

ĐẠI HỌC HUẾ
KHOA KỸ THUẬT VÀ CÔNG NGHỆ



BÁO CÁO ĐỒ ÁN

Học kỳ I, năm học 2022-2023

Học phần:

NGÔN NGỮ LẬP TRÌNH PYTHON

Số phách

(do hội đồng chấm thi ghi)

Thừa Thiên Huế, ngày 15 tháng 1 năm 2023

ĐẠI HỌC HUẾ
KHOA KỸ THUẬT VÀ CÔNG NGHỆ



ĐẠI HỌC HUẾ
KHOA KỸ THUẬT VÀ CÔNG NGHỆ

(Bìa phụ 2)
BÁO CÁO ĐỒ ÁN
Học kỳ I, năm học 2022-2023
Học phần:
NGÔN NGỮ LẬP TRÌNH PYTHON

Lớp: Ngôn ngữ lập trình Python_K3

Giáo viên hướng dẫn: Nguyễn Đình Hoa Cương

Sinh viên thực hiện: Văn Khiêm Chương

22E1020002

Số phách
(do hội đồng chấm thi ghi)

Thừa Thiên Huế, ngày 15 tháng 1 năm 2023

LỜI CẢM ƠN

Bài báo cáo đồ án cuối học kì của môn chuyên ngành ngôn ngữ lập trình python là kết quả nỗ lực không ngừng nghỉ của bản thân và được sự giúp đỡ thầy bộ môn . Qua bài báo cáo này em xin gửi lời cảm ơn tới thầy Nguyễn Đình Hoa Cương vì đã giúp đỡ em trong quá trình hoàn thiện đồ án và giúp em hiểu rõ về chuyên ngành mình đang theo đuổi.

Em xin chân thành cảm ơn đến lãnh đạo và thầy cô bộ môn của khoa kĩ thuật và công nghệ đã tạo điều kiện cho chúng em có một môi trường học tập đầy mới mẻ và hiện tại.

DANH MỤC HÌNH ẢNH

Chương 1:

hình 1	1
hình 2	2
hình 3	3
hình 4	4
hình 5	6
hình 6	8

Chương 2:

hình chương 2. 1	8
hình chương 2. 2	9
hình chương 2. 3	9
hình chương 2. 4	9
hình chương 2. 5	10
hình chương 2. 6	10
hình chương 2. 7	11
hình chương 2. 8	13
hình chương 2. 9	14
hình chương 2. 10	15

DANH MỤC BẢNG BIỂU

MỤC LỤC

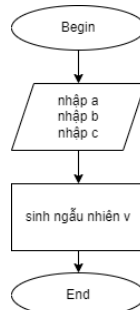
LỜI CẢM ƠN	i
DANH MỤC HÌNH ẢNH	ii
DANH MỤC BẢNG BIỂU	iii
MỤC LỤC	iv
CHƯƠNG 1: LƯU ĐỒ THUẬT TOÁN VÀ LẬP TRÌNH HÀM	1
1.1 Vẽ lưu đồ thuật toán	1
1.2 Lập trình hàm :	5
CHƯƠNG 2: LẬP TRÌNH VỚI CÁC THƯ VIỆN	7
2.1 Thư viện NumPy:	7
2.2 Thư viện SymPy	8
2.3 Thư viện Matplotlib	10
CHƯƠNG 3: LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG	16
3.1 Thiết kế lớp	16
3.2 Sắp xếp và tìm kiếm :	16
3.3 Lưu trữ	17
TÀI LIỆU THAM KHẢO	18
KẾT QUẢ KIỂM TRA ĐẠO VĂN	19

CHƯƠNG 1: LƯU ĐỒ THUẬT TOÁN VÀ LẬP TRÌNH HÀM

1.1 Vẽ lưu đồ thuật toán

Câu 1: Lưu đồ bài Sinh ngẫu nhiên:

Lưu đồ sinh ngẫu nhiên các giá trị cho 1 list các số thực trong khoảng $[a, b]$ được biểu diễn như hình 1:



