```
 1: import os
 2: import matplotlib.pyplot as plt
 3: from datetime import datetime
 4:
 5:
 6: class Output():
 7:     def __init__(self):
 8:
 9:         self.date = datetime.now().strftime('%Y-%m-%d %H:%M:%S')
10:
11:     def Describe(self, title, **kwargs):
12:
13:         name = f'EXP_{title}'
14:
15:         file = open(f'{title}.txt', 'w')
16:
17:         file.write(f'{title}, {self.date} \n \n')
18:
19:
20:         for e in kwargs:
21:
22:             if e == 'fixed':
23:                 fixed = kwargs[e]
24:                 file.write(f'\nFIXED PARAMETERS   : \n                Min       Max
    ds       Mode \n')
25:                 for e in fixed.attribute:
26:                     gen = f'gen_{e}'
27:                     cha = f'cha_{e}'
28:                     if hasattr(fixed, gen):
29:                         file.write(getattr(fixed, gen))
30:                     if hasattr(fixed, cha):
31:                         file.write(getattr(fixed, cha))
32:
33:             if e == 'stochastic':
34:                 stochastic = kwargs[e]
35:                 file.write(f'\nSTOCHASTIC PARAMETERS   : \n              Min
 Max       Std       distribution[NbStep, NbDraw] \n')
36:                 for e in stochastic.attribute:
37:                     gen = f'gen_{e}'
38:                     cha = f'cha_{e}'
39:                     comp = f'comp_{e}'
40:                     if hasattr(stochastic, gen):
41:                         file.write(getattr(stochastic, gen))
42:                     if hasattr(stochastic, cha):
43:                         file.write(getattr(stochastic, cha))
44:                     if hasattr(stochastic, comp):
45:                         file.write(getattr(stochastic, comp))
46:
47:             if e == 'creep':
48:                 creep = kwargs[e]
49:                 file.write(f'\n \nCREEP LAW   : \n')
50:                 config = getattr(creep, 'configurations')
51:                 for element in config:
52:                     file.write(f'{element} \n \n ')
53:
54:     def SaveAttributes(self, obj, attribute = 'all'):
55:
56:         if attribute == 'all':
57:             attribute = getattr(obj, 'attribute')
58:
59:         for element in attribute:
60:             df = getattr(obj, element)
61:             df.to_csv(f'{element}.txt', sep = ';')
62:
63:
64:
65:
66:
67:
68:
```