó:
- Tạo các lớp GiaoDich, GiaoDienTienTe:



```
class GiaoDich(object):
    """
    classdocs: class GiaoDich
    """

    def __init__(self, ma, ngay, don_gia, so_luong, loai):
        """
        Constructor
        """
        self.ma = ma
        self.ngay = ngay
        self.don_gia = don_gia
        self.so_luong = so_luong
        self.loai = loai

    def thanh_tien(self):
        return self.so_luong * self.don_gia

    def in_giao_dich(self):
        return self.ma + " - " + self.ngay + " - " + self.loai + " - " + str(self.so_luong) \
            + " - " + str(self.don_gia) + " - Thành tiền = " + str(self.thanh_tien())
```

- Tạo lớp GiaoDichTienTe kế thừa từ lớp GiaoDich

```
class GiaoDichTienTe(GiaoDich):
    """
    classdocs: class GiaoDichTienTe kế thừa từ lớp giao dich
    """

    def __init__(self, ma, ngay, don_gia, so_luong, loai, mua):
        """
        Constructor
        """
        self.mua = mua
        GiaoDich.__init__(self, ma, ngay, don_gia, so_luong, loai)

    def thanh_tien(self):
        if self.mua:
            return GiaoDich.thanh_tien(self)
        else:
            return GiaoDich.thanh_tien(self) * 1.05

    def in_giao_dich(self):
        if self.mua:
            return "GD mua: " + GiaoDich.in_giao_dich(self)
        else:
            return "GD bán: " + GiaoDich.in_giao_dich(self)
```

- Gọi các lớp đã xây dựng để xử lý tính toán và hiển thị.

1.7. Quản lý sách thư viện (kế thừa)

✓ Yêu cầu: Xây dựng ứng dụng quản lý sách trong thư viện

Mô tả: Thư viện X quản lý danh sách các loại sách. Thông tin về các loại sách:

- Sách giáo khoa: Mã sách, tên sách, ngày nhập (ngày/tháng/năm), đơn giá, số lượng, nhà xuất bản, tình trạng (mới, cũ)



- Nếu tình trạng sách là mới thì: thành tiền = số lượng * đơn giá
- Nếu tình trạng sách là cũ thì: thành tiền = số lượng * đơn giá * 50%
- Sách tham khảo: Mã sách, tên sách, ngày nhập (ngày/tháng/năm), đơn giá, số lượng, nhà xuất bản, thuế
 - Thành tiền = (số lượng * đơn giá) + (số lượng * đơn giá) * thuế (với thuế có giá trị từ 1% đến 20%)

```

Nhập mã sách:      gk001
Nhập tên sách:     Tiếng Việt 1 - Tập 1
Ngày nhập:        17/03/2017
Nhập đơn giá:     10500
Nhập số lượng:    100
Nhập NXB:         Giáo Dục
Đây là sách giáo khoa hay tham khảo? 1: Giáo Khoa, 2: Tham khảo:      1
Sách cũ hay mới? 1: Mới, 0: Cũ:      1
Sách giáo khoa:
gk001 - Tiếng Việt 1 - Tập 1 - Giáo Dục - 17/03/2017 - 10500 -100 - Thành tiền = 1050000
Tổng tiền = 1050000
Sách mới = 100 , Sách cũ = 0
Bạn có tiếp tục nhập sách? 1: Có, 0: Không:      1
Nhập mã sách:      gk002
Nhập tên sách:     Toán 1 - Tập 1
Ngày nhập:        18/03/2017
Nhập đơn giá:     9800
Nhập số lượng:    50
Nhập NXB:         Giáo Dục
Đây là sách giáo khoa hay tham khảo? 1: Giáo Khoa, 2: Tham khảo:      1
Sách cũ hay mới? 1: Mới, 0: Cũ:      0
Sách giáo khoa:
gk001 - Tiếng Việt 1 - Tập 1 - Giáo Dục - 17/03/2017 - 10500 -100 - Thành tiền = 1050000
gk002 - Toán 1 - Tập 1 - Giáo Dục - 18/03/2017 - 9800 -50 - Thành tiền = 245000.0
Tổng tiền = 1295000.0
Sách mới = 100 , Sách cũ = 50
Bạn có tiếp tục nhập sách? 1: Có, 0: Không:      1
Nhập mã sách:      tk001
Nhập tên sách:     Hướng dẫn Làm toán 1
Ngày nhập:        20/03/2017
Nhập đơn giá:     25000
Nhập số lượng:    30
Nhập NXB:         ĐHQG TP.HCM
Đây là sách giáo khoa hay tham khảo? 1: Giáo Khoa, 2: Tham khảo:      2
Nhập thuế (từ 1 - 20 %):      10
Sách tham khảo:
tk001 - Hướng dẫn Làm toán 1 - ĐHQG TP.HCM - 20/03/2017 - 25000 -30 - Thành tiền = 825000.0
Tổng tiền = 825000.0
Bạn có tiếp tục nhập sách? 1: Có, 0: Không:      0
    
```

Dựa vào mô tả trên, hãy:

- Tạo lớp **Sach** với các thuộc tính và phương thức chung.
- Tạo lớp **SachGiaoKhoa** và lớp **SachThamKhao** kế thừa từ lớp **Sach** với các thuộc tính riêng và phương thức cần thiết.
- Nhập xuất danh sách các loại sách.
- Tính tổng thành tiền cho từng loại.



- Cho biết trong sách giáo khoa có bao nhiêu sách cũ, bao nhiêu sách mới.

✓ **Hướng dẫn sử dụng:**

- Nhập thông tin chung về sách, sau đó chọn loại sách (giáo khoa/ tham khảo) và thông tin chi tiết cho loại sách được chọn => thêm vào list tương ứng => Hiển thị thông tin các sách trong thư viện với tổng thành tiền của từng loại và tổng sách giáo khoa mới/cũ.
- Sau mỗi lần thêm, chương trình sẽ hỏi người dùng có thêm tiếp hay không? (1: có, 0: không). Nếu chọn 1 tiếp tục thêm, nếu chọn 0 dừng chương trình.

✓ **Tóm tắt yêu cầu**

▪ **Nhập:**

- Mã sách
- Tên sách
- Ngày nhập
- Nhà xuất bản
- Đơn giá
- Số lượng
- Sách giáo khoa (Tình trạng)/ Sách tham khảo (Thuế)

▪ **Xuất:**

- Danh sách sách
- Tổng thành tiền sách giáo khoa/ sách tham khảo
- Tổng số sách giáo khoa mới/ cũ

✓ **Hướng dẫn**

- Trong package Bai6, tạo module **quan_ly_sach.py**. Trong đó,
- Tạo lớp Sach:



```
class Sach(object):
    """
    classdocs: class sach
    """

    def __init__(self, ma_sach, ten_sach, ngay_nhap, don_gia, so_luong, nxb):
        """
        Constructor
        """
        self.ma_sach = ma_sach
        self.ten_sach = ten_sach
        self.ngay_nhap = ngay_nhap
        self.don_gia = don_gia
        self.so_luong = so_luong
        self.nxb = nxb

    def tinh_thanh_tien(self):
        return self.so_luong * self.don_gia

    def in_sach(self):
        return self.ma_sach + " - " + self.ten_sach + " - " + self.nxb + " - " \
            + self.ngay_nhap + " - " + str(self.don_gia) + " - " + str(self.so_luong) \
            + " - Thành tiền = " + str(self.tinh_thanh_tien())
```

- Tạo lớp SachGiaoKhoa kế thừa từ lớp Sach:

```
class SachGiaoKhoa(Sach):
    """
    classdocs: class SachGiaoKhoa có thêm tình trạng sách
    """

    sach_cu = 0
    sach_moi = 0

    def __init__(self, ma_sach, ten_sach, ngay_nhap, don_gia, so_luong, nxb, tinh_trang):
        """
        Constructor: class SachThamKhao có thêm tình trạng = True (mới)/ False (cũ)
        """
        Sach.__init__(self, ma_sach, ten_sach, ngay_nhap, don_gia, so_luong, nxb)
        self.tinh_trang = tinh_trang
        if (self.tinh_trang):
            SachGiaoKhoa.sach_moi += self.so_luong
        else:
            SachGiaoKhoa.sach_cu += self.so_luong

    def tinh_thanh_tien(self):
        if (self.tinh_trang):
            return Sach.tinh_thanh_tien(self)
        else:
            return Sach.tinh_thanh_tien(self) * 0.5
```

- Tạo lớp SachThamKhao kế thừa từ lớp Sach:



```
class SachThamKhao(Sach):
    ...
    classdocs: class SachThamKhao có thêm thuế
    ...

    def __init__(self, ma_sach, ten_sach, ngay_nhap, don_gia, so_luong, nxb, thue):
        ...
        Constructor: Thuế 1% - 20%
        ...
        Sach.__init__(self, ma_sach, ten_sach, ngay_nhap, don_gia, so_luong, nxb)
        self.thue= thue

    def tinh_thanh_tien(self):
        return Sach.tinh_thanh_tien(self) + (Sach.tinh_thanh_tien(self) * self.thue / 100)
```

- Thực hiện:
 - Gọi sử dụng class sách tương ứng với thông tin nhập
 - Đưa sách vừa nhập vào danh sách Sách giáo khoa hoặc Sách tham khảo
 - Tính toán
 - Xuất kết quả

1.8. Tính chu vi & diện tích các hình (abstract)

✓ **Yêu cầu:** Viết chương trình tính chu vi và diện tích của một số hình như sau:

- Hình tròn
- Hình chữ nhật
- Hình tam giác

```
Bạn chọn hình nào? 1: Tròn, 2: Chữ nhật, 3: Tam giác:    1
Nhập bán kính:    10
Chu vi = 62.83
Diện tích = 314.16
```

```
Bạn chọn hình nào? 1: Tròn, 2: Chữ nhật, 3: Tam giác:    2
Nhập chiều dài:    5
Nhập chiều rộng:    12
Chu vi = 34
Diện tích = 60
```

```
Bạn chọn hình nào? 1: Tròn, 2: Chữ nhật, 3: Tam giác:    3
Nhập cạnh a:    2
Nhập cạnh b:    3
Nhập cạnh c:    4
Chu vi = 9
Diện tích = 2.9
```

✓ **Hướng dẫn sử dụng:**



- Chọn hình muốn tính diện tích và chu vi.
- Ứng với hình được chọn: nhập bán kính hình tròn/chiều dài và chiều rộng hình chữ nhật / ba cạnh hình tam giác => hiển thị diện tích và chu vi tương ứng

✓ **Tóm tắt yêu cầu**

▪ **Nhập:**

- Loại hình
- Bán kính/ chiều dài & chiều rộng/ cạnh

▪ **Xuất:**

- Chu vi
- Diện tích

✓ **Hướng dẫn**

- Xây dựng lớp trừu tượng Hình như sau:

```
class Hình(object):
    """
    classdocs: Hình là abstract base class
    """
    __metaclass__ = abc.ABCMeta

    @classmethod
    @abc.abstractmethod
    def __init__(self, metaclass__):
        self.__metaclass__ = metaclass__

    def tinh_chu_vi(self):
        pass

    def tinh_dien_tich(self):
        pass
```

- Xây dựng lớp HìnhTron kế thừa từ lớp trừu tượng Hình

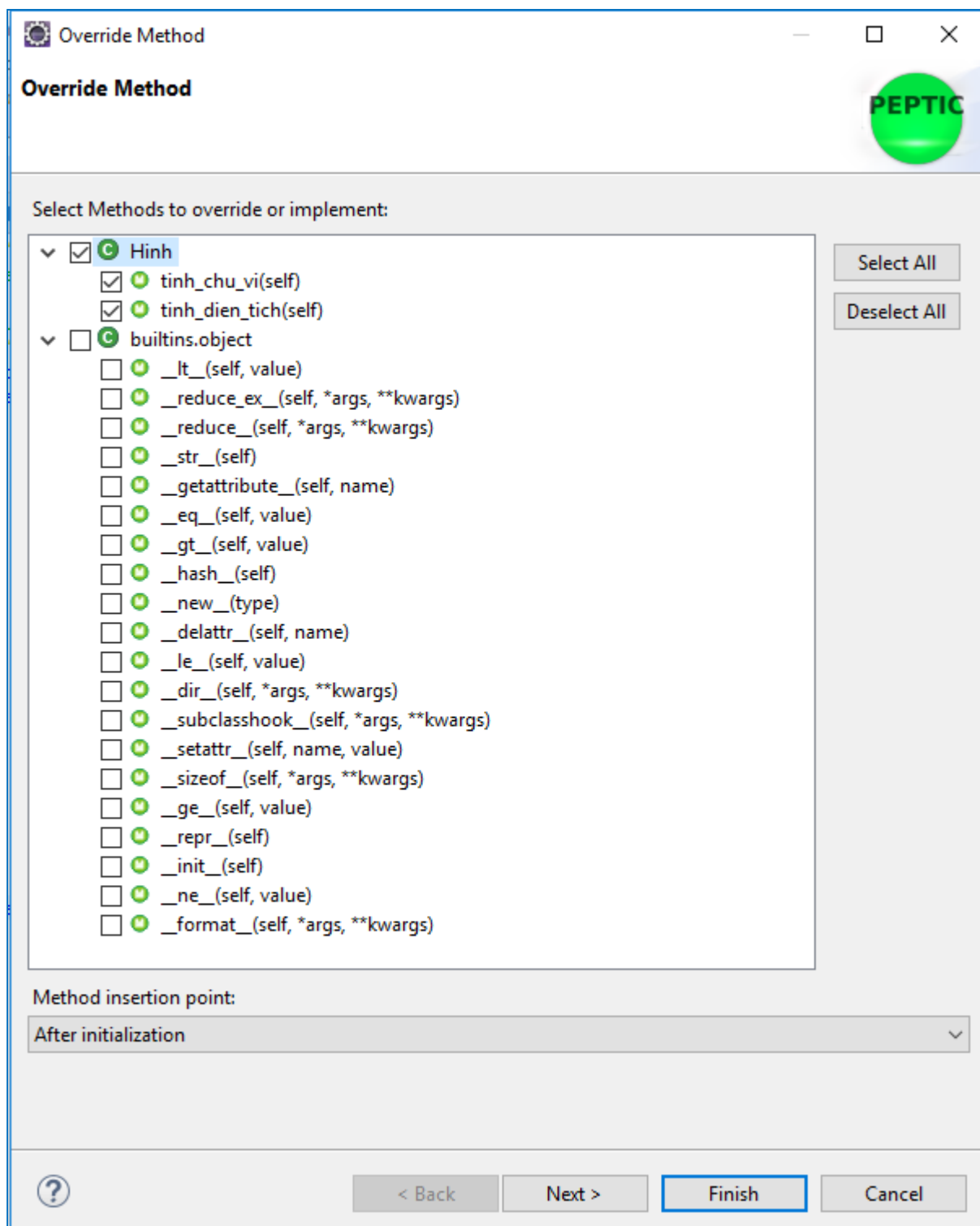
```
class HìnhTron(Hình):
    """
    classdocs: HìnhTron tính chu vi và diện tích
    """

    def __init__(self, r):
        self.r = r

    def tinh_chu_vi(self):
        return 2 * math.pi * self.r

    def tinh_dien_tich(self):
        return math.pi * math.pow(self.r, 2)
```

- Làm tương tự cho các lớp còn lại
- Khởi tạo và gọi sử dụng các lớp này
- Ghi chú: để override abstract method: click chuột phải chọn Refactoring > Override/Implement methods... > chọn các method cần override > Finish



1.9. Tính lương (kế thừa) – làm tiếp từ bài tính lương phía trên (bài làm thêm)

- ✓ **Yêu cầu:** Xây dựng chương trình tính lương cho nhân viên của công ty ABC.