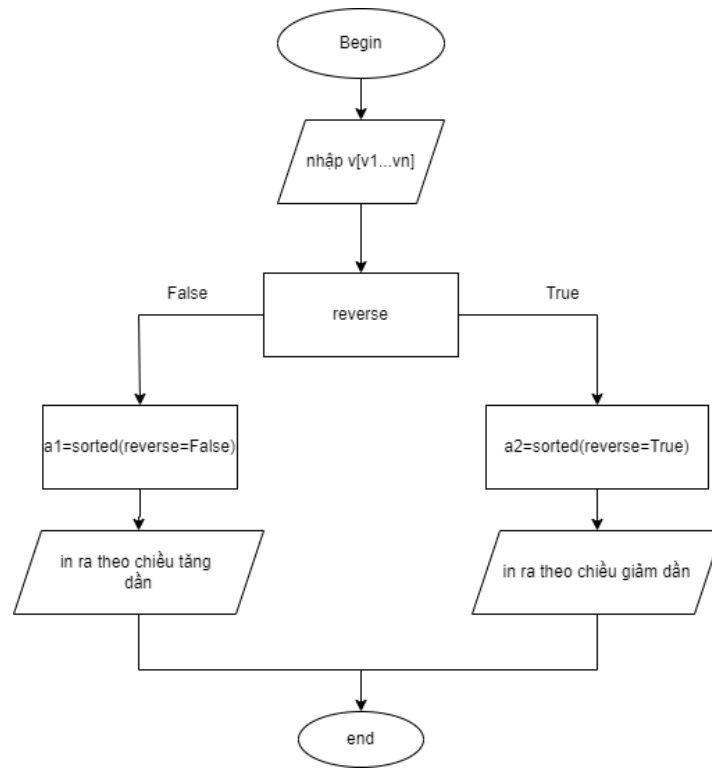
hình 1

- Miêu tả lưu đồ thuật toán :

- Giá trị được tạo ra qua quá trình thực hiện với thư viện numpy, được giới hạn giá trị phần tử trong đoạn $[a, b]$, số lượng phần tử bằng c .
- Câu lệnh `np.linspace(start=a, stop=b, num=c, dtype=float)` để sinh số thực.

Câu 2: Lưu đồ bài Sắp xếp list:

Lưu đồ sắp xếp list các số thực ở trên theo chiều tăng dần (nếu tham số `reverse= True`), giảm dần (nếu tham số `reverse= False`) được biểu diễn như hình 2:

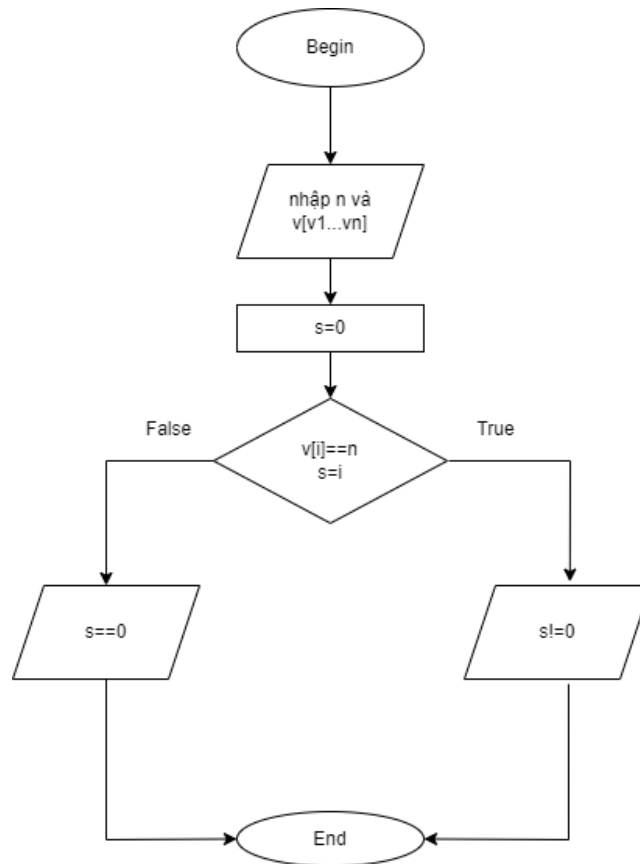


hình 2

- Miêu tả lưu đồ thuật toán :
 - List được lấy từ hình 2, reverse được nhập từ bàn phím .
 - Sử dụng hàm sorted để sắp xếp:
 - Nếu reverse = True thì in ra màn hình list theo chiều giảm dần .
 - Nếu reverse = False thì in ra màn hình list theo chiều tăng dần .

