

```
1 from random import random
2
3
4 class Particle():
5
6     def __init__(self, particle_size):
7         self.position = [random() for i in range(particle_size)]
8         self.velocity = [0 for i in range(particle_size)]
9         self.calculate_fitness()
10
11     def calculate_fitness(self):
12         self.fitness = 0
13
14
15 class ParticleSwarmOptimization():
16
17     def __init__(self, pop_size, particle_size, k=None):
18         self.k = k
19         self.pop_size, self.particle_size = pop_size, particle_size
20
21     def initPops(self):
22         self.pops = [Particle(self.particle_size) for n in range(self.pop_size)]
23         self.p_best = self.pops
24         self.g_best = self.get_g_best()
25
26     def get_g_best(self):
27
28         p_best_sorted = self.p_best
29         p_best_sorted.sort(key=lambda x: x.fitness, reverse=True)
30         return p_best_sorted[0]
31
32     def velocity_clamping(self, vnew):
33         if self.k is None:
34             return vnew
```