Công ty ABC cần xây dựng chức năng tính lương cho nhân viên của công ty. Thông tin của nhân viên trong công ty bao gồm: họ tên, hệ số lương, số người giảm trừ gia cảnh và phụ cấp tháng

Nhân viên trong công ty được chia thành 2 loại:

- **Nhân viên kinh doanh:** mỗi nhân viên kinh doanh có thêm mức lương kinh doanh hàng tháng, được hưởng dựa trên **doanh số (tỷ lệ lương kinh doanh)** đạt được trong tháng. Tỷ lệ này từ 0% – 200%

Lương thưởng = lương cơ bản * tỷ lệ lương kinh doanh

- **Nhân viên sản xuất:** mỗi nhân viên sản xuất có một **định mức sản phẩm** làm ra hàng tháng là 450 sản phẩm, nếu **số lượng sản phẩm** vượt định mức sẽ được hưởng thêm tương ứng với số lượng sản phẩm vượt định mức được tính theo công thức sau:

Lương thưởng = (số sản phẩm-định mức sản phẩm) * đơn giá gia công sản phẩm

Nếu không vượt định mức thì Lương thưởng = 0

Với lương cơ bản cho sẵn là 1.260.000, đơn giá gia công sản phẩm = 10000

- Lương của nhân viên được tính như sau:
 - Thu nhập = hệ số lương * lương cơ bản + lương thưởng
 - Thu nhập chịu thuế = thu nhập – 9.000.000 – số người giảm trừ gia cảnh * 3.600.000
 - Thuế TNCN = tính theo biểu thuế dựa vào thu nhập chịu thuế
 - Thực lĩnh = thu nhập – thuế TNCN

Bậc thuế	Thu nhập chịu thuế/tháng (triệu đồng)	Thuế suất (%)	Số tiền thuế tối đa của bậc thuế
1	Thu nhập chịu thuế < 5	5	250000
2	5 ≤ thu nhập chịu thuế < 10	10	500000
3	10 ≤ thu nhập chịu thuế < 18	15	1200000
4	18 ≤ thu nhập chịu thuế < 32	20	2800000
5	32 ≤ thu nhập chịu thuế < 52	25	5000000
6	52 ≤ thu nhập chịu thuế < 80	30	8400000
7	Thu nhập chịu thuế > 80	35	

Dựa vào mô tả trên, hãy:

- Tạo lớp nhân viên sản xuất và lớp nhân viên kinh doanh kế thừa từ lớp nhân viên (bài đã làm 1.6) với các thuộc tính riêng và phương thức cần thiết.
- Nhập xuất danh sách các nhân viên theo loại.
- Tính lương trung bình theo loại

✓ **Hướng dẫn sử dụng:**



- Nhập thông tin chung về nhân viên, sau đó chọn loại nhân viên và nhập thông tin chi tiết cho loại nhân viên được chọn => thêm vào list tương ứng => Hiển thị thông tin các nhân viên theo loại và tính lương trung bình của từng loại.
- Sau mỗi lần thêm, chương trình sẽ hỏi người dùng có thêm tiếp hay không? Nếu chọn 1: có - tiếp tục thêm, nếu chọn 0: không - dừng chương trình.

✓ **Tóm tắt yêu cầu**

▪ **Nhập:**

- Họ tên
- Hệ số lương
- Số người giảm trừ gia cảnh
- Phụ cấp
- Nhân viên kinh doanh: lương kinh doanh, tỷ lệ lương kinh doanh
- Nhân viên sản xuất: số sản phẩm

▪ **Xuất:**

- Danh sách nhân viên theo loại, mỗi nhân viên gồm tất cả các thông tin và thu nhập, thu nhập chịu thuế, thuế TNCN, thực lĩnh
- Lương trung bình thu nhập theo loại

✓ **Hướng dẫn**

- Trong package Bai1, tạo module **quan_ly_nhan_vien_ke_thua.py**. Trong đó:
- Tạo lớp **NhanVienKinhDoanh** và **NhanVienSanXuat** kế thừa lại lớp **NhanVien** đã xây dựng trong module **quan_ly_nhan_vien.py**
- Thực hiện:
 - Gọi sử dụng class nhân viên tương ứng với thông tin nhập
 - Đưa nhân viên vừa nhập vào danh sách nhân viên kinh doanh/ nhân viên sản xuất
 - Tính toán
 - Xuất kết quả

BÀI 2: Làm việc với dữ liệu JSON



Mục tiêu chính:

- *Làm việc với dữ liệu JSON từ API trên Internet: đọc, hiển thị nội dung*
- *Làm việc với tập tin .json: đọc, hiển thị, ghi tập tin .json*

2.1. Đọc và hiển thị các sản phẩm từ API trên Internet

✓ **Yêu cầu: Xây dựng chương trình đọc và hiển thị các sản phẩm từ API trên Internet:**

- Thông tin các sản phẩm được lưu tại địa chỉ: <http://api.laptrinhpython.net/san-pham>
- Khi chạy chương trình, sẽ hiển thị danh sách các sản phẩm nằm trong API này như sau:
 - Tổng số sản phẩm
 - Danh sách sản phẩm: STT, Tên sản phẩm, Hình sản phẩm, Giá size S, Giá size M

DANH SÁCH SẢN PHẨM (73)				
STT	TÊN SẢN PHẨM	HÌNH SẢN PHẨM	GIÁ SIZE S	GIÁ SIZE M
1	Bắp Hấp FM	Family-Mart-Mar-05-06-2.jpg	6000	0
2	Cá Viên Cơm Xanh Mayo	familymart-Oct-W2-2.jpg	12000	0
3	Cá Viên Tôm	FamilyMart-Sep-W2-4-768x768.gif	12000	0
4	Cánh Gà Chiên	1CanhGaChien.jpg	25000	0
5	Pizza Phô Mai Chiên	1Piza.jpg	14000	0
6	Set Khoai Tây Nghiền	familymart-Sep-W3-2.png	8000	0
7	Takoyaki Set	Family-Mart-Oct-02-03-2.jpg	22000	0
8	Bò Viên	Bò-Viên-TVS.jpg	6000	0
9	Chả Cá Bọc Bắp Non	FM-180413-chacabapnon.jpg	5000	0
10	Chả Cá Bọc Trứng Cút	FM-180508-chacaboctrungcut-s.jpg	6000	0
11	Chả Tôm	FamilyMart-July-W3-2.png	6000	0
12	Đậu Hũ Bọc Hải Sản	pinggosh_fullsize_distri.png	5000	0
13	Khoai Tây Surimi	Khoai-Tay-Surimi.jpg	6000	0
14	Mực Viên	Family-Mart-Nov-03-01.jpg	8000	0
15	Thanh Chả Heo	6.jpg	9000	0
16	Viên Măng Rau Củ	FamilyMart-May-W3-4-2.jpg	5000	0
17	Viên Mực Rau Củ	Family-Mart-Mar-04-02.jpg	6000	0
18	Xúc Xích HotPot	22.jpg	10000	0
19	Trà Sữa Táo Xanh	tra_sua_tao_xanh.jpg	35000	40000
20	Trà Sữa Trân Châu Đường Đen	tra_sua_duong_den_1.jpg	37000	42000
21	Trà Sữa Dâu	traa_sua_sau.jpg	37000	42000
22	Trà Sữa Đào	tra_sua_dao.jpg	35000	40000
23	Trà Sữa Mật Ong	tra_sua_mat_ong.jpg	37000	42000
24	Trà Sữa Dưa Lưới	tra_sua_dua_luoi.jpg	35000	40000
25	Trà Sữa Kiwi	tra_sua_kiwi.jpg	35000	40000
26	Trà Sữa Khoai Môn	tra_sua_khoa_mon.jpg	37000	42000
27	Trà Sữa Truyền Thống	tra_sua_tuyen_thong.jpg	30000	35000
28	Trà Sữa Hoa Hướng Dương	tra_sua_hoa_huong_duong.jpg	42000	44000
29	Hồng Trà Sữa	hong_tra_sua.jpg	30000	35000
30	Trà Sữa Gạo	tra_sua_gao_gDfHuVp.jpg	37000	42000
31	Trà Xanh 888	tra_sua_888.jpg	27000	29000
32	Bá Tước Sủi Bọt	ba_tuoc_sui_bot.jpg	27000	29000
33	Ôlong Sủi Bọt	oLong_sui_bot.jpg	27000	29000
34	Lục Trà Sủi Bọt	luc_tra_sui_bot.jpg	27000	29000
35	Hồng Trà Sủi Bọt	hong_tra_sua_zgUaRwf.jpg	27000	29000
36	Bá Tước Sữa	ba_tuoc_sua.jpg	37000	39000
37	Ôlong Sữa	olong_sua.jpg	37000	39000
38	Lục Trà Tắc Chanh Dây	luc_tra_tac_chanh_day.jpg	35000	40000
39	Lục Trà Dâu	luc_tra_dau.jpg	29000	32000
40	Lục Trà Chanh Tươi	luc_tra_chanh_tuoi.jpg	29000	32000
41	Matcha Macchiato	Matcha_Macchiato.jpg	55000	0
42	Matcha Đậu Đỏ	Matcha_dau_do.jpg	49000	55000
43	Lục Trà Bọt Sữa	luc_tra_bot_sua.jpg	45000	0
44	Hồng Trà Bọt Sữa	hong_tra_bot_sua.jpg	35000	0
45	Nước Ép Chanh Dây	nuoc_ep_chai_day.jpg	3700	0
46	Nước ép cam	nuoc_ep_cam.jpg	37000	0
47	Nước ép táo	nuoc_ep_tao.jpg	42000	0
48	Nước ép bưởi	nuoc_ep_buoi.jpg	37000	0
49	Nước lô hội mật ong chanh	nuoc_ep_lohoi_mat_ong_chanh.jpg	37000	0
50	Cà phê sữa	ca_phe_sua.jpg	37000	0
51	Cà phê đen	ca_phe_den.jpg	37000	0
52	Hạt 3Q Ngọc Trai	hat_3q_ngoc_trai.jpg	10000	0
53	Đào Miếng	dao_mieng.jpg	10000	0
54	Kem Sữa	kem_sua.jpg	10000	0
55	Hạt thủy tinh Yogurt	hat_3q_ngoc_trai_D8H3aeu.jpg	10000	0
56	Bánh flan trứng	banh_plan_trung.jpg	7000	0
57	Đậu đỏ	dau_do.jpg	10000	0
58	Hạt lô hội	hat_lo_hoi.jpg	7000	0
59	Hạt trái cây	hat_trai_cay.jpg	7000	0
60	Hạt Cà Phê	hat_ca_phe.jpg	7000	0
61	Sương Sáo	suong_sao.jpg	7000	0
62	Hạt Trân Châu	hat_tran_chau.jpg	7000	0
63	Sinh tố	sinhto.jpg	15000	17000



✓ **Tóm tắt yêu cầu**

▪ **Qui tắc xử lý:**

- Cung cấp đường link
- Đọc và hiển thị các sản phẩm như định dạng trên

✓ **Hướng dẫn**

- Một item sản phẩm trong đường dẫn có cấu trúc như sau:

```
{
  "chi_tiet": "<p><img alt=\"\"
src=\"http://localhost:8081/ThucAnNhanh/public/hinh_chi_tiet_san_pham/images/Family-
Mart-Mar-05-06-2.jpg\" style=\"height:500px; width:500px\" /></p>\r\n\r\n<p>Bắp được
cải thiện bằng c&acute;ch l&agrave;m ch&iacute;n bằng hấp rồi mới cấp đ&ocirc;ng.
V&igrave; vậy vẫn được giữ nguy&ecirc;n vị ngọt v&agrave; độ căng mọng của từng hạt
bắp.</p>",
  "created_at": "2018-12-26 08:33:47",
  "gia_size_m": 0,
  "gia_size_s": 6000,
  "giam_gia": 0,
  "hinh_san_pham": "Family-Mart-Mar-05-06-2.jpg",
  "hot": 0,
  "id": 1,
  "like": 1,
  "ma_loai": 13,
  "mo_ta_tom_tat": "Bắp được cải thiện bằng cách làm chín bằng hấp rồi mới cấp đông. Vì
vậy vẫn được giữ nguyên vị ngọt và độ căng mọng của từng hạt bắp.",
  "san_pham_an_cung": null,
  "ten_san_pham": "Bắp Hấp FM",
  "ten_url": "bap-hap-fm",
  "trang_thai": 1,
  "updated_at": "2018-12-26 08:33:47",
  "xoa": 0
}
```

- Trong project Python_nang_cao, tạo **package Chuong_JSON**. Trong đó:
- Tạo module **read_json_from_internet.py**. Trong đó:
 - Viết function **doc_json_api_unicode(URL)**: Đọc và trả về nội dung file JSON:

```
import json
import urllib.request

def read_json_from_internet_unicode(url):
    DEFAULT_ENCODING = 'utf-8'
    urlResponse = urllib.request.urlopen(url)
```



```

if hasattr(urlResponse.headers, 'get_content_charset'):
    encoding = urlResponse.headers.get_content_charset(DEFAULT_ENCODING)
else:
    encoding = urlResponse.headers.getparam('charset') or DEFAULT_ENCODING
noi_dung = json.loads(urlResponse.read().decode(encoding))
return noi_dung

if __name__ == "__main__":
    # Gọi hàm và xử lý

```

- Gọi hàm để lấy nội dung
- Xử lý nội dung => hiển thị kết quả

2.2. Đọc và hiển thị danh sách tivi theo nhóm từ API trên Internet

- ✓ **Yêu cầu: Xây dựng chương trình đọc và hiển thị danh sách tivi theo mã nhóm từ API trên Internet:**

- Thông tin các tivi được lưu tại địa chỉ: <http://api.laptrinhpython.net/tivi>
- Khi chạy chương trình, sẽ hiển thị tivi theo mã nhóm nằm trong API này như sau:
 - Danh sách xxx tivi thuộc nhóm yyy: (với xxx là số lượng, yyy là tên nhóm)
 - Danh sách tivi: Tên tivi, Mã số, Ký hiệu, Đơn giá bán, Số lượng tồn

```

Danh sách 5 tivi thuộc nhóm SAMSUNG:
1 / Smart Tivi Cong Samsung 49 inch
- Mã số: TIVI_1
- Ký hiệu: UA49MU6500
- Đơn giá bán: 29.900.000 đ
- Số lượng tồn: 99
2 / Smart Tivi Samsung 43 inch
- Mã số: TIVI_3
- Ký hiệu: UA43M5500
- Đơn giá bán: 12.390.000 đ
- Số lượng tồn: 9
3 / Internet Tivi Samsung 32 inch
- Mã số: TIVI_8
- Ký hiệu: UA32J4303
- Đơn giá bán: 6.890.000 đ
- Số lượng tồn: 11
4 / Smart Tivi Samsung 40 inch
- Mã số: TIVI_11
- Ký hiệu: UA40K5300
- Đơn giá bán: 8.640.000 đ
- Số lượng tồn: 62
5 / TV SamSung
- Mã số: TIVI_25
- Ký hiệu: 49X8000E
- Đơn giá bán: 11.290.000 đ
- Số lượng tồn: 75