Câu 3: Lưu đồ bài Tìm kiếm số n trong list:

Lưu đồ tìm kiếm số n trong list được biểu diễn như hình 3:

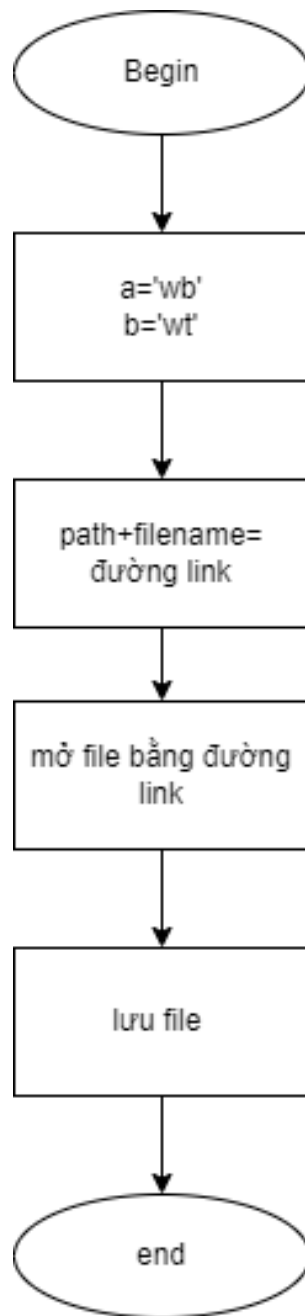


hình 3

- Miêu tả lưu đồ thuật toán:
 - List được lấy từ hình 3, tham số n được nhập từ bàn phím.
 - Cho s=0
 - Sử dụng vòng lặp for với giới hạn của list hình 3, sử dụng if-else để tìm vị trí số thực:
 - Nếu s==0 thì in ra màn hình không tìm thấy vị trí n .
 - Nếu s!=0 thì in ra màn hình vị trí n cần tìm .

Câu 4: Lưu đồ bài Lưu list vào tập tin:

Lưu đồ lưu list vào tập tin có tùy chọn tham số để xác định là lưu tập tin văn bản hay tập tin nhị phân được biểu diễn như hình 4.



hình 4

- Miêu tả lưu đồ :

- List được lấy từ hình 4, tên thư mục , địa chỉ thư mục và tham số nhập từ bàn phím.
- Tham số nhập từ bàn phím:
 - Nhập vào x mở thư mục và viết thư mục ở dạng nhị phân.
 - Nhập vào y mở thư mục và viết thư mục ở dạng văn bản .

1.2 Lập trình hàm :

Từ các lưu đồ thuật toán ở 1.1 ta có đoạn code python như sau:

```
import numpy as np
#1
a=int(input('a='))
b=int(input('b='))
c=int(input('c='))
def list_so_thuc():
    v=np.linspace(start=a,stop=b,num=c,dtype=float)
    return v
#2
```

```
def tang_dan():
    a1 = sorted(list_so_thuc(), reverse=False)
    return a1
print( tang_dan())
def giam_dan():
    a2 = sorted(list_so_thuc(), reverse=True)
    return a2
print(giam_dan())
#3
```

```
def tim_vi_tri(list_so_thuc):
    n = float(input('n='))
    s=0
    for i in range(len(list_so_thuc)):
        if list_so_thuc[i]==n:
            s=i
    if s==0:
        print("không tìm thấy vị trí :",n)
    else:
        print('vị trí',s)
```

```
def tim_vi_tri1(giam_dan):
    n = float(input('n='))
    s=0
    for i in range(len(giam_dan)):
        if giam_dan[i]==n:
            s=i
    if s==0:
        print("không tìm thấy vị trí :",n)
    else:
        print('vị trí',s)
#4
```

```

import os
import pickle
def luu_tap_tin(list_so_thuc):
    x='wb'
    y='wt'
    path="C:\\Users\\HOT_BOY\\PycharmProjects\\pythonProject"
    filename= "chuong.txt"

    with open(os.path.join(path,filename),x) as f:
        pickle.dump(list_so_thuc,f)
    print("kết thúc chương trình lưu tập tin")

# luu_tap_tin(list_so_thuc())
def luu_tap_tin1(giam_dan):
    x = 'wb'
    y = 'wt'
    path = "C:\\Users\\HOT_BOY\\PycharmProjects\\pythonProject"
    filename = "chuong.txt"

    with open(os.path.join(path, filename), x) as f:
        pickle.dump(giam_dan, f)
    print("kết thúc chương trình lưu tập tin")

#5
def main():
    list_so_thuc()
    luu_tap_tin(list_so_thuc)
    giam_dan()
    luu_tap_tin1(giam_dan)
    tim_vi_tri1(giam_dan())

if __name__=="__main__":
    main()

