

现代程序设计第9周作业

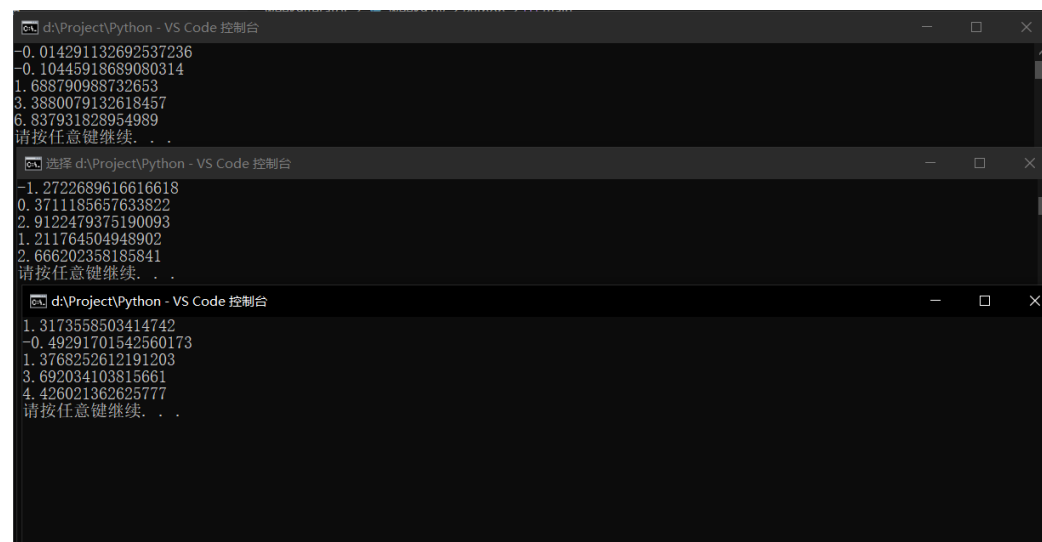
谢奕飞 20377077

一、随机游走

```
1 '''
2 一、生成器函数
3 '''
4 import numpy as np
5 def random_walk(mu,x_0,sigma,N):
6     '''
7     随机游走生成器函数
8     '''
9     x=x_0
10    for i in range(N):
11        yield x+np.random.normal(0,sigma)
12        w=np.random.normal(0,sigma)
13        x=mu+x+w
14    return 'NaN'
```

random_walk生成器

```
1 def main():
2     r=random_walk(1,0,1,5)
3     print(next(r))
4     print(next(r))
5     print(next(r))
6     print(next(r))
7     print(next(r))
```



```
d:\Project\Python - VS Code 控制台
-0.014291132692537236
-0.10445918689080314
1.688790988732653
3.3880079132618457
6.837931828954989
请按任意键继续. . .

d:\Project\Python - VS Code 控制台
-1.2722689616616618
0.3711185657633822
2.9122479375190093
1.211764504948902
2.666202358185841
请按任意键继续. . .

d:\Project\Python - VS Code 控制台
1.3173558503414742
-0.49291701542560173
1.3768252612191203
3.692034103815661
4.426021362625777
请按任意键继续. . .
```

```

1 def main():
2     r=random_walk(10,0,10,5)
3     print(next(r))
4     print(next(r))
5     print(next(r))
6     print(next(r))
7     print(next(r))

```

Three screenshots of a VS Code terminal window showing the output of the random_walk function. The first screenshot shows the first five random numbers: -4.457404711554038, 28.67053021946098, 34.515143845013895, 43.41526464174441, 39.11618028609001. The second screenshot shows the next five random numbers: 0.3836391821721038, 2.883067524999001, -1.1516851162017865, 55.58773986690411, 49.28946888398609. The third screenshot shows the final five random numbers: -8.675577659880487, 26.14724271710187, 27.96973273637845, 62.29425436360072, 59.15422541547187.

zip

```

1 def main():
2     r1=random_walk(1,0,1,10)
3     r2=random_walk(1,0,0.5,10)
4     r3=random_walk(1,0,4,10)
5     z=zip(r1,r2,r3)
6     for each in z:
7         print(each)