```




✓ **Tóm tắt yêu cầu**

▪ **Qui tắc xử lý:**

- Cung cấp đường link kèm các tham số điều kiện (**Mã nhóm**: SAMSUNG, LG, SONY, KHAC): http://api.laptrinhpython.net/tivi/Mã_nhóm

Ví dụ: <http://api.laptrinhpython.net/tivi/SAMSUNG>

- Đọc và hiển thị các tivi theo nhóm như định dạng trên

✓ **Hướng dẫn**

- Một item tivi trong đường dẫn có cấu trúc như sau:

```
{ "Don_gia_Ban": 29900000,
  "Don_gia_Nhap": 18100000,
  "Ky_hieu": "UA49MU6500",
  "Ma_so": "TIVI_1",
  "Nhom_Tivi": { "Ma_so": "SAMSUNG", "Ten": "Samsung" },
  "So_luong_Ton": 99,
  "Ten": "Smart Tivi Cong Samsung 49 inch" }
```

- Trong **package Chuong_JSON**.
- Tạo module **read_json_from_internet_tivi_theo_nhom.py**. Trong đó:
- Gọi hàm đã viết ở bài 1 để lấy nội dung
- Xử lý nội dung => hiển thị kết quả

2.3. Thống kê nhân viên theo đơn vị

✓ **Yêu cầu: Xây dựng chương trình đọc và thống kê nhân viên theo đơn vị:**

- Khi chạy chương trình, sẽ hiển thị thống kê nhân viên theo đơn vị này như sau:
 - Tên công ty, địa chỉ, tổng số nhân viên
 - Thống kê nhân viên: tên đơn vị, số nhân viên, tỷ lệ % so với tổng số nhân viên



```
Tên công ty: Công ty Dịch vụ Hoàng hôn Sớm
Địa chỉ: 11223 Trần hưng Đạo Q.1 TP HCM
Tổng số nhân viên: 96
--- Thống kê số nhân viên theo đơn vị ---
1 / Tên đơn vị: Đơn vị A1
    - Số nhân viên: 14
    - Tỷ lệ: 14.58 %
2 / Tên đơn vị: Đơn vị A2
    - Số nhân viên: 15
    - Tỷ lệ: 15.62 %
3 / Tên đơn vị: Đơn vị B1
    - Số nhân viên: 14
    - Tỷ lệ: 14.58 %
4 / Tên đơn vị: Đơn vị B2
    - Số nhân viên: 20
    - Tỷ lệ: 20.83 %
5 / Tên đơn vị: Đơn vị B3
    - Số nhân viên: 19
```

✓ **Tóm tắt yêu cầu**

▪ **Qui tắc xử lý:**

- Cung cấp tập tin: QLCT_1.json
- Đọc và hiển thị thống kê nhân viên như định dạng trên

✓ **Hướng dẫn**

- File .json có cấu trúc như sau:

```
{'CONG_TY': [{ 'Dia_chi': '11223 Trần hưng Đạo Q.1 TP HCM ',
                'Dien_thoai': '08-83222145',
                'ID': 1,
                'Mail': 'hhsom2016@gmail.com',
                'Muc_Luong_Toi_thieu': 3500000,
                'Ten': 'Công ty Dịch vụ Hoàng hôn Sớm',
                'Tuoi_Toi_da': 50,
                'Tuoi_Toi_thieu': 20}],
'DON_VI': [{ 'ID': 1,
              'ID_CHI_NHANH': 1,
              'So_Nhan_vien': '14',
              'Ten': 'Đơn vị A1',
              'Ty_le': '14.58'},
            { 'ID': 2,
              'ID_CHI_NHANH': 1,
```



```

        'So_Nhan_vien': '15',
        'Ten': 'Đơn vị A2',
        'Ty_le': '15.62'},
        ...
    ]
}

```

- Trong **package Chuong_JSON**:
- Tạo module **read_json_file_thong_ke.py**. Trong đó:
 - Viết function `doc_noi_dung_json(filename)`: Đọc và trả về nội dung file JSON:

```

import json

def doc_noi_dung_json(filename):
    data_file = open(filename, encoding = "utf-8")
    data = json.load(data_file)
    data_file.close()
    return data

```

- Gọi function đã viết để lấy nội dung
- Xử lý nội dung => hiển thị kết quả

2.4. Ghi dữ liệu quản lý giao dịch vào tập tin tập tin JSON

✓ Yêu cầu: Xây dựng chương trình ghi các giao dịch vào tập tin JSON:

- Sử dụng lại bài quản lý giao dịch đã làm.
- Sau khi người dùng không tiếp tục thực hiện giao dịch nữa thì hỏi người dùng có muốn ghi vào tập tin hay không? Nếu người dùng chọn 1: Có => ghi vào tập tin. 0: Không => Không ghi
- Khi người dùng chọn ghi:
 - Lấy thông tin ngày hiện tại theo định dạng: nam-thang-ngay-gio-phut-giay
 - Lưu danh sách các giao dịch tiền và vàng vào tập tin nam-thang-ngay-gio-phut-giay.json



```

Quản lý giao dịch:
Nhập mã GD:      gd001
Nhập ngày GD:    21-06-2017
Nhập số lượng:   200
Chọn loại giao dịch: 1: Vàng, 2: Tiền Tệ:      1
Chọn loại: 18k / 24k / 9999:      24k
Nhập đơn giá:    33520000
gd001 - 21-06-2017 - 24k - 200 - 33520000 - Thành tiền = 6704000000
Tổng số lượng: 200
Tổng số tiền: 6704000000
Bạn muốn tiếp tục giao dịch? 1: Có, 0: Không      1
Nhập mã GD:      gd002
Nhập ngày GD:    20-06-2017
Nhập số lượng:   500
Chọn loại giao dịch: 1: Vàng, 2: Tiền Tệ:      2
Chọn loại: USD / EUR / AUD:      USD
Nhập tỷ giá:      22650
Bạn mua hay bán? 1: mua, 0: bán:      1
GD mua: gd002 - 20-06-2017 - USD - 500 - 22650 - Thành tiền = 11325000
Tổng số lượng: 500
Tổng số tiền: 11325000
Bạn muốn tiếp tục giao dịch? 1: Có, 0: Không      0
Bạn có muốn ghi nội dung vào file .json? 1: Có, 0: Không      1
Đã ghi nội dung vào tập tin: 2017-06-21-04-33-26.json
    
```

- Kiểm tra lại nội dung sau khi ghi:

```

Nhập tên tập tin:
2017-06-21-04-33-26.json
Nội dung tập tin:
{'giao_dich_tien': [{'don_gia': '22650',
                      'loai': 'USD',
                      'ma': 'gd002',
                      'mua': 'True',
                      'ngay': '20-06-2017',
                      'so_luong': '500'}],
 'giao_dich_vang': [{'don_gia': '33520000',
                      'loai': '24k',
                      'ma': 'gd001',
                      'ngay': '21-06-2017',
                      'so_luong': '200'}]}
    
```

✓ Tóm tắt yêu cầu

- Qui tắc xử lý:

- Ghi các giao dịch theo định dạng vào file json: nam-thang-ngay-gio-phut-giay.json

✓ Hướng dẫn

- File .json sẽ có cấu trúc như sau:

```

2017-06-21-04-33-26.json
1  {
2    "giao_dich_tien": [
3      {
4        "mua": "True",
5        "loai": "USD",
6        "ngay": "20-06-2017",
7        "don_gia": "22650",
8        "so_luong": "500",
9        "ma": "gd002"
10     }
11   ],
12   "giao_dich_vang": [
13     {
14       "loai": "24k",
15       "ngay": "21-06-2017",
16       "ma": "gd001",
17       "so_luong": "200",
18       "don_gia": "33520000"
19     }
20   ]
21 }

```

- Trong **package Chuong_JSON**:
- Copy module **quan_ly_giao_dich.py** sau đó bổ sung thêm chức năng ghi file JSON:



```
ghi_file = eval(input("Bạn có muốn ghi nội dung vào file .json? 1: Có, 0: Không \t"))
if ghi_file == 1:
    data = {}
    data['giao_dich_vang'] = []
    for item in list_vang:
        data['giao_dich_vang'].append({
            'ma': item.ma,
            'ngay': item.ngay,
            'don_gia': str(item.don_gia),
            'so_luong': str(item.so_luong),
            'loai': str(item.loai)
        })
    data['giao_dich_tien'] = []
    for item in list_tien:
        data['giao_dich_tien'].append({
            'ma': item.ma,
            'ngay': item.ngay,
            'don_gia': str(item.don_gia),
            'so_luong': str(item.so_luong),
            'loai': str(item.loai),
            'mua': str(item.mua)
        })

    thoi_gian_ghi = strftime("%Y-%m-%d-%H-%M-%S", gmtime())
    ten_tap_tin = thoi_gian_ghi + ".json"
    f = open(ten_tap_tin, 'w')
    json.dump(data, f, indent=4)
    f.close()
    print("Đã ghi nội dung vào tập tin:", ten_tap_tin)
```

2.5. Thêm thông tin phòng karaoke

- ✓ **Yêu cầu:** Xây dựng chương trình ghi thêm thông tin phòng karaoke vào tập tin .json:

- Ghi thêm thông tin phòng karaoke vào tập tin quan_ly_phong_karaoke.json.
- Người dùng nhập thông tin của một phòng karaoke: gồm id, ten, mã số, id loại phòng (1 hoặc 2 hoặc 3), đơn giá, số khách tối đa, trạng thái (CON_TRONG, CO_KHACH).
- Ghi thông tin vừa nhập vào file.
- Thông báo kết quả ghi
- Xuất nội dung tập tin .json sau khi ghi.

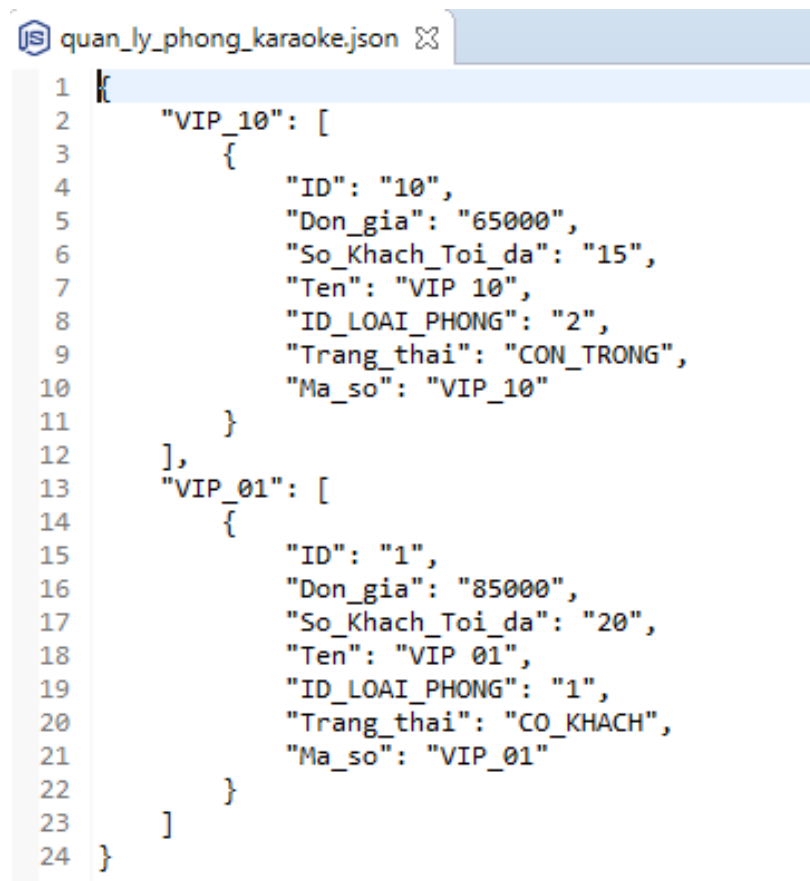
```

Nhập thông tin phòng Karaoke
Nhập ID: 1
Nhập Tên: VIP 01
Nhập mã số: VIP_01
Nhập ID loại phòng (1 hoặc 2 hoặc 3): 1
Nhập đơn giá: 85000
Nhập Số khách tối đa: 20
Nhập trạng thái (CON TRONG hoặc CO KHACH): CO_KHACH
Đã ghi xong
{'VIP_01': [{'Don_gia': '85000',
              'ID': '1',
              'ID_LOAI_PHONG': '1',
              'Ma_so': 'VIP_01',
              'So_Khach_Toai_da': '20',
              'Ten': 'VIP 01',
              'Trang_thai': 'CO_KHACH'}]},
{'VIP_10': [{'Don_gia': '65000',
              'ID': '10',
              'ID_LOAI_PHONG': '2',
              'Ma_so': 'VIP_10',
              'So_Khach_Toai_da': '15',
              'Ten': 'VIP 10',
              'Trang_thai': 'CON TRONG'}]}}

```

✓ Hướng dẫn

- Tập tin quan_ly_phong_karaoke.json được cung cấp sẵn có cấu trúc như sau:



```

quan_ly_phong_karaoke.json
1  [
2    "VIP_10": [
3      {
4        "ID": "10",
5        "Don_gia": "65000",
6        "So_Khach_Toai_da": "15",
7        "Ten": "VIP 10",
8        "ID_LOAI_PHONG": "2",
9        "Trang_thai": "CON TRONG",
10       "Ma_so": "VIP_10"
11     }
12   ],
13   "VIP_01": [
14     {
15       "ID": "1",
16       "Don_gia": "85000",
17       "So_Khach_Toai_da": "20",
18       "Ten": "VIP 01",
19       "ID_LOAI_PHONG": "1",
20       "Trang_thai": "CO_KHACH",
21       "Ma_so": "VIP_01"
22     }
23   ]
24 ]

```

- Trong **package Chuong_JSON**:
- Tạo module **them_phong_karaoke.py** thực hiện công việc ghi thêm phòng karaoke vào file JSON.
 - Chú ý: chú ý cấu trúc của một phần tử trong tập tin JSON để thêm vào cho đúng.

```

6 import json
7 from pprint import pprint
8
9 print("Nhập thông tin phòng Karaoke")
10 id1 = input("Nhập ID:\t")
11 ten = input("Nhập Tên:\t")
12 ma_so = input("Nhập mã số:\t")
13 id_loai_phong = input("Nhập ID Loại phòng (1 hoặc 2 hoặc 3):\t")
14 don_gia = input("Nhập đơn giá:\t")
15 so_khach_toi_da = input("Nhập Số khách tối đa:\t")
16 trang_thai = input("Nhập trạng thái (CON TRONG hoặc CO KHACH):\t")
17
18 phong = {"ID":id1, "Ten":ten, "Ma_so":ma_so, "ID_LOAI_PHONG":id_loai_phong, "Don_gia":don_gia, \
19          "So_Khach_Toai_da":so_khach_toi_da, "Trang_thai" : trang_thai}
20
21
22 list1 = []
23 list1.append(phong)
24 new = {ma_so:list1}
25
26 f = open('quan_ly_phong_karaoke.json', encoding = "utf-8")
27 data = json.load(f)
28 data.update(new)
29
30 f = open('quan_ly_phong_karaoke.json', 'w', encoding = "utf-8")
31 json.dump(data, f, indent=4)
32 f.close()
33 print("Đã ghi xong")
34
35 f = open('quan_ly_phong_karaoke.json', encoding = "utf-8")
36 data = json.load(f)
37 pprint(data)
    
```