```

Kết quả đoạn code trên :

```

a=1
b=10
c=5
[1.0, 3.25, 5.5, 7.75, 10.0]
[10.0, 7.75, 5.5, 3.25, 1.0]
kết thúc chương trình lưu tập tin
kết thúc chương trình lưu tập tin
n=5.5
vị trí 2

```

```
Process finished with exit code 0
```

hình 5

CHƯƠNG 2: LẬP TRÌNH VỚI CÁC THƯ VIỆN

2.1 Thư viện NumPy:

Câu 1: Chương trình thực hiện các phép toán vector và ma trận :

```
import numpy as np
np.random.seed(123)

#1
def vector_nhan_matran():
    x=np.arange(5,dtype=int)
    _A=np.random.randint(low=-10,high=10,size=20).reshape(4,5)
    A=np.array(_A)
    b=x*A
    print("x:",x)
    print('A:',A)
    print('b=',b)
    return b

#2
def nhan2_ma_tran():
    _A = np.random.randint(low=-10, high=10, size=20).reshape(4, 5)
    A = np.array(_A)
    _B=np.random.randint(low=-10,high=10,size=20).reshape(4,5)
    B=np.array(_B)
    D=np.multiply(A,B)
    print('D=',D)
    return D

def nhan2_ma_tran1():
    _A = np.random.randint(low=-10, high=10, size=20).reshape(4, 5)
    A = np.array(_A)
    A_chuyenvi=A.T
    _B = np.random.randint(low=-10, high=10, size=20).reshape(4, 5)
    B = np.array(_B)
    Z=np.dot(A_chuyenvi,B)
    print('z=',Z)
    return Z

def main():
    vector_nhan_matran()
    nhan2_ma_tran()
    nhan2_ma_tran1()

if __name__=='__main__':
```

Kết quả đoạn code trên được biểu diễn như hình chương 2. 1:

```
x: [0 1 2 3 4]
A: [[ 3 -8 -8 -4 7]
 [ 9 0 -9 -10 7]
 [ 5 -1 -10 4 -10]
 [ 5 9 4 -6 -10]]
b= [[ 0 -8 -16 -12 28]
 [ 0 0 -18 -30 28]
 [ 0 -1 -20 12 -40]
 [ 0 9 8 -18 -40]]
D= [[-30 60 7 56 0]
 [-9 -64 -30 30 -3]
 [-2 28 -12 -81 -48]
 [36 8 6 -7 0]]
z= [[114 40 132 -90 7]
 [-24 30 -55 82 -33]
 [61 20 76 27 -17]
 [-26 -49 38 14 1]
 [-16 127 -62 76 -88]]
```

Process finished with exit code 0

hình chương 2. 1

hình 6

2.2 Thư viện SymPy

Câu 2 Chương trình thực hiện các phép toán :

Câu 2.1 Viết hàm tính giới hạn: $\lim_{x \rightarrow \infty} (\sqrt[3]{x^3 - 3x^2} + \sqrt{x^2 - 2x})$ biểu diễn bằng ngôn ngữ lập trình python

```
from sympy import *
#1
def he_phuong_trinh():
    x,y,z=symbols("x,y,z")
    eq1=Eq(2*x-5*y+z,-5)
    eq2=Eq(4*x+2*y-2*z,2)
    eq3=Eq(x+y-z,0)
    answer=solve((eq1,eq2,eq3),(x,y,z))
    print('a1=',answer)
    return answer
```