```

Three screenshots of a VS Code terminal window showing the output of the zip function. The first screenshot shows the first five tuples: (0.4994293305013716, 0.11973842169538053, 4.5616563913184764), (2.211052040707383, 1.2311278671382213, -4.723544582981526), (1.9941068435647473, 3.062955237449128, -3.319627561114013), (2.894239631658542, 4.612875652053452, -6.878785261752727), (-1.2949276952566522, 5.567313417657303, -12.52237894784788). The second screenshot shows the next five tuples: (2.06172655449726, 5.69676162023532, -7.939835489395674), (3.3173692055819877, 6.625492782651141, -11.359960252123676), (5.661859751485441, 7.775050037501796, -9.45734591411212), (5.573863767246233, 7.684243956883414, -11.94099258612368), (8.87116558657557, 9.746889574651169, 1.1036076697370714). The third screenshot shows the final five tuples: (0.005367932510898599, 0.9415744166599207, 6.273952352344537), (0.6728925902577946, 0.9478781883427128, -1.0632927037641244), (3.5304017577137476, 2.017910417996646, -2.5278362458620274), (3.601484638620075, 2.3613108698630967, 10.657170693026572), (4.010841245736302, 3.3850597307650925, 16.66192796554367).

二、迭代器

```

1  '''
2  二、迭代器
3  '''
4  from PIL import Image
5  import numpy as np
6  import os
7
8  class FaceDataset:
9      def __init__(self,path):
10         self.path=path
11         self.n=len(os.listdir(path))
12         self.i=0
13         self.image_list=os.listdir(path)
14     def __iter__(self):
15         return self
16     def __next__(self):
17         if self.i<self.n:
18             img=Image.open(self.path+'/'+self.image_list[self.i])
19             img=np.array(img)
20             self.i+=1
21             return img
22         else:
23             raise StopIteration()

```

FaceDataset迭代类

```

1  def main():
2      path='D:/Project/Python/week9Iterator/originalPics/2003/06/07/big'
3      image_list=FaceDataset(path)
4      for image in image_list:
5          print(image)

```

```

d:\Project\Python - VS Code 控制台
[160 131 75]
[152 123 67]
[146 117 61]
...
[[189 128 74]
[187 126 72]
[185 124 70]
...
[172 119 79]
[169 117 77]
[160 111 70]]
[[191 131 79]
[189 129 77]
[194 134 82]
...
[174 118 81]
[171 117 79]
[164 112 73]]
[[189 131 81]
[187 129 79]
[192 134 84]
...
[174 118 83]
[172 116 81]
[165 111 75]]
请按任意键继续...

```

附录-全部代码

```

1  '''
2  一、生成器函数
3  '''

```

```

4 import numpy as np
5 def random_walk(mu,x_0,sigma,N):
6     '''
7     随机游走生成器函数
8     '''
9     x=x_0
10    for i in range(N):
11        yield x+np.random.normal(0,sigma)
12        w=np.random.normal(0,sigma)
13        x=mu+x+w
14    return 'NaN'
15    '''
16 def main():
17     r1=random_walk(1,0,1,10)
18     r2=random_walk(1,0,0.5,10)
19     r3=random_walk(1,0,4,10)
20     z=zip(r1,r2,r3)
21     for each in z:
22         print(each)
23     '''
24     '''
25 二、迭代器
26     '''
27 from PIL import Image
28 import numpy as np
29 import os
30
31 class FaceDataset:
32     def __init__(self,path):
33         self.path=path
34         self.n=len(os.listdir(path))
35         self.i=0
36         self.image_list=os.listdir(path)
37     def __iter__(self):
38         return self
39     def __next__(self):
40         if self.i<self.n:
41             img=Image.open(self.path+'/'+self.image_list[self.i])
42             img=np.array(img)
43             self.i+=1
44             return img
45         else:
46             raise StopIteration()
47 def main():
48     path='D:/Project/Python/week9Iterator/originalPics/2003/06/07/big'
49     image_list=FaceDataset(path)
50     for image in image_list:
51         print(image)
52 if __name__=='__main__':main()

```