BÀI 3: Làm việc với tập tin XML



Mục tiêu chính:

- *Làm việc với tập tin XML: dùng SAX API đọc, hiển thị tập tin XML*
- *Làm việc với tập tin XML: dùng DOM API đọc, hiển thị, ghi tập tin XML*

3.1. Đọc và hiển thị danh bạ điện thoại với SAX API

✓ **Yêu cầu: Xây dựng chương trình hiển thị danh bạ điện thoại trong tập tin XML:**

- Thông tin của contact được lưu trữ trong tập tin contact.xml.
- Khi chạy chương trình, sẽ hiển thị danh sách các contact nằm trong tập tin này như sau:

```
List of contacts:
--- Contact 1 ---
Name: Anh Thư
Phone: 0989741258
List of contacts:
--- Contact 2 ---
Name: Thanh Thủy
Phone: 0913258963
List of contacts:
--- Contact 3 ---
Name: Kim Cương
Phone: 0934369147
List of contacts:
--- Contact 4 ---
Name: Sơn Nam
Phone: 0915654789
List of contacts:
--- Contact 5 ---
Name: Mai Quý
Phone: 0982654123
List of contacts:
--- Contact 6 ---
Name: Tuấn Thanh
Phone: 0915369231
List of contacts:
--- Contact 7 ---
Name: Chu Vân
Phone: 0934852852
List of contacts:
--- Contact 8 ---
Name: Khang Khang
Phone: 0907412365
List of contacts:
--- Contact 9 ---
Name: Mai Uyên
Phone: 0909741147
List of contacts:
--- Contact 10 ---
Name: Thiên Tuấn
Phone: 0989636363
```



✓ **Tóm tắt yêu cầu**

▪ **Qui tắc xử lý:**

- Cung cấp tập tin contact.xml (có sẵn các contact)
- Đọc và hiển thị các contact như định dạng trên

✓ **Hướng dẫn**

- Trong project Python_nang_cao, tạo **package Bai3**. Trong đó:
- Tập tin contact.xml có cấu trúc như sau:

```
<?xml version="1.0" encoding="UTF-8"?>
<contacts shelf="Danh sách Contact">
    <contact phone="0989741258" name="Anh Thu" />
    <contact phone="0913258963" name="Thanh Thủy" />
    <contact phone="0934369147" name="Kim Cương" />
    <contact phone="0915654789" name="Sơn Nam" />
    <contact phone="0982654123" name="Mai Quý" />
    <contact phone="0915369231" name="Tuấn Thanh" />
    <contact phone="0934852852" name="Chu Vân" />
    <contact phone="0907412365" name="Khang Khang" />
    <contact phone="0909741147" name="Mai Uyên" />
    <contact phone="0989636363" name="Thiên Tuấn" />
</contacts>
```

- Tạo module **read_contacts.py**. Trong đó:
- Đọc và hiển thị danh sách contact:

```
import xml.sax

class StudentHandler(xml.sax.ContentHandler):
    sl = 0;

    def __init__(self):
        self.CurrentData = ""
        self.phone = ""
        self.name = ""

    # Call when an element starts
    def startElement(self, tag, attributes):
        self.CurrentData = tag
```



```

print("List of contacts:")

if tag == "contact":
    print ("--- Contact ", StudentHandler.sl, "---")
    name = attributes["name"]
    print ("Name:", name)
    phone = attributes["phone"]
    print ("Phone:", phone)
    StudentHandler.sl += 1;

if (__name__ == "__main__"):
    # create an XMLReader
    parser = xml.sax.make_parser()
    # turn off namespaces
    parser.setFeature(xml.sax.handler.feature_namespaces, 0)

    # override the default ContextHandler
    Handler = StudentHandler()
    parser.setContentHandler(Handler)

    parser.parse("contact.xml")
    
```

3.2. Đọc và hiển thị danh sách nhân viên với DOM XML

✓ **Yêu cầu: Xây dựng chương trình hiển thị danh sách nhân viên như sau:**

- Mở tập tin don_vi.xml
- Đọc và đưa các đơn vị lấy được từ tập tin xml vào danh sách đơn vị
- Mở tập tin nhan_vien.xml
- Đọc và thông tin nhân viên lấy được từ tập tin xml vào danh sách nhân viên.
- Cho người dùng chọn đơn vị => hiển thị danh sách nhân viên tương ứng với đơn vị được chọn, thống kê tổng số nhân viên, tổng số nam, tổng số nữ
- Cho người dùng nhập một tên => tìm tương đối trong toàn đơn vị xem có ai có tên này thì in ra.



--- Danh sách đơn vị ---

- 1 - Đơn vị A
- 2 - Đơn vị B
- 3 - Đơn vị C
- 4 - Đơn vị D
- 5 - Đơn vị E

Bạn muốn xem thống kê đơn vị nào (nhập số)? 2

---Kết quả thống kê---

- 31 - Trần tiến Quân - 011468433
- 32 - Lê minh Khang - 052250493
- 33 - Dương thanh Bình - 013496480
- 34 - Trương phước thảo Trúc - 071576262
- 35 - Mai đình Dương - 077321551
- 36 - Lâm văn Thiết - 065515564
- 37 - Bàng bùi Trường - 040383605
- 38 - Cao kiến đồ Quyền - 037068744
- 39 - Bạch lê Khánh - 072256152
- 40 - Lý đình hải Bích - 071635073
- 41 - Lê sơn Đông - 089913729
- 42 - Đỗ bùi Tâm - 004839602
- 43 - Trần duy Lan - 010570899
- 44 - Bàng bùi Trường - 019209897
- 45 - Trần lê bạch Mai - 074732263
- 46 - Đỗ bích Tế - 057959424
- 47 - Hứa đình Trường - 096535270
- 48 - Bùi lê Bình - 072759534
- 49 - Phạm bảo Khiêm - 090500821
- 50 - Bạch kiến Thiết - 059367222
- 51 - Trang quyết Nguyên - 031786681
- 52 - Bùi hồng Bình - 067100471
- 53 - Phi tắc tuần Tài - 097735542
- 54 - Phạm thế Tạo - 039814616
- 55 - Triệu đình Quân - 081948710
- 56 - Đỗ phước Trang - 068600331
- 57 - Triệu bích thảo Nguyên - 098270145
- 58 - Đình văn hải Bích - 098366371
- 59 - Mai phạm Tạo - 085616797

Tổng số nhân viên: 29 - Trong đó có: 20 nam, và 9 nữ

Nhập tên nhân viên cần tìm: sơn

--- Kết quả tìm kiếm ---

- 2 - Đỗ sơn thụy Trang - 083239154
- 13 - Trần ngọc hải Sơn - 018992163
- 29 - Trần quyết sơn Trúc - 089531662
- 41 - Lê sơn Đông - 089913729
- 72 - Trần sơn Viễn - 039888062



Nhập tên nhân viên cần tìm: phương
 --- Kết quả tìm kiếm ---
 Không tìm thấy NV nào!

✓ **Hướng dẫn sử dụng:**

- Chọn một đơn vị => Hiển thị thông tin chi tiết của các nhân viên thuộc đơn vị được chọn.
- Nhập tên => Hiển thị danh sách các nhân viên có tên tương đối giống tên cần tìm

✓ **Tóm tắt yêu cầu**

▪ **Nhập:**

- Chọn Đơn vị
- Tên cần tìm

▪ **Xuất:**

- Thông tin nhân viên thuộc đơn vị được chọn
- Tổng số nhân viên
- Tổng số nam/nữ
- Kết quả tìm kiếm

✓ **Hướng dẫn**

- Tập tin don_vi.xml có cấu trúc như sau:

```
<?xml version="1.0" encoding="UTF-8"?>
<DON_VI_S shelf="Danh sách đơn vị">
  <DON_VI ID="1" Ten="Đơn vị A" />
  <DON_VI ID="2" Ten="Đơn vị B" />
  <DON_VI ID="3" Ten="Đơn vị C" />
  <DON_VI ID="4" Ten="Đơn vị D" />
  <DON_VI ID="5" Ten="Đơn vị E" />
</DON_VI_S>
```

- Tập tin nhan_vien.xml có cấu trúc như sau:

```
<?xml version="1.0" encoding="UTF-8"?>
<NHAN_VIEN_S shelf="New Arrivals">
  <NHAN_VIEN ID="1" Ho_ten="Phí quyết Nguyên" Gioi_tinh="true" Ngay_sinh="31/08/1973" CMND="060605210" Muc_luong="5200000" Dia_chi="
  <NHAN_VIEN ID="2" Ho_ten="Đỗ sơn thủy Trang" Gioi_tinh="false" Ngay_sinh="14/03/1979" CMND="083239154" Muc_luong="8400000" Dia_chi="
  <NHAN_VIEN ID="3" Ho_ten="Triệu thị trang Đài" Gioi_tinh="false" Ngay_sinh="26/04/1990" CMND="005487440" Muc_luong="5000000" Dia_chi="
  <NHAN_VIEN ID="4" Ho_ten="Bùi nam quang Viễn" Gioi_tinh="false" Ngay_sinh="21/05/1993" CMND="092700391" Muc_luong="5800000" Dia_chi="
```

- Trong đó, 1 nhân viên có thông tin như sau:

```
<NHAN_VIEN ID="1" Ho_ten="Phí quyết Nguyên" Gioi_tinh="true" Ngay_sinh="31/08/1973"
CMND="060605210" Muc_luong="5200000" Dia_chi="424 C Phường 4- Quận 10- Tỉnh/Tp
CC&#xD;&#xA;" ID_DON_VI="1" />
```

- Trong **package Bai2**:
- Tạo module **class_don_vi.py**. Trong đó, tạo lớp DonVi (đơn vị) và NhanVien (nhân viên) để xử lý:



```
class DonVi(object):
    """
    classdocs: Class Don Vi
    """

    def __init__(self, iddv, ten):
        """
        Constructor
        """
        self.iddv = iddv
        self.ten = ten

class NhanVien(object):
    """
    classdocs: Class Nhân viên
    """

    def __init__(self, idnv, ho_ten, gioi_tinh, ngay_sinh, cmnd, muc_luong, dia_chi, iddv):
        """
        Constructor
        """
        self.idnv = idnv
        self.ho_ten = ho_ten
        self.gioi_tinh = gioi_tinh
        self.ngay_sinh = ngay_sinh
        self.cmnd = cmnd
        self.muc_luong = muc_luong
        self.dia_chi = dia_chi
        self.iddv = iddv

    def in_thong_tin(self):
        gioi_tinh = ""
        if self.gioi_tinh == "true":
            gioi_tinh = "Nam"
        else:
            gioi_tinh = "Nữ"

        return self.idnv + " - " + self.ho_ten + " - " + gioi_tinh + " - " + self.ngay_sinh + \
            " - " + self.cmnd + " - " + self.muc_luong + " - " + self.dia_chi
```

- Tạo module **thong_ke_don_vi.py**. Trong đó:
- Viết hàm `tao_danh_sach_don_vi(list_don_vi)` : Đọc file `don_vi.xml` lấy dữ liệu đưa vào danh sách đơn vị:

```
def tao_danh_sach_don_vi(list_don_vi):
    DOMTree = xml.dom.minidom.parse("don_vi.xml")
    collection = DOMTree.documentElement
    don_vi_s = collection.getElementsByTagName("DON_VI")

    for don_vi in don_vi_s:
        if don_vi.hasAttribute("ID") and don_vi.hasAttribute("Ten"):
```



```

        dv =
chapter2.class_don_vi.DonVi(don_vi.getAttribute("ID"),
don_vi.getAttribute("Ten"))

        list_don_vi.append(dv)

    return list_don_vi

```

- Viết hàm tao_danh_sach_nhan_vien(list_nhan_vien): Đọc file nhan_vien.xml lấy dữ liệu đưa vào danh sách nhân viên tương tự như đơn vị.
- Viết hàm thong_ke_don_vi(iddv, list_nhan_vien): để thống kê nhân viên theo đơn vị
- Viết hàm tim_kiem_nhan_vien(ten, list_nhan_vien): để tìm kiếm nhân viên
- Gọi các hàm đã viết để thực hiện các yêu cầu

3.3. Đọc - Ghi tập tin XML với DOM

✓ **Yêu cầu: Xây dựng chương trình đọc - ghi CD với XML DOM:**

```

Bạn muốn làm gì? 1: Thêm CD mới, 2: Xem danh sách CD      1
Nhập tên CD:      Chat với Mozart
Nhập tên ca sỹ: Mỹ Linh
Nhập số bài hát:      6
Giá thành:      185000
Đã thêm CD
Bạn muốn tiếp tục không? 1: Có, 2: Không1
Bạn muốn làm gì? 1: Thêm CD mới, 2: Xem danh sách CD      2
---- CD -----
Tên CD: Happy new year
Ca sỹ: ABBA
Format: 10
Rating: 25000
---- CD -----
Tên CD: Ngày không mưa
Ca sỹ: Hồng Nhung
Format: 10
Rating: 250000
---- CD -----
Tên CD: Xuân 2017
Ca sỹ: Nhiều ca sỹ
Format: 8
Rating: 150000
---- CD -----
Tên CD: Chat với Mozart
Ca sỹ: Mỹ Linh
Format: 6
Rating: 185000
Bạn muốn tiếp tục không? 1: Có, 2: Không

```

- Ghi thông tin 1 CD vào tập tin cd.xml.