Kết quả của đoạn code trên được biểu diễn như hình chương 2. 2:

```
a1= {x: 1, y: 2, z: 3}
```

hình chương 2. 2

câu 2.2 Viết hàm tính giới hạn: $\lim_{x \rightarrow \infty} (\sqrt[3]{x^3 - 3x^2} + \sqrt{x^2 - 2x})$ biểu diễn bằng ngôn ngữ lập trình python

```
from sympy import *
def tinh_gioi_han():
    x=symbols('x')
    f=(x**3-3*x**2)**1/3+sqrt(x**2-2*x)
    answer=limit(f,x,+oo)
    print('a2=',answer)
    return answer
```

Kết quả đoạn code trên được biểu diễn như hình chương 2. 3:

```
a2=  oo
```

hình chương 2. 3

câu 2.3 Viết hàm tính đạo hàm: $f(x) = \frac{2x-1}{x+2}$ biểu diễn bằng ngôn ngữ lập trình python

```
from sympy import *
def dao_ham():
    x=symbols('x')
    f=(2*x-1)/(x+2)
    answer=diff(f,x)
    print('a3=',answer)
    return answer
```

kết quả đoạn code trên được biểu diễn như hình hình chương 2. 4:

```
a3= 2/(x + 2) - (2*x - 1)/(x + 2)**2
```

hình chương 2. 4

câu 2.4 Viết hàm tính nguyên hàm: $\int \frac{x}{x^2 + 1} dx$ biểu diễn bằng ngôn ngữ lập trình python

```
from sympy import *
def nguyen_ham():
    x=symbols('x')
    f=x/(x**2+1)
    answer=integrate(f,x)
    print('a4',answer)
    return answer
```


Kết quả của đoạn code trên được biểu diễn như hình chương 2. 5:

```
a4 log(x**2 + 1)/2
```

hình chương 2. 5

câu 2.5 Viết hàm tính tích phân: $\int_{\frac{2\pi}{3}}^{\pi} \frac{1 - x \cdot \tan(x)}{x^2 \cos(x) + x}$ biểu diễn bằng ngôn ngữ lập trình python

```
from sympy import *
def tich_phan():
    x=symbols("x")
    f=(1-x*tan(x))/(x**2*cos(x)+x)
    answer=integrate(f,(x,pi,(2+pi)/3))
    print('a5=',answer)
    return answer
```

Kết quả đoạn code trên được biểu diễn như hình chương 2. 6:

```
a5= -Integral(x*tan(x)/(x**2*cos(x) + x), (x, pi, 2/3
+ pi/3)) - Integral(-1/(x**2*cos(x) + x), (x, pi, 2/
3 + pi/3))
```

hình chương 2. 6

Câu 2.6 Chương trình chính

```
def main():
    he_phuong_trinh()
    tinh_gioi_han()
    dao_ham()
    nguyen_ham()
    tich_phan()

if __name__=='__main__':
    main()
```

2.3 Thư viện Matplotlib

Câu 3: Chương trình vẽ đồ thị hàm số y :

Xét hàm số $y = x^4 - 2x^2 - 3$, hãy viết chương trình vẽ đồ thị hàm số y, y', y'' và y''' xét $x \in [-10, 10]$ bằng ngôn ngữ lập trình python

Kết quả chương trình trên được biểu diễn như hình chương 2. 7:

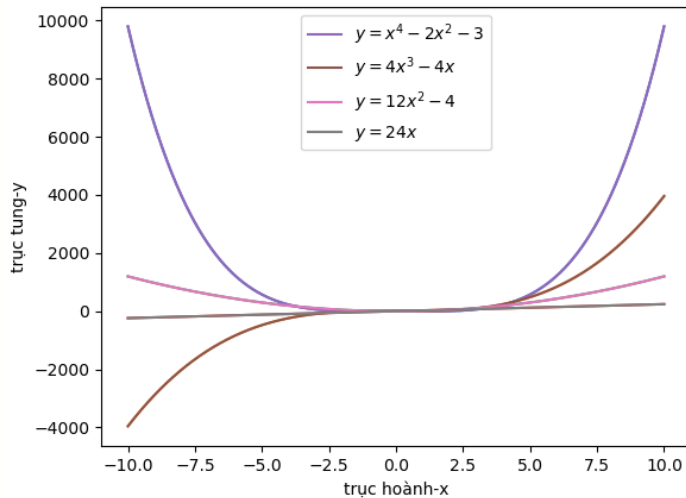
```
import matplotlib.pyplot as plt
import numpy as np

fig,ax = plt.subplots()
x=np.linspace(start=-10,stop=10,num=1000)
def ham_bac_4(a,b,x):
```

```

f=a*x**4+b*x**2-3
return f
def ham_bac_3(a,b,x):
    f = a*x**3+b*x
    return f
def ham_bac_2(a,b,x):
    f = a*x**2+b
    return f
def ham_bac_1(a,x):
    f = a*x
    return f
y1 = ham_bac_4(1,-2,x)
y2=ham_bac_3(4,-4,x)
y3=ham_bac_2(12,-4,x)
y4=ham_bac_1(24,x)
ax.plot(x,y1)
ax.plot(x,y2)
ax.plot(x,y3)
ax.plot(x,y4)
ax.plot(x,y1,label=r'$y=x^4-2x^2-3$')
ax.plot(x,y2,label=r'$y=4x^3-4x$')
ax.plot(x,y3,label=r'$y=12x^2-4$')
ax.plot(x,y4,label=r'$y=24x$')
ax.set_xlabel("trục hoành-x")
ax.set_ylabel("trục tung-y")
ax.legend()
plt.show()

