Imagine a game where one or more rats can attack a player. Each individual rat has an initial attack value of 1. However, rats attack as a swarm, so each rat's attack value is actually equal to the total number of rats in play.

Given that a rat enters play through the initializer and leaves play (dies) via its \_\_exit\_\_ method, please implement the Game and Rat classes so that, at any point in the game, the Attack value of a rat is always consistent.

Here's a sample unit test your code should pass:

def test\_three\_rats\_one\_dies(self):

game = Game()

rat = Rat(game)

self.assertEqual(1, rat.attack)

rat2 = Rat(game)

self.assertEqual(2, rat.attack)

self.assertEqual(2, rat2.attack)

with Rat(game) as rat3:

self.assertEqual(3, rat.attack)

self.assertEqual(3, rat2.attack)

self.assertEqual(3, rat3.attack)

self.assertEqual(2, rat.attack)

self.assertEqual(2, rat2.attack)