- Đọc nội dung cd.xml sau khi thực hiện việc ghi và xuất ra màn hình
 - Khi chạy chương trình, hỏi người dùng xem họ muốn làm gì:
 - 1 – Thêm CD vào tập tin cd.xml
 - 2 – Hiển thị danh sách CD trong cd.xml
- (có thể lặp lại công việc này nếu như muốn làm nhiều lần)

✓ **Tóm tắt yêu cầu**

▪ **Nhập:**

- Thông tin CD

▪ **Xuất:**

- Danh sách các CD

✓ **Hướng dẫn**

- Trong package Bai3:
- Tạo tập tin cd.xml có cấu trúc như sau:

```
<?xml version="1.0" ?><!--Sample XML Document-->
<cds>
    <cd ten="Happy new year">
        <ca_sy>ABBA</ca_sy>
        <so_bai_hat>10</so_bai_hat>
        <gia_thanh>25000</gia_thanh>
    </cd>
    ...
</cds>
```

- Tạo module read_write_cd_dom.py. Trong đó:
- Xây dựng hàm ghi cd vào file xml:

```
def make_xml(ten_file, cd):

    if (os.path.isfile(ten_file)):
        doc = xml.dom.minidom.parse(ten_file)
        root_xml = doc.documentElement
    else:
        doc = Document()
        root_xml = doc.createElement('cds')
        doc.appendChild(root_xml)

    child_node = doc.createElement('cd')

    child_node.setAttribute('ten', cd.ten)
```




```

root_xml.appendChild(child_node)

ca_sy = doc.createElement('ca_sy')
ca_sy.appendChild(doc.createTextNode(cd.ca_sy))
child_node.appendChild(ca_sy)

so_bai_hat = doc.createElement('so_bai_hat')
so_bai_hat.appendChild(doc.createTextNode(cd.so_bai_hat))
child_node.appendChild(so_bai_hat)

gia_thanh = doc.createElement('gia_thanh')
gia_thanh.appendChild(doc.createTextNode(cd.gia_thanh))
child_node.appendChild(gia_thanh)

return doc

```

- Xây dựng hàm đọc danh sách CD từ file XML

```

def read_xml(ten_file):
    DOMTree = xml.dom.minidom.parse(ten_file)
    collection = DOMTree.documentElement

    cds = collection.getElementsByTagName("cd")

    for cd in cds:
        print ("---- CD ----")
        if cd.hasAttribute("ten"):
            print ("Tên CD: %s" % cd.getAttribute("ten"))
        ca_sy = cd.getElementsByTagName('ca_sy')[0]
        print ("Ca sỹ: %s" % ca_sy.childNodes[0].data)
        so_bai_hat = cd.getElementsByTagName('so_bai_hat')[0]
        print ("Format: %s" % so_bai_hat.childNodes[0].data)
        gia_thanh = cd.getElementsByTagName('gia_thanh')[0]
        print ("Rating: %s" % gia_thanh.childNodes[0].data)

    return

```

- Gọi sử dụng các hàm đã xây dựng:

```

if __name__ == '__main__':

```



```

tiếp_tuc = 1
while tiếp_tuc == 1:
    i = int(input("Bạn muốn làm gì? 1: Thêm CD mới, 2: Xem danh sách CD\t"))
    if i == 1:
        ten = input("Nhập tên CD:\t")
        ca_sy = input("Nhập tên ca sỹ:\t")
        so_bat_hat = input("Nhập số bài hát:\t")
        gia_thanh = input("Giá thành:\t")

        cd = CD(ten, ca_sy, so_bat_hat, gia_thanh)
        make_xml('cd.xml', cd).writexml(open(file='cd.xml', mode='w', encoding='utf8'), indent='', addindent='', newl='')
        print("Đã thêm CD")
    elif i == 2:
        read_xml('cd.xml')
    tiếp_tuc = int(input("Bạn muốn tiếp tục không? 1: Có, 0: Không\t"))

```

BÀI 4: Xây dựng ứng dụng với CSDL



Mục tiêu chính: trang bị kiến thức và kỹ năng

- Hiện thực được ứng dụng kết nối đến CSDL SQLite
- Xây dựng ứng dụng CRUD

4.1. Xây dựng ứng dụng Quản lý sản phẩm

✓ **Yêu cầu:** Xây dựng ứng dụng quản lý sản phẩm như sau:

- Tạo CSDL product.db
- Trong CSDL product.db, tạo bảng sản phẩm (product) có cấu trúc như sau:

Tên cột	Loại dữ liệu	Thông tin thêm
Id	Integer	Primary key
Name	Text	Not null
Price	Real	Not null
Amount	Integer	Not null

Chương trình cho phép người dùng:

- Hiển thị danh sách sản phẩm
- Thêm các sản phẩm vào bảng sản phẩm
- Tìm kiếm thông tin sản phẩm theo tên
- Cập nhật đơn giá và số lượng của một sản phẩm theo id cụ thể
- Xóa một sản phẩm theo id cụ thể

(Mở rộng: các công việc này có thể lặp lại nhiều lần)

Kết quả:

- Khi người dùng chọn thêm:

```
Bạn muốn làm gì?
1: Hiển thị danh sách sản phẩm
2: Thêm sản phẩm mới
3: Tìm kiếm sản phẩm theo tên
4: Cập nhật sản phẩm
5: Xóa sản phẩm 2
Nhập tên sản phẩm:      Kem đánh răng PS
Nhập đơn giá:    25000
Nhập số lượng:   30
Thêm mới thành công
Bạn muốn tiếp tục hay không? 1: Có, 0: Không    1
```



- Khi người dùng chọn hiển thị:

Bạn muốn làm gì?

1: Hiển thị danh sách sản phẩm

2: Thêm sản phẩm mới

3: Tìm kiếm sản phẩm theo tên

4: Cập nhật sản phẩm

5: Xóa sản phẩm 1

ID = 1

NAME = Táo xanh Mỹ

PRICE = 65000.0

AMOUNT = 50

ID = 2

NAME = Nho Mỹ không hạt

PRICE = 255000.0

AMOUNT = 10

ID = 3

NAME = Nho Mỹ không hạt

PRICE = 195000.0

AMOUNT = 10

ID = 4

NAME = Xà phòng Lux

PRICE = 7500.0

AMOUNT = 50

ID = 5

NAME = Kem đánh răng PS

PRICE = 25000.0

AMOUNT = 30

Đọc và hiển thị thành công

Bạn muốn tiếp tục hay không? 1: Có, 0: Không

- Khi người dùng chọn tìm kiếm:

Bạn muốn làm gì?

1: Hiển thị danh sách sản phẩm

2: Thêm sản phẩm mới

3: Tìm kiếm sản phẩm theo tên

4: Cập nhật sản phẩm

5: Xóa sản phẩm 3

Nhập tên sản phẩm: táo

Kết quả tìm kiếm

ID = 1

NAME = Táo xanh Mỹ

PRICE = 65000.0

AMOUNT = 50

Bạn muốn tiếp tục hay không? 1: Có, 0: Không

- Khi người dùng chọn cập nhật:



Bạn muốn làm gì?

1: Hiển thị danh sách sản phẩm

2: Thêm sản phẩm mới

3: Tìm kiếm sản phẩm theo tên

4: Cập nhật sản phẩm

5: Xóa sản phẩm 4

ID của sản phẩm cần cập nhật: 4

Nhập đơn giá mới: 7800

Số lượng mới: 25

Cập nhật thành công

Bạn muốn tiếp tục hay không? 1: Có, 0: Không

– Khi người dùng chọn xóa:

Bạn muốn làm gì?

1: Hiển thị danh sách sản phẩm

2: Thêm sản phẩm mới

3: Tìm kiếm sản phẩm theo tên

4: Cập nhật sản phẩm

5: Xóa sản phẩm 5

ID của sản phẩm cần xóa: 1

Xoá thành công

✓ Hướng dẫn sử dụng:

- Chạy ứng dụng, cho phép người dùng chọn 1 trong các chức năng sau:
- 1. Hiển thị danh sách sản phẩm
- 2. Thêm sản phẩm mới
- 3. Tìm kiếm sản phẩm theo tên
- 4. Cập nhật đơn giá và số lượng của một sản phẩm
- 5. Xóa một sản phẩm

✓ Tóm tắt yêu cầu

▪ Nhập:

- 2. Thông tin sản phẩm: tên, đơn giá, số lượng
- 3. Tên sản phẩm
- 4. Id muốn cập nhật
- 5. Id muốn xóa

▪ Xuất:

- Theo yêu cầu trên

✓ Hướng dẫn

- Trong project Python_nang_cao tạo **package Bai4**:
- Trong package Bai4: tạo module 4_1_quan_ly_san_pham.py. Trong đó:
- Áp dụng các hàm đã được giới thiệu trong bài học để thực hiện ứng dụng này



- Gợi ý các truy vấn:

- Tạo CSDL:

```
conn = sqlite3.connect('product.db')
```

- Truy vấn tạo bảng

```
sql = '''
CREATE TABLE product (
    id INTEGER PRIMARY KEY,
    name TEXT NOT NULL,
    price REAL NOT NULL,
    amount INTEGER NOT NULL
);'''
```

- Truy vấn lấy danh sách:

```
sql = "SELECT * FROM product"
```

- Truy vấn cập nhật:

```
sql = ''' UPDATE PRODUCT
           SET price = ?, amount = ?
           WHERE id = ?'''
```

- Truy vấn xóa:

```
sql = ''' DELETE FROM PRODUCT
           WHERE id = ?'''
```

- Lần lượt thực hiện các công việc theo yêu cầu.

4.2. Xây dựng ứng dụng Quản lý nhân viên

- ✓ **Yêu cầu:** Xây dựng ứng dụng quản lý nhân viên như sau:

- Tạo CSDL ql_nhan_vien.db

```
import sqlite3

conn = sqlite3.connect(r"ql_nhan_vien.db")
print ("New database is created!")
```

- Trong CSDL ql_nhan_vien.db, tạo bảng phòng ban (PHONG) có cấu trúc như sau:

Tên cột	Loại dữ liệu	Thông tin thêm
---------	--------------	----------------



id	Integer	Primary key
ten	Text	Not null
chuc_nang	Text	Not null

- Cú pháp:

```
sql = '''
CREATE TABLE PHONG (
    id INTEGER PRIMARY KEY,
    ten TEXT NOT NULL UNIQUE,
    chuc_nang TEXT NOT NULL
);'''
```

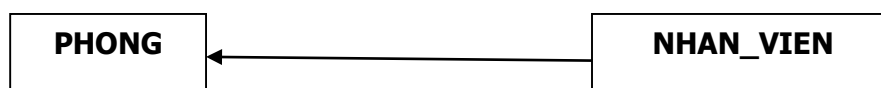
- Thêm các phòng ban vào bảng phòng

```
sql1 = "INSERT INTO PHONG (ten,chuc_nang) \
VALUES ('Hành chính', 'Giải quyết các công việc hành chính của công ty' )"
sql2 = "INSERT INTO PHONG (ten,chuc_nang) \
VALUES ('Kỹ thuật', 'Thực hiện các dự án kỹ thuật của công ty' )"
```

- Trong CSDL ql_nhan_vien.db, tạo bảng nhân viên (NHAN_VIEN) có cấu trúc như sau: có Id_phong là khóa ngoại (1 phòng có nhiều nhân viên, 1 nhân viên chỉ thuộc về một phòng)

Tên cột	Loại dữ liệu	Thông tin thêm
id	Integer	Primary key
ho_ten	Text	Not null
tuoi	Integer	Not null
dia_chi	Text	Not null
luong	Real	Not null
Id_phong	Integer	Foreign key – tham chiếu tới id của bảng PHONG

Sơ đồ quan hệ:





```
sql = '''CREATE TABLE NHAN_VIEN(
    id INTEGER PRIMARY KEY,
    ho_ten TEXT NOT NULL,
    tuoi INTEGER NOT NULL,
    dia_chi TEXT NOT NULL,
    Luong REAL,
    phong_id INTEGER NOT NULL,
    FOREIGN KEY(phong_id) REFERENCES PHONG(id));'''
```

- Thêm các nhân viên vào theo phòng

```
sql = "INSERT INTO NHAN_VIEN (ho_ten, tuoi, dia_chi, luong, phong_id) \
VALUES ('Trần Minh', 32, 'Quận 1', 8500000.00, 1)"
...
```

- Liệt kê danh sách nhân viên theo phòng

```
import sqlite3

conn = sqlite3.connect('ql_nhan_vien.db')
print ("Opened database successfully")

phong = conn.execute("SELECT * from PHONG")
nhan_vien = conn.execute("SELECT * from NHAN_VIEN")

list_nv = []
list_phong = []
for row in phong:
    list_phong.append(row)
for row1 in nhan_vien:
    list_nv.append(row1)

for row in list_phong:
    print ("---- ID = ", row[0], "- Tên:", row[1])
    for nv in list_nv:
        if nv[5] == row[0]:
            print ("ID = ", nv[0], "- Tên:", nv[1], "- Tuổi:", nv[2], "- 
Địa chỉ:", nv[3], "- Lương:", nv[4])

print ("Operation done successfully")
```




```
conn.commit()
conn.close()
```

✓ **Hướng dẫn sử dụng:**

- Chạy ứng dụng, cho phép người dùng chọn 1 trong các chức năng sau:
- 1. Hiển thị danh sách phòng
- 2. Thêm phòng mới
- 3. Thêm nhân viên theo phòng
- 4. Hiển thị danh sách nhân viên theo phòng

✓ **Tóm tắt yêu cầu**

▪ **Nhập:**

- 2. Thông tin phòng: tên, chức năng
- 3. Id Phòng, Thông tin nhân viên: ho_ten, tuoi, dia_chi, luong
- 4. Id phòng

▪ **Xuất:**

- Theo yêu cầu trên

✓ **Hướng dẫn**

- Trong package Bai4, tạo module **4_2_quan_ly_nhan_vien.py**. Trong đó:
- Dựa vào những gợi ý trên, thực hiện các công việc tương tự như bài Quản lý sản phẩm.

BÀI 5: Thread - Multithread



Mục tiêu chính:

- Phân biệt những khái niệm chương trình và tiến trình
- Hiện thực một chương trình có sử dụng thread và sử dụng cơ chế xử lý đồng thời

5.1. Ứng dụng multithread

✓ **Yêu cầu:** Viết chương trình tạo và thực thi nhiều luồng độc lập trên console như sau:

```
Starting Google
Starting Yahoo
Starting Facebook
Facebook: Thu Mar 23 10:40:43 2017
Yahoo: Thu Mar 23 10:40:43 2017
Google: Thu Mar 23 10:40:43 2017
Facebook: Thu Mar 23 10:40:44 2017
Google: Thu Mar 23 10:40:44 2017
Yahoo: Thu Mar 23 10:40:44 2017
Yahoo: Thu Mar 23 10:40:45 2017
Google: Thu Mar 23 10:40:45 2017
Facebook: Thu Mar 23 10:40:45 2017
Facebook: Thu Mar 23 10:40:46 2017
Yahoo: Thu Mar 23 10:40:46 2017
Google: Thu Mar 23 10:40:46 2017
Facebook: Thu Mar 23 10:40:47 2017
Yahoo: Thu Mar 23 10:40:47 2017
Google: Thu Mar 23 10:40:47 2017
Exiting Facebook
Exiting Yahoo
Exiting Google
Exiting Main Thread
```

Kết quả khi thực thi 3 luồng trong 5s

✓ **Hướng dẫn**

- Trong project Python_nang_cao, tạo **package Bai5**.
- Trong package Bai5, tạo module **5_1_thread.py**. Trong đó:
- Tạo lớp MyThread kế thừa từ lớp threading.Thread:

```
import threading
import time

exitFlag = 0
```



```
class MyThread (threading.Thread):
    def __init__(self, threadID, name, counter):
        threading.Thread.__init__(self)
        self.threadID = threadID
        self.name = name
        self.counter = counter

    def run(self):
        print ("Starting " + self.name)
        print_time(self, threadName=self.name, delay=self.counter,
counter=5)
        print ("Exiting " + self.name)

def print_time(self, threadName, delay, counter):
    while counter:
        if exitFlag:
            threadName.exit()
        time.sleep(delay)
        print ("%s: %s" % (threadName, time.ctime(time.time())))
        counter -= 1
```

- Sau đó, tạo và thực thi 3 luồng độc lập như sau:

```
if __name__ == "__main__":
    # Create new threads
    thread1 = MyThread(1, "Google", 1)
    thread2 = MyThread(2, "Yahoo", 1)
    thread2 = MyThread(2, "Facebook", 1)
    # Start new Threads
    thread1.start()
    thread2.start()
    thread1.join()
    thread2.join()
    print ("Exiting Main Thread")
```

5.2. Tính tổng các phần tử trong list

- ✓ **Yêu cầu:** Sử dụng Thread để tính tổng các phần tử trong list



- Chương trình được xây dựng để tính tổng cho một list có n phần tử (n lớn) có giá trị ngẫu nhiên từ 0 - 10
- Yêu cầu tạo ra Thread để tính tổng một list con (1 phần trong list lớn)
- Hãy tạo ra một số Thread để thực hiện việc tính tổng này

```

Nhập số phần tử:      40
List: [7, 3, 5, 4, 7, 4, 6, 2, 1, 7, 7, 7, 4, 7, 0, 7, 2, 2, 4, 5, 6, 0, 2, 5, 8, 0, 5, 4, 7, 3, 4, 1, 0, 0, 9, 6, 5, 0, 3, 9]
Nhập vào số thread:  3
Thread 1 : 7; 3; 5; 4; 7; 4; 6; 2; 1; 7; 7; 7; 4;
Thread 2 : 7; 0; 7; 2; 2; 4; 5; 6; 0; 2; 5; 8; 0;
Thread 3 : 5; 4; 7; 3; 4; 1; 0; 0; 9; 6; 5; 0; 3; 9;
Tổng list= 168
    
```

✓ **Hướng dẫn sử dụng:**

- Nhập vào số phần tử cho list => Tạo list ngẫu nhiên với n phần tử.
- Nhập vào số Thread cần tạo => Tính tổng list dựa trên số Thread này.