```



hình chương 2. 7

Câu 4: Chương trình thực hiện vẽ các đồ thị :

- Chương trình vẽ đồ thị mặt yên ngựa :

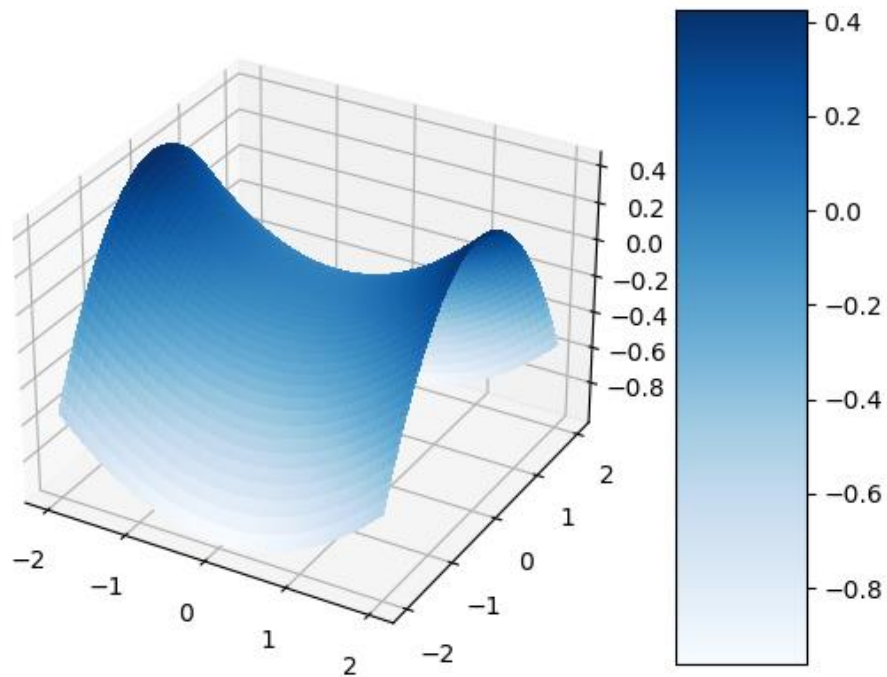
Viết hàm vẽ đồ thị mặt yên ngựa: $\left(\frac{x}{3}\right)^2 - \left(\frac{y}{2}\right)^2 = z$ bằng ngôn ngữ lập trình python

```
import matplotlib.pyplot as plt
import numpy as np
from matplotlib import cm

def cau4_1():
    def ham_mat_yen_ngua(x,y):
        z=(x/3)**2-(y/2)**2
        return z
    x=np.linspace(start=-2.0,stop=2,num=2000)
    y=np.linspace(start=-2,stop=2,num=2000)
    x,y=np.meshgrid(x,y)
    z=ham_mat_yen_ngua(x,y)

    fig, ax =plt.subplots(subplot_kw={"projection":"3d"})
    rosen_surf=ax.plot_surface(x,y,z,cmap= cm.Blues,linewidth=0,antialiased=False)
    fig.colorbar(rosen_surf,shrink=1,aspect=5)
    plt.show()
```

Kết quả chương trình được biểu diễn như hình chương 2. 8:



hình chương 2. 8

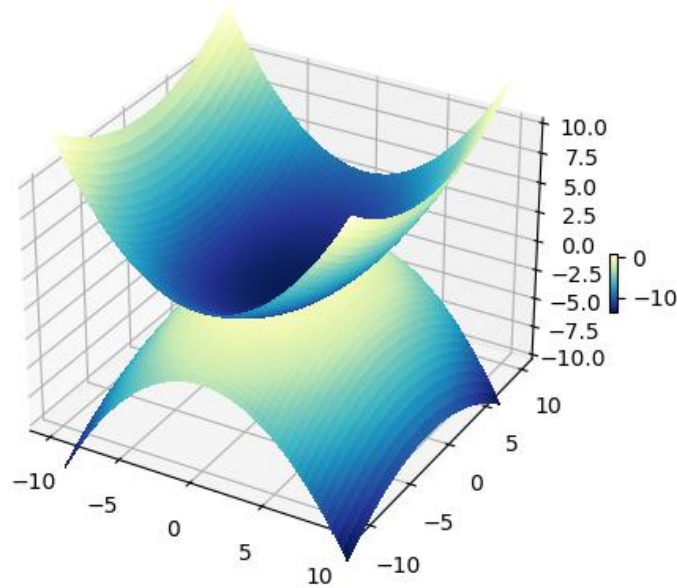
- Chương trình vẽ đồ thị mặt hyperbolic 1 tầng:

Viết hàm vẽ đồ thị mặt hyperbolic 1 tầng: $\left(\frac{x}{3}\right)^2 + \left(\frac{y}{5}\right)^2 - \left(\frac{z}{2}\right)^2 = 1$ bằng ngôn ngữ lập trình python

```
import matplotlib.pyplot as plt
import numpy as np
from matplotlib import cm
def cau4_2():
    def ham_mat_hyperbolic1(x,y):
        q=(x/3)**2+(y/5)**2-1
        return q
    def ham_mat_hyperbolic2(x,y):
        q1=-((x/3)**2+(y/5)**2-1)
        return q1
    fig,ax=plt.subplots()
    x=np.linspace(start=-10.0,stop=10.0,num=1000)
    y=np.linspace(start=-10.0,stop=10.0,num=1000)
    x,y=np.meshgrid(x,y)
    z=ham_mat_hyperbolic1(x,y)
    z1=ham_mat_hyperbolic2(x,y)
```

```
fig,ax=plt.subplots(subplot_kw={"projection":"3d"})
rosen_surf=ax.plot_surface(x,y,z,cmap=plt.cm.YlGnBu_r,linewidth=0,antialiased=False)
rosen_surf=ax.plot_surface(x,y,z1,cmap=plt.cm.YlGnBu_r,linewidth=0,antialiased=False)
ax.set_zlim(-10,10)
fig.colorbar(rosen_surf,shrink=0.1,aspect=7)
plt.show()
```

Kết quả chương trình được biểu diễn như hình chương 2. 9:



hình chương 2. 9

- Chương trình vẽ đồ thị mặt cầu:
Viết hàm vẽ đồ thị mặt cầu: $(x + 2)^2 + (y - 1)^2 + (z - 4)^2 = 4$ bằng ngôn ngữ lập trình python

```
import matplotlib.pyplot as plt
import numpy as np
from matplotlib import cm

def ham_mat_cau2():
    def ham_mat_cau(x,y):
        z=np.sqrt(4-(x+2)**2-(y-1)**2)+4

        return z
    def ham_mat_cau1(x,y):
        z1=-np.sqrt(4-(x+2)**2-(y-1)**2)+4
        return z1
```