✓ **Tóm tắt yêu cầu**

▪ **Nhập:**

- Số phần tử
- Số Thread

▪ **Xuất:**

- Tổng của list

✓ **Hướng dẫn**

- Trong package Bai5, tạo module **5_2_tong_thread.py**. Trong đó:
- Tạo lớp SumThread kế thừa từ threading.Thread để tính tổng mảng con:

```

import threading
import random

class SumThread (threading.Thread):

    def __init__(self, lo, hi, list_nums):
        threading.Thread.__init__(self)
        self.lo = lo
        self.hi = hi
        self.list_nums = list_nums
        self.sum = 0

    def run(self):
        self.set_sum()

    def set_sum(self):
    
```



```

        i = self.lo
        while i < self.hi:
            self.sum += self.list_nums[i]
            i += 1
        return

    def get_sum(self):
        return self.sum

    def print_list(self):
        str_list = ""
        i = self.lo
        while i < self.hi:
            str_list += str(self.list_nums[i]) + "; "
            i += 1
        return str_list

```

- Xây dựng hàm `tinh_tong()` để tính tổng và gọi hàm ở main:

```

def tinh_tong(list_numbers, n_threads):
    tong = 0
    so_pt = len(list_numbers)
    list_threads = []

    i = 0
    while i < n_threads:
        lo = int((i*so_pt)/n_threads)
        hi = int((i+1)*so_pt/n_threads)
        thread = SumThread(lo, hi, list_numbers)
        list_threads.append(thread)
        list_threads[i].start()
        i += 1

    j=0
    while j < n_threads:
        list_threads[j].join()
        tong += list_threads[j].get_sum()
        print("Thread ", j+1, ":", list_threads[j].print_list())

```



```

        j +=1

    return tong

if __name__ == "__main__":
    list_numbers = []
    n = int(input("Nhập số phần tử:\t"))
    i = 0
    while i<n:
        list_numbers.append(random.randrange(10))
        i += 1

    print("List:", list_numbers)
    n_threads = int(input("Nhập vào số thread:\t"))
    sum = tinh_tong(list_numbers, n_threads);
    print("Tổng List=", sum)

```

5.3. Tìm giá trị lớn nhất trong list

✓ **Yêu cầu: Sử dụng Thread để tìm giá trị lớn nhất trong list**

- Chương trình được xây dựng để tìm giá trị lớn nhất trong một list có n phần tử (n lớn) có giá trị ngẫu nhiên từ 0 - 100
- Yêu cầu tạo ra Thread để tìm max trong một list con (1 phần trong list lớn)
- Hãy tạo ra một số Thread để thực hiện việc tìm max này

✓ **Hướng dẫn sử dụng:**

- Nhập vào số phần tử cho list => Tạo list ngẫu nhiên với n phần tử.
- Nhập vào số Thread cần tạo => Giá trị lớn nhất của list dựa trên số Thread này

✓ **Tóm tắt yêu cầu**

▪ **Nhập:**

- Số phần tử
- Số Thread

▪ **Xuất:**

- Giá trị lớn nhất

✓ **Hướng dẫn**

- Trong package Bai5, tạo module **5_3_tim_max.py**. Trong đó:
- Tạo lớp MaxThread kế thừa từ threading.Thread để tìm max mảng con.



- Xây dựng hàm `tim_max()` để tìm max và gọi hàm ở main:
- Làm tương tự bài tính tổng

BÀI 6: Regular Expression



Mục tiêu chính:

- Xây dựng chương trình sử dụng biểu thức chính quy để kiểm tra dữ liệu (validation)

6.1. Kiểm tra chuỗi ký tự

- ✓ **Yêu cầu:** Viết chương trình kiểm tra một chuỗi nhập vào hợp lệ khi chỉ chứa các character (cho phép là a-z, A-Z và 0-9)

Nhập chuỗi: ABCDEFabcdef123450
Hợp lệ

Nhập chuỗi: 123\$%ssaSA&
Không hợp lệ

- ✓ **Tóm tắt yêu cầu**

- **Nhập:**

- Chuỗi

- **Xuất:**

- Chuỗi có đúng mẫu hay không

- ✓ **Hướng dẫn**

- Trong project Python_nang_cao, tạo **package Bai6**.
- Trong package Bai6, tạo module **kiem_tra_chuoi_ky_tu.py**. Trong đó:
- Sử dụng Regular Expression để giải quyết yêu cầu
- Gợi ý:

```
import re

def is_allowed_specific_char(string):
    charRe = re.compile(r'^[a-zA-Z0-9.]')
    string = charRe.search(string)
    return not bool(string)

string = input("Nhập chuỗi: ")
if is_allowed_specific_char(string):
    print("Hợp lệ")
else:
```




```
print("Không hợp lệ")
```

6.2. In các phần tử trong list bắt đầu bằng 'P'

- ✓ **Yêu cầu:** Viết chương trình cho người dùng nhập vào 1 list. Chương trình sẽ kiểm tra và in ra các phần tử trong list bắt đầu bằng 'P':

```
List: ['Python', 'PHP', 'C/C++', 'Java', 'Perl', 'C#']
('Python',)
('PHP',)
('Perl',)
```

- ✓ **Tóm tắt yêu cầu**

- **Nhập:**
 - List
- **Xuất:**
 - Các phần tử trong list bắt đầu bằng 'P'

- ✓ **Hướng dẫn**

- Trong package Bai6, tạo module **kiem_tra_match.py**. Trong đó:
- Sử dụng Regular Expression để giải quyết yêu cầu
- Gợi ý:

```
import re

# nhập list.
words = ["Python", "PHP", "C/C++", "Java", "Perl", 'C#']
print("List:", words)

for w in words:
    m = re.match("(P\w+)", w)
    # kiểm tra
    if m:
        print(m.groups())
```

6.3. Kiểm tra chuỗi chữ thường có _

- ✓ **Yêu cầu:** Viết chương kiểm tra một chuỗi hợp lệ khi chuỗi có các ký tự thường đi kèm với dấu gạch dưới _ .

```
Input string:  abcdef_ass
Found a match!
```



```
Input string: 123abcde_wer
Not matched!
```

✓ **Tóm tắt yêu cầu**

▪ **Nhập:**

- Chuỗi

▪ **Xuất:**

- Chuỗi có theo mẫu chữ thường và gạch dưới _ hay không?

✓ **Hướng dẫn**

- Trong package Bai6, tạo module kiểm_tra_chuoi_thuong_.py. Trong đó:
- Sử dụng Regular Expression để giải quyết yêu cầu
- Gợi ý:

```
import re
def text_match(text):
    patterns = '^[a-z]+_[a-z]+$'
    if re.search(patterns, text):
        return 'Found a match!'
    else:
        return('Not matched!')

string = input("Input string: \t")
print(text_match(string))
```

6.4. Kiểm tra chuỗi bắt đầu bằng 'a' và kết thúc bằng 'b'

- ✓ **Yêu cầu:** Viết chương kiểm tra một chuỗi hợp lệ khi chuỗi bắt đầu bằng 'a' sau đó là gì cũng được và kết thúc bằng 'b'.

```
Input string: a12h#&hgb
Found a match!
```

```
Input string: abcdef23bba
Not matched!
```

✓ **Tóm tắt yêu cầu**

▪ **Nhập:**

- Chuỗi

▪ **Xuất:**

- Chuỗi có đúng mẫu hay không?



✓ **Hướng dẫn**

- Trong package Bai6, tạo module **kiem_tra_chuoi_a_b.py**. Trong đó:
- Sử dụng Regular Expression để giải quyết yêu cầu
- Gợi ý:

```
import re
def text_match(text):
    patterns = 'a.*?b$'
    if re.search(patterns, text):
        return 'Found a match!'
    else:
        return('Not matched!')
string = input("Input string:\t")
print(text_match(string))
```

6.5. Tìm các chuỗi con trong chuỗi

- ✓ **Yêu cầu:** Viết chương trình tìm chuỗi con (là mẫu) trong một chuỗi như sau:

```
Nhập chuỗi:      read book, read book and read book
Nhập mẫu:        book
Tìm thấy mẫu ' book ' 3 lần
```

```
Nhập chuỗi:      rain rain go away
Nhập mẫu:        again
Không tìm thấy mẫu ' again '
```

✓ **Tóm tắt yêu cầu**

- **Nhập:**
 - Chuỗi
 - Mẫu
- **Xuất:**
 - Có chuỗi con trong chuỗi hay không?

✓ **Hướng dẫn**

- Trong package Bai6, tạo module **tim_chuoi_con_trong_chuoi.py**. Trong đó:
- Sử dụng Regular Expression để giải quyết yêu cầu
- Gợi ý:

```
text = input("Nhập chuỗi:\t")
```



```

pattern = input("Nhập mẫu:\t")
times = 0

for match in re.findall(pattern, text):
    times += 1
if times > 0:
    print("Tìm thấy mẫu '", pattern, "'", times, "Lần")
else:
    print("Không tìm thấy mẫu '", pattern, "'")

```

BÀI 7: wxPython



Mục tiêu chính:

- Phát triển ứng dụng GUI đơn giản với Python (Giao diện đồ họa)

7.1. Thông tin công ty

Yêu cầu: Viết chương trình hiển thị màn hình có dạng như sau:

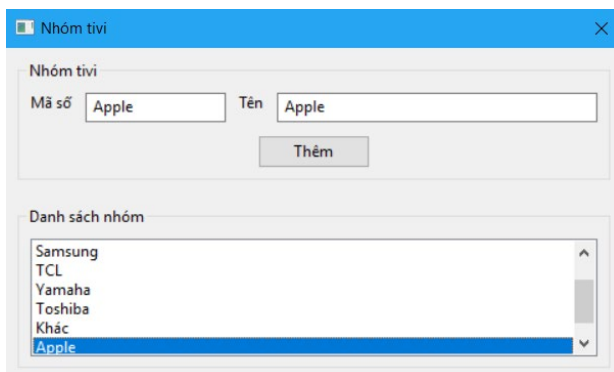


Hướng dẫn:

- Sử dụng phần mềm wxFormBuilder để xây dựng giao diện
- Kết xuất ra file python

7.2. Nhóm Tivi

Yêu cầu: Viết chương trình thêm nhóm tivi và hiển thị danh sách nhóm tivi



- Khởi tạo danh sách nhóm: 'Sony', 'LG', 'Samsung', 'TCL', 'Yamaha', 'Toshiba', 'Khác'
- Khi người dùng nhập nhóm tivi -> click nút Thêm thực hiện thêm Tên vào danh sách nhóm

Hướng dẫn:

- Sử dụng chương trình wxFormBuilder xây dựng giao diện
- Kết xuất thành file python
- Viết code cho sự kiện nút button để lấy giá trị trong TextCtrl và thêm vào list

```
def ThemNhom(self, event):
    TenTivi = self.m_txtTen.GetValue()
    self.m_lstTivi.Append(TenTivi)
```

7.3. Thêm nhân viên

Yêu cầu:

- Viết chương trình thêm nhân viên

- Khi click nút Thêm thì kiểm tra và thông báo lỗi nếu nhập dữ liệu không đầy đủ như sau:

- Nếu nhập dữ liệu đầy đủ thì hiển thị thông báo như sau:

Hướng dẫn:

- Sử dụng chương trình wxFormBuilder xây dựng giao diện
- Lập trình trên sự kiện onClick của nút thêm

```
def ThemNhanVien(self, event):
    Ten=self.m_txtHoTen.GetValue().strip()
    MaSo=self.m_txtMaSo.GetValue().strip()
    TenDangNhap=self.m_txtTenDangNhap.GetValue().strip()
    MatKhai=self.m_txtMatKhai.GetValue().strip()
    MatKhaiXacNhan=self.m_txtMatKhaiXacNhan.GetValue().strip()
    flag=1
    if len(Ten)==0:
        flag=0
        self.m_stHoTenError.SetLabel("!*")
    else:
```



```

        self.m_stHoTenError.SetLabel("")

    if len(MaSo)==0:
        flag=0
        self.m_stMaSoError.SetLabel("*")
    else:
        self.m_stMaSoError.SetLabel("")

    if len(TenDangNhap)==0:
        flag=0
        self.m_stTenDangNhapError.SetLabel("*")
    else:
        self.m_stTenDangNhapError.SetLabel("")

    if len(MatKhau)==0:
        flag=0
        self.m_stMatKhauError.SetLabel("*")
    else:
        self.m_stMatKhauError.SetLabel("")

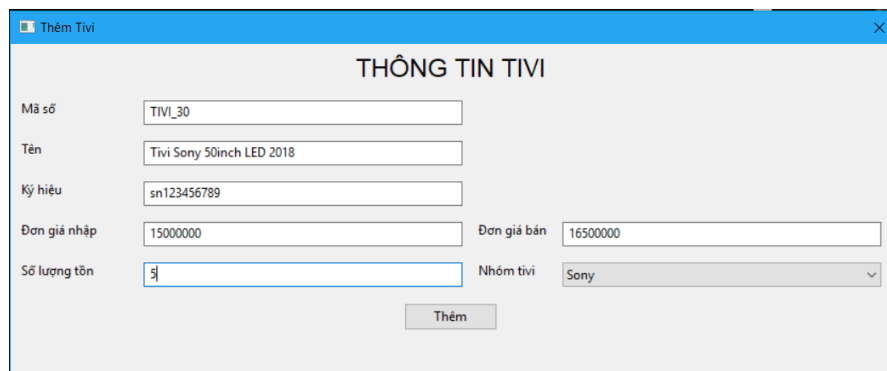
    if len(MatKhauXacNhan)==0:
        flag=0
        self.m_stMKXacNhanError.SetLabel("*")
    else:
        self.m_stMKXacNhanError.SetLabel("")

    if MatKhau!=MatKhauXacNhan:
        flag=0
        self.m_stMatKhauError.SetLabel("*")
    else:
        self.m_stMatKhauError.SetLabel("")

    if flag==1:
        wx.MessageBox("Bạn nhập dữ liệu thành công","Thông báo", wx.OK |
wx.ICON_INFORMATION, self)

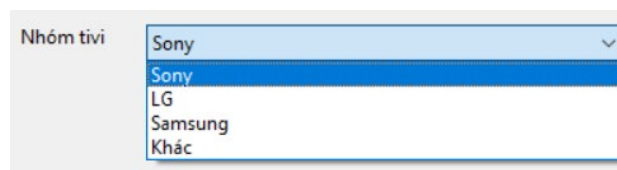
```

7.4. Thêm tivi



Yêu cầu:

- Xây dựng màn hình thêm tivi như trên
- Khởi tạo danh sách nhóm tivi:



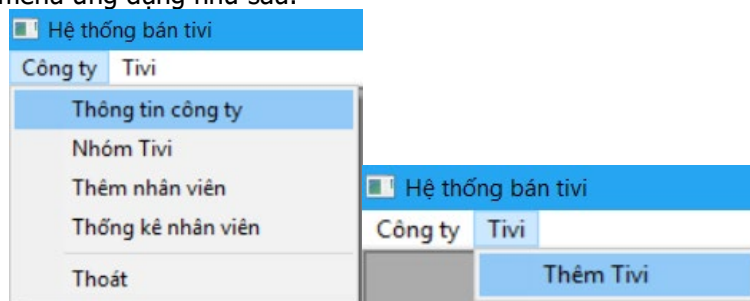
- Chọn mặc định trong nhóm là item đầu tiên

Hướng dẫn:

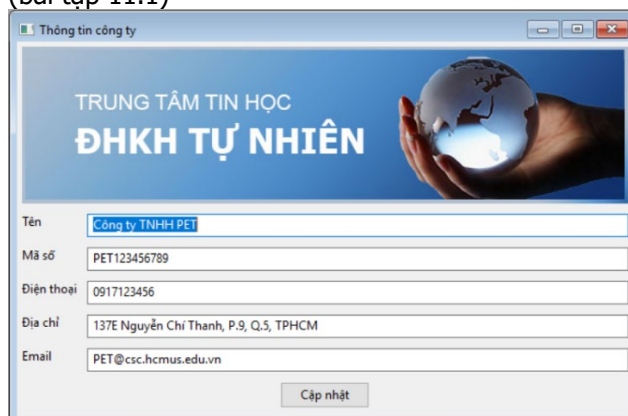
- Thực hiện tương tự như bài 11.3

7.5. Menu ứng dụng

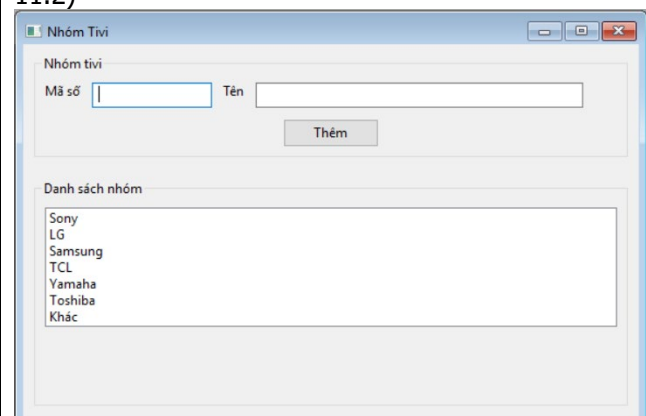
Yêu cầu: Tạo menu ứng dụng như sau:



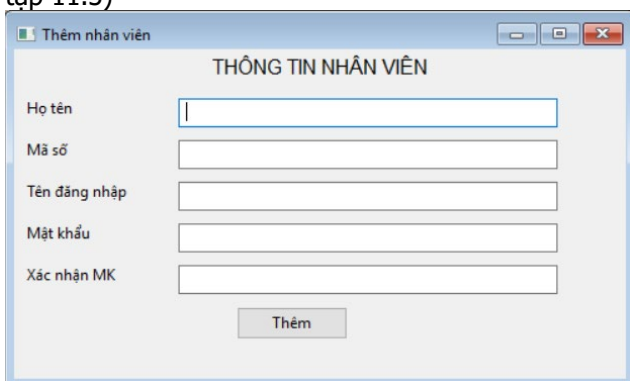
Khi click menu Thông tin công ty thì mở màn hình (bài tập 11.1)



Khi click menu Nhóm tivi thì mở màn hình (bài tập 11.2)

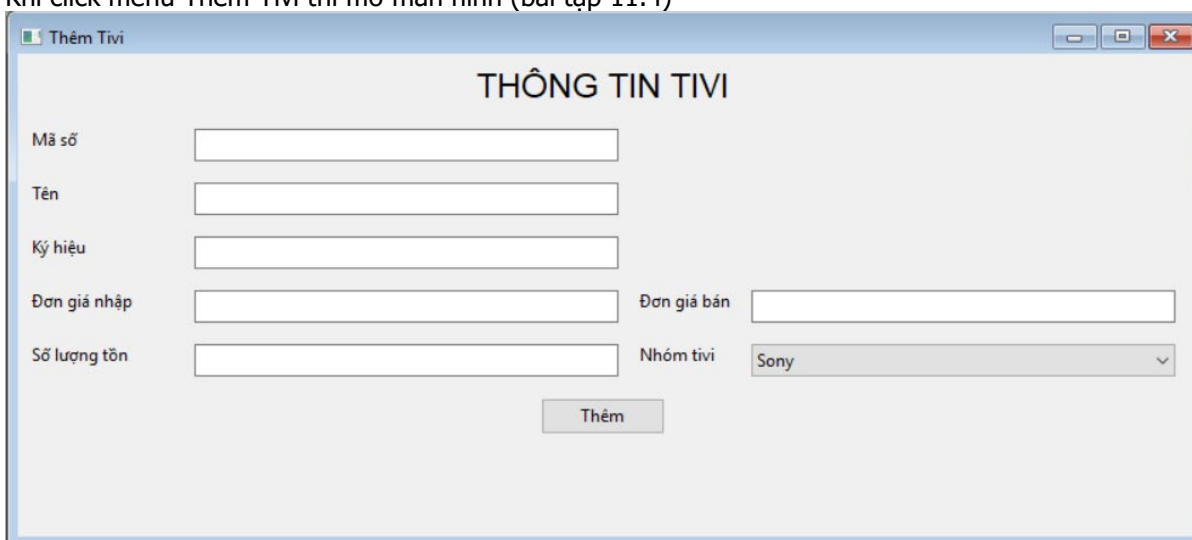


Khi click menu Thêm nhân viên thì mở màn hình (bài tập 11.3)



Khi click menu Thoát thì đóng ứng dụng

Khi click menu Thêm Tivi thì mở màn hình (bài tập 11.4)



BÀI 8: Debugging – Testing



Mục tiêu chính:

- Thực hiện công việc tìm kiếm, bắt lỗi trong code, unit testing để nâng cao chất lượng phần mềm

8.1. Áp dụng Debugging

✓ Yêu cầu: Hãy thực hiện công việc debugging

- Cho các ứng dụng đã được xây dựng Bai1

✓ Hướng dẫn

- Áp dụng debugging để:
 - Tìm kiếm
 - Bắt lỗi
 - Sửa lỗi (nếu có)

8.2. Áp dụng Unit test – kiểm tra tam giác

✓ Yêu cầu: Hãy thực hiện công việc unit test

- Áp dụng unit test cho việc kiểm tra tam giác:
- Xây dựng phương thức kiểm tra tam giác theo thông tin sau:
 - **la_tam_giac(a, b, c)** => trả về True nếu là tam giác, ngược lại trả về False (Là tam giác nếu: $a+b>c$ và $b+c>a$ và $c+a>b$; với a, b, c là 3 cạnh của tam giác)

✓ Hướng dẫn:

- Áp dụng unit test với dữ liệu kiểm tra như sau:

stt	a	b	c	Kết quả mong muốn	Kết quả thực tế	Kết luận kiểm thử
1	1	2	4	FALSE		
2	2	3	4	TRUE		
3	3	1	5	FALSE		
4	4	5	6	TRUE		
5	10	5	20	FALSE		
6	1	1	1	TRUE		
7	2	1	4	FALSE		
8	5	10	8	TRUE		
9	6	3	10	FALSE		
10	2	4	5	TRUE		

- Trong kết quả mong muốn có 2 bộ test: trắng là mong muốn đúng, xám là mong muốn sai
- Chú ý: Kết quả kiểm thử là kết quả kiểm tra trên Unit Test: đúng, hoặc sai

- Phát hiện các vấn đề trong chương trình (nếu có) => sửa lỗi

8.3. Áp dụng Unit test – tính tiền điện

✓ **Yêu cầu: Hãy thực hiện công việc unit test**

- Áp dụng unit test cho bài tính tiền điện:
- Xây dựng phương thức tính tiền điện (có thể tái sử dụng ở môn Python cơ bản) theo thông tin sau:
 - **tinh_tien_dien(sokw)** => trả về số tiền phải trả dựa trên số kw tiêu thụ
 - Số Kwh tiêu thụ và cách thức tính theo quy định như sau:

STT	Mức sử dụng của một hộ trong tháng	Giá bán điện (đồng/kWh)
1	Cho 50kWh (cho hộ nghèo và thu nhập thấp)	1.484
2	Cho kWh từ 51 - 100	1.533
3	Cho kWh từ 101 - 200	1.786
4	Cho kWh từ 201 - 300	2.242
5	Cho kWh từ 301 – 400	2.503
6	Cho kWh từ 401 trở lên	2.587

(Biểu giá áp dụng từ ngày 16/03/2015 theo <https://evnspc.vn/>)

✓ **Hướng dẫn**

- Áp dụng unit test với dữ liệu kiểm tra như sau:

stt	soKW	Thành tiền mong muốn	Thành tiền thực tế	Kết luận kiểm thử
1	45	66,780		
2	145	231,200		
3	245	430,340		
4	345	666,200		
5	445	920,365		
6	50	74,250		
7	100	150,850		
8	200	329,500		
9	300	553,650		
10	400	803,900		

- Trong kết quả mong muốn có 2 bộ test: trắng là mong muốn đúng, xám là mong muốn sai
- Chú ý: Kết quả kiểm thử là kết quả kiểm tra trên Unit Test: đúng, hoặc sai
 - Phát hiện các vấn đề trong chương trình (nếu có) => sửa lỗi

8.4. Áp dụng Unit test – tính lương



✓ **Yêu cầu: Hãy thực hiện công việc unit test**

- Áp dụng unit test cho bài tính lương:
- Xây dựng phương thức tính lương với công thức như sau: tiền lương = hệ số lương * lương cơ bản + trợ cấp + thưởng

✓ **Hướng dẫn**

- Áp dụng unit test với dữ liệu kiểm tra như sau:

stt	heSoLuong	luongCoban	troCap	thuong	Lương mong muốn	Lương thực tế	Kết luận kiểm thử
1	2.34	1,260,000	100,000	250000	3,298,400		
2	3	1,260,000	150,000	250000	4,180,500		
3	3.33	1,260,000	200,000	300000	4,695,800		
4	3.66	1,260,000	300,000	300000	5,211,700		
5	3.99	1,260,000	400,000	350000	5,777,400		
6	2.22	1,260,000	100,000	150000	3,047,300		
7	2.04	1,260,000	150,000	150000	2,870,400		
8	2.67	1,260,000	200,000	250000	3,814,500		
9	4.22	1,260,000	500,000	450000	6,267,200		
10	4.55	1,260,000	300,000	500000	6,533,400		

- Trong kết quả mong muốn có 2 bộ test: trắng là mong muốn đúng, xám là mong muốn sai
- Chú ý: Kết quả kiểm thử là kết quả kiểm tra trên Unit Test: đúng, hoặc sai
 - Phát hiện các vấn đề trong chương trình (nếu có) => sửa lỗi

BÀI 9: Lập trình Game đơn giản với PyGame

(Bài đọc thêm)



Mục tiêu chính:

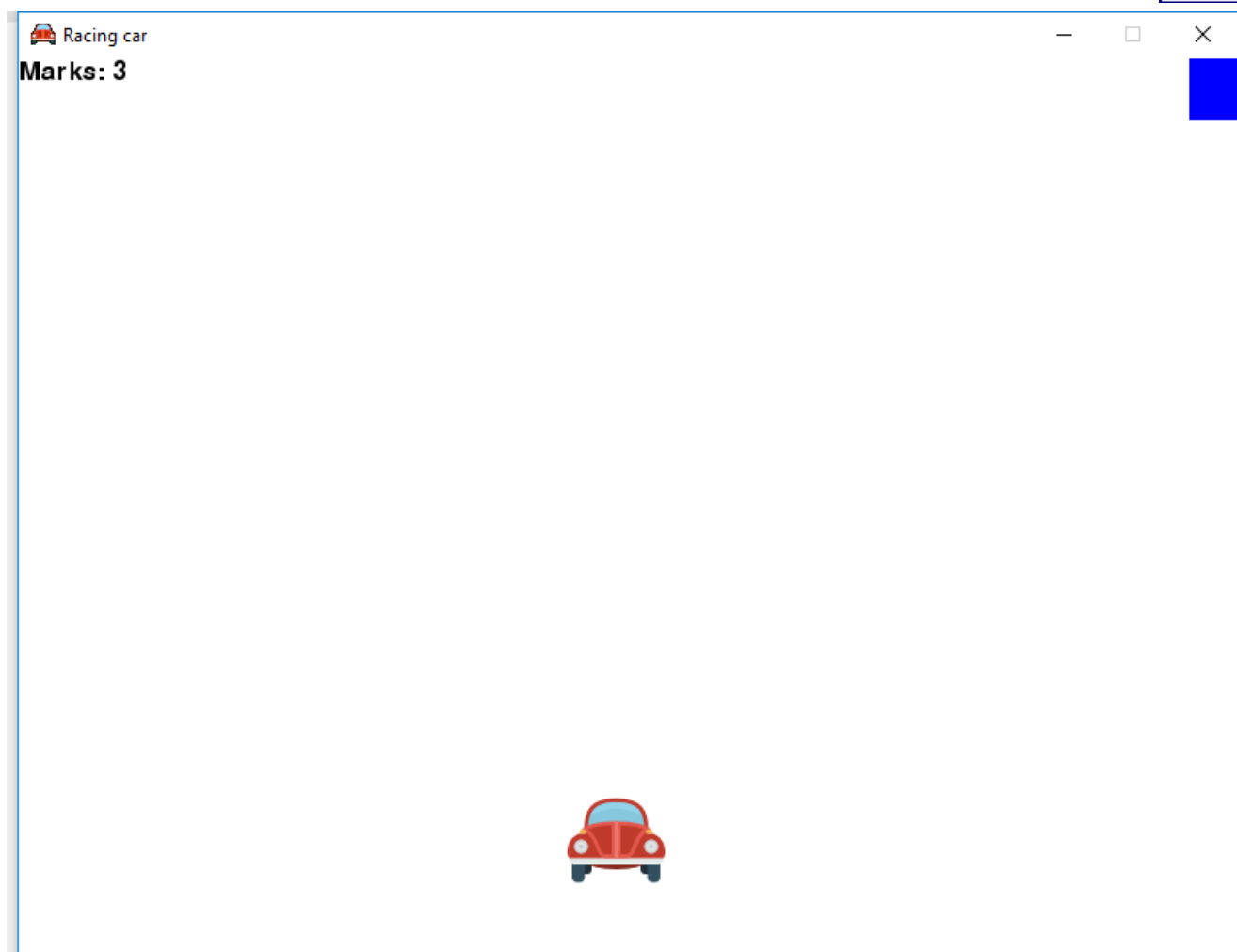
- *Xây dựng ứng dụng Game đơn giản với PyGame*