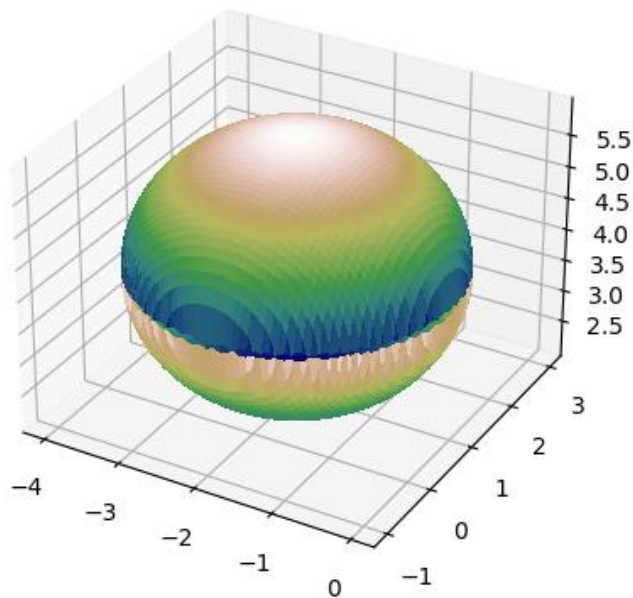
```

x = np.linspace(-4, 0, 2000)
y = np.linspace(-1, 3, 2000)
x, y = np.meshgrid(x, y)
z=ham_mat_cau(x,y)
z1=ham_mat_cau1(x,y)
fig,ax=plt.subplots(subplot_kw={"projection":"3d"})
ax.plot_surface(x,y,z1,cmap=cm.gist_earth,linewidth=0,antialiased=False)
ax.plot_surface(x,y,z,cmap=cm.gist_earth,linewidth=0,antialiased=False)

plt.show()

```

Kết quả chương trình trên được biểu diễn như hình chương 2. 10:



hình chương 2. 10

- Chương trình chính :

```

def main():
    cau4_1()
    cau4_2()
    ham_mat_cau2()

if __name__=='__main__':
    main()

```

CHƯƠNG 3: LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

3.1 Thiết kế lớp

- Chương trình xây dựng lớp :

```
import pickle
import os
class NhanVien:
    def __init__(self, hoten:str , tuoi: int, luong: str):
        self.hoten=hoten
        self.tuoi=tuoi
        self.luong=luong

    def __str__(self)->str:
        nhanvien='[hoten: '+self.hoten +' ;tuổi: '\
            + str(self.tuoi)+' ;luong'\
            +str(self.luong)+' ]'
        return nhanvien
```

- Chương trình nhập dữ liệu cho các đối tượng thuộc lớp:

```
def nhan_vien():
    nv1=NhanVien('Văn Khiêm Chương',18,'20 triệu')
    nv2=NhanVien('Bảo Tín',17,'18 triệu')
    nv3=NhanVien('Trịnh Quốc Dân',18,'17 triệu')
    nv=[nv1,nv2,nv3]
    return nv
```

- Chương trình hiển thị các đối tượng của lớp:

```
def hien_thi_list():
    nv=nhan_vien()
    for i in nv:
        print(i)
```

3.2 Sắp xếp và tìm kiếm :

- Chương trình sắp xếp các đối tượng trong lớp:

```
def sap_xep():
    def __gt__(self,other):
        return (self.tuoi>other.tuoi)
    def __ge__(self,other):
        return(self.tuoi>=other.tuoi)
    def __lt__(self,other):
        return(self.tuoi<other.tuoi)
    def __le__(self,other):
        return(self.tuoi<=other.tuoi)
    def __eq__(self,other):
        return(self.tuoi==other.tuoi)
```

3.3 Lưu trữ

- Chương trình lưu các đối tượng thuộc lớp vào tập tin nhị phân:

```
def luu_list():  
    path='C:\\Users\\HOT_BOY\\PycharmProjects\\pythonProject'  
    filename='Nhanvien.txt'  
    with open(os.path.join(path,filename),'wb') as f:  
        pickle.dump(nhan_vien,f)  
    print('Kết thúc quá trình lưu')
```

- Chương trình đọc đối tượng thuộc lớp từ tập tin nhị phân:

```
def doc_list():  
    path = 'C:\\Users\\HOT_BOY\\PycharmProjects\\pythonProject'  
    filename = 'Nhanvien.txt'  
    try:  
        f = open(os.path.join(path, filename), 'rt')  
        content = f.readlines()  
        f.close()  
        print(content)  
    except Exception as e:  
        print('Error: ', e)  
    print('Kết thúc quá trình đọc')
```

- Chương trình chính :

```
def main():  
  
    nhan_vien()  
    hien_thi_list()  
    sap_xep()  
    luu_list()  
    doc_list()  
  
if __name__=='__main__':  
    main()
```


TÀI LIỆU THAM KHẢO

KẾT QUẢ KIỂM TRA ĐẠO VĂN