9.1. Xe tránh vật

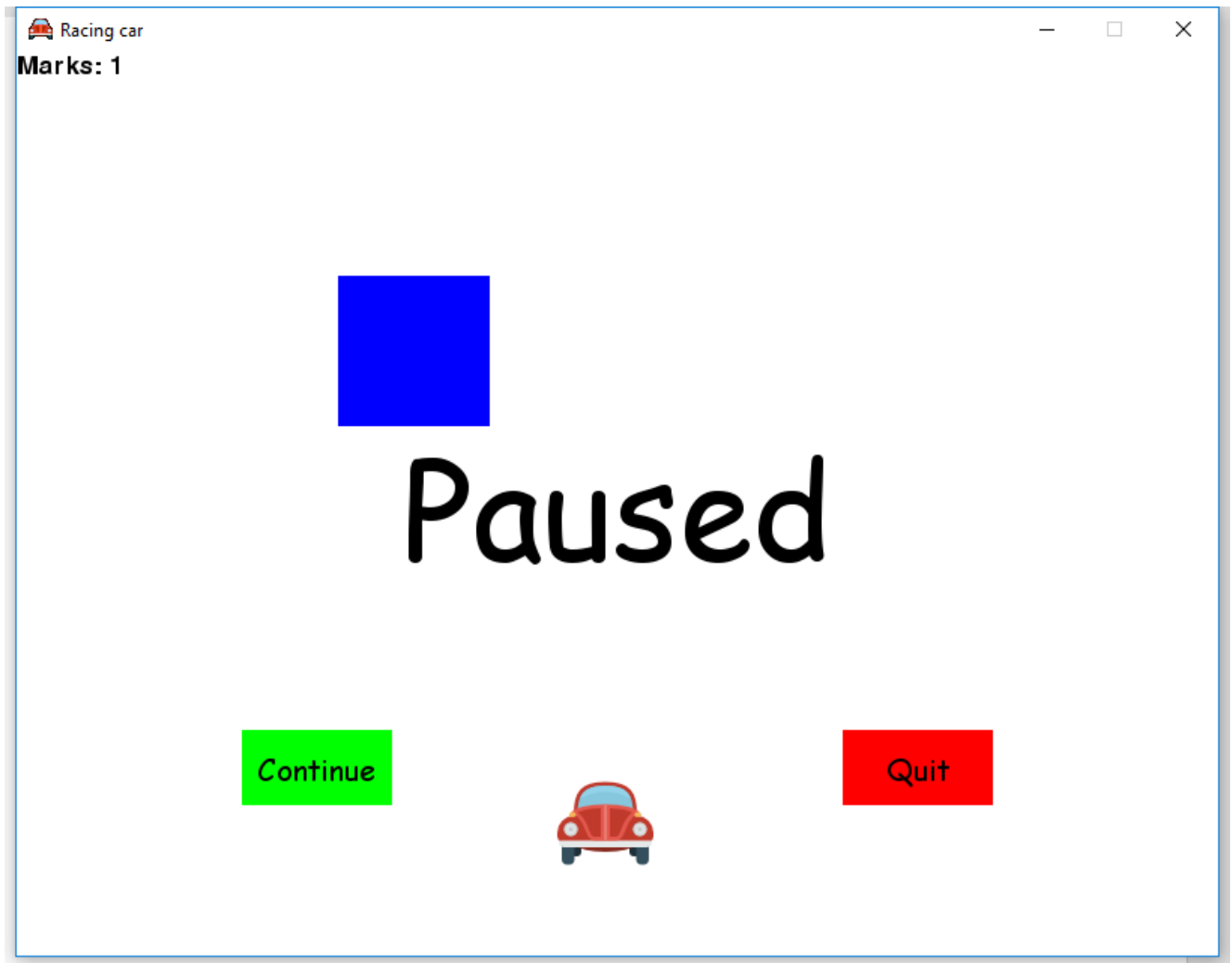
✓ **Yêu cầu:** Xây dựng ứng dụng Xe tránh vật như đã hướng dẫn trong bài học:



Khi thực thi ứng dụng



Khi nhấn nút start



Khi nhấn nút phím 'P' để tạm dừng



Khi trò chơi kết thúc

✓ **Hướng dẫn sử dụng:**

- Người dùng chọn Start để chơi
- Nhấn phím "P" để tạm dừng
- Nhấn Play Again để chơi lại
- Nhấn Quit để thoát
- Trò chơi sẽ kết thúc khi vật rơi trúng xe hoặc xe chạy đụng các biên trái, phải

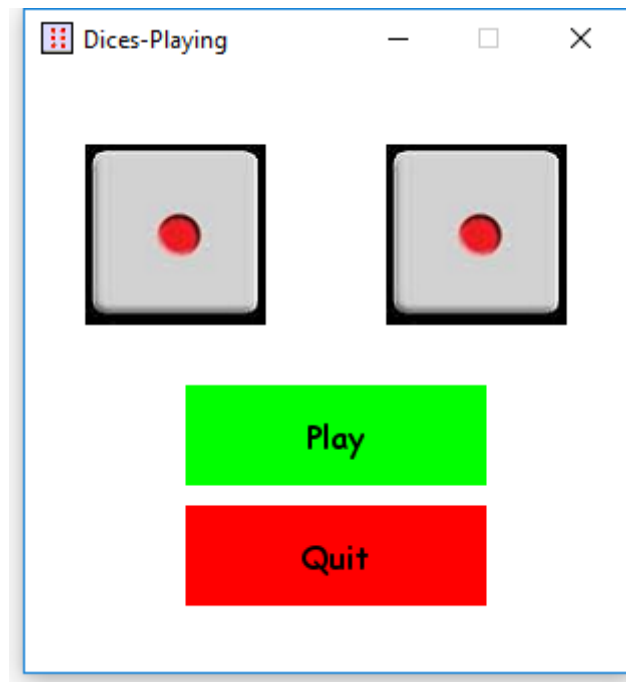
✓ **Hướng dẫn**

- Tạo project **GamePython**
- Trong project GamePython, tạo **package Game_xe**. Trong đó:
- Tạo module **xe_tranh_vat.py**:
- Dựa trên các hướng dẫn trong slide để hoàn thành game Xe tránh vật.

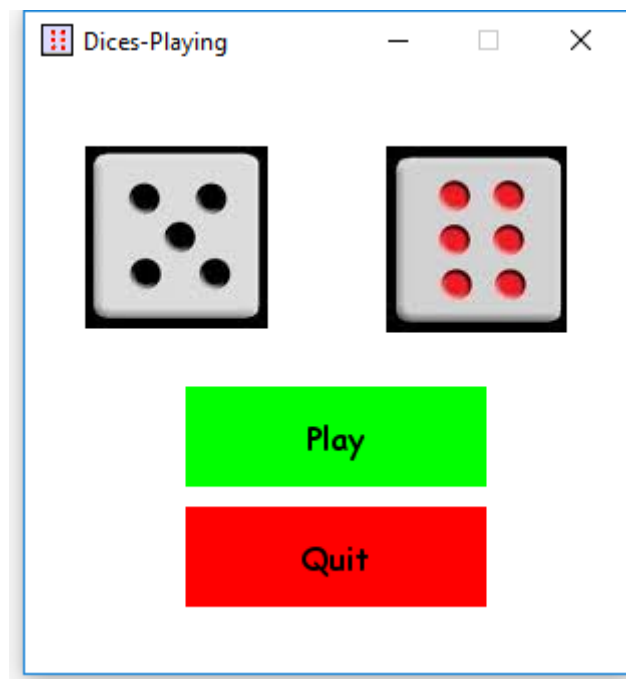
9.2. Đồ xúc xắc

- ✓ **Yêu cầu:** Xây dựng trò chơi "Đồ xúc xắc":

- Tận dụng lại một số hàm trong trò chơi "Xe tránh vật" ở trên để xây dựng trò chơi "Đổ xúc xắc" như sau:



Khi thực thi ứng dụng



Khi nhấn nút Play

Khi nhấn nút Quit => Ngừng chơi và thoát khỏi chương trình

✓ **Hướng dẫn sử dụng:**

- Người dùng sẽ nhấn Play để đổ hai con xúc xắc => xem số điểm mình có dựa trên hai xúc xắc này.
- Nhấn Quit để ngừng chơi và thoát khỏi chương trình



✓ **Hướng dẫn**

- Trong project GamePython, tạo **package Game_xuc_xac**. Trong đó:
- HV được cung cấp sẵn hình ảnh của xúc xắc.
- Tạo module **do_xuc_xac.py**.
- Code gợi ý:

```
import pygame
from random import randint
import sys
import time

sys.setrecursionlimit(123456789)

pygame.init()

# thiết lập các hằng số hiển thị
display_width = 300
display_height = 300
black = (0,0,0)
white = (255,255,255)
red = (255,0,0)
green = (0,255,0)
blue = (0,0,255)
bright_red = (255,0,0)
bright_green = (0,255,0)

# hiển thị màn hình theo kích cỡ, hiển thị icon
gameDisplay = pygame.display.set_mode((display_width,display_height))
pygame.display.set_caption("Dices-Playing")
clock = pygame.time.Clock()

gameIcon = pygame.image.load('sau.png')
pygame.display.set_icon(gameIcon)
```



```
# tạo các hiển thị xúc xắc dựa trên hình được cung cấp và đưa chúng vào
list

xuc_xac_1 = pygame.image.load('mot.png')
xuc_xac_2 = pygame.image.load('hai.png')
xuc_xac_3 = pygame.image.load('ba.png')
xuc_xac_4 = pygame.image.load('bon.png')
xuc_xac_5 = pygame.image.load('nam.png')
xuc_xac_6 = pygame.image.load('sau.png')

list_xuc_xac = []
list_xuc_xac.append(xuc_xac_1)
list_xuc_xac.append(xuc_xac_1)
list_xuc_xac.append(xuc_xac_2)
list_xuc_xac.append(xuc_xac_3)
list_xuc_xac.append(xuc_xac_4)
list_xuc_xac.append(xuc_xac_5)
list_xuc_xac.append(xuc_xac_6)

# viết hàm vẽ một con xúc xắc
def ve_co(x,y,ten_co):
    gameDisplay.blit(ten_co, (x,y))

# thiết lập vị trí hiển thị cho 2 con xúc xắc trên màn hình
x1 = (display_width/2) - 120
y1 = (display_height/8)
x2 = (display_width/2) + 30
y2 = (display_height/8)

# viết hàm hiển thị chữ viết
def text_objects(text, font):
    textSurface = font.render(text, True, black)
    return textSurface, textSurface.get_rect()
```



```
# viết hàm vẽ nút bấm
def button(msg,x,y,w,h,ic,ac,action=None):
    mouse = pygame.mouse.get_pos()
    click = pygame.mouse.get_pressed()

    if x+w > mouse[0] > x and y+h > mouse[1] > y:
        pygame.draw.rect(gameDisplay, ac,(x,y,w,h))

        if click[0] == 1 and action != None:
            action()
        else:
            pygame.draw.rect(gameDisplay, ic,(x,y,w,h))

    smallText = pygame.font.SysFont("comicsansms",16)
    textSurf, textRect = text_objects(msg, smallText)
    textRect.center = ( (x+(w/2)), (y+(h/2)))
    gameDisplay.blit(textSurf, textRect)
    clock.tick(15)

# viết hàm thoát game
def quit_game():
    pygame.quit()
    quit()

# viết hàm khởi động game
def game_intro():
    intro = True
    while intro:
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                #intro = False
                pygame.quit()
```



```

        quit()

    gameDisplay.fill(white)
    ve_co(x1,y1,xuc_xac_1)
    ve_co(x2,y2,xuc_xac_1)

    button("Play",x1+50,y2+120,150,50,green,bright_green, game_loop)
    button("Quit",x1+50,y2+180,150,50,red,bright_red,quit_game)
    pygame.display.update()
    clock.tick(15)

# viết hàm chơi game
def game_loop():
    so1 = randint(1, 6)
    so2 = randint(1, 6)
    gameExit = False
    while not gameExit:
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                #gameExit = True
                pygame.quit()
                quit()

        gameDisplay.fill(white)
        ve_co(x1,y1,list_xuc_xac[so1])
        ve_co(x2,y2,list_xuc_xac[so2])

        button("Play",x1+50,y2+120,150,50,green,bright_green, game_loop)
        button("Quit",x1+50,y2+180,150,50,red,bright_red,quit_game)

        pygame.display.update()
        clock.tick(30)

# gọi các hàm đã viết

```



```
game_intro()
game_loop()
pygame.quit()
quit()
```



Mục lục

BÀI 1: Lập trình hướng đối tượng	2
1.1. Giải phương trình bậc nhất.....	2
1.2. Tính toán hai số	3
1.3. Quản lý CD	5
1.4. Tính khoảng cách giữa 2 điểm	6
1.5. Tính lương (bài làm thêm).....	8
1.6. Quản lý giao dịch (kế thừa)	9
1.7. Quản lý sách thư viện (kế thừa).....	12
1.8. Tính chu vi & diện tích các hình (abstract)	16
1.9. Tính lương (kế thừa) – làm tiếp từ bài tính lương phía trên (bài làm thêm).....	18
BÀI 2: Làm việc với dữ liệu JSON	21
2.1. Đọc và hiển thị các sách từ API trên Internet.....	21
2.2. Đọc và hiển thị sách nổi bật từ API trên Internet.....	24
2.3. Thống kê nhân viên theo đơn vị.....	25
2.4. Ghi dữ liệu quản lý giao dịch vào tập tin tập tin JSON.....	27
2.5. Thêm thông tin phòng karaoke	30
BÀI 3: Làm việc với tập tin XML	33
3.1. Đọc và hiển thị danh bạ điện thoại với SAX API.....	33
3.2. Đọc và hiển thị danh sách nhân viên với DOM XML.....	35
3.3. Đọc - Ghi tập tin XML với DOM	39
BÀI 4: Xây dựng ứng dụng với CSDL.....	43
4.1. Xây dựng ứng dụng Quản lý sản phẩm.....	43
4.2. Xây dựng ứng dụng Quản lý nhân viên.....	46
BÀI 5: Thread - Multithread	50
5.1. Ứng dụng multithread	50
5.2. Tính tổng các phần tử trong list	51
5.3. Tìm giá trị lớn nhất trong list	54
BÀI 6: Regular Expression.....	56
6.1. Kiểm tra chuỗi ký tự	56
6.2. In các phần tử trong list bắt đầu bằng 'P'	57
6.3. Kiểm tra chuỗi chữ thường có _	57
6.4. Kiểm tra chuỗi bắt đầu bằng 'a' và kết thúc bằng 'b'	58



6.5. Tìm các chuỗi con trong chuỗi	59
--	----

BÀI 7: wxPython..... 61

7.1. Thông tin công ty	61
------------------------------	----

7.2. Nhóm Tivi	61
----------------------	----

7.3. Thêm nhân viên	61
---------------------------	----

7.4. Thêm tivi	64
----------------------	----

7.5. Menu ứng dụng	64
--------------------------	----

BÀI 8: Debugging – Testing 66

8.1. Áp dụng Debugging	66
------------------------------	----

8.2. Áp dụng Unit test – kiểm tra tam giác	66
--	----

8.3. Áp dụng Unit test – tính tiền điện	67
---	----

8.4. Áp dụng Unit test – tính lương	67
---	----

BÀI 9: Lập trình Game đơn giản với PyGame (Bài đọc thêm) 69

9.1. Xe tránh vật	69
-------------------------	----

9.2. Đồ xúc xắc	72
-----------------------	----

Mục lục.....	79
---------------------	-----------