

Ahmet ESMER

ahmetesmer14@hotmail.com

https://github.com/AHMET-ESMER14/Python_Case

Verilen case görevimi Python dilini kullanarak gerçekleştirdim.

```
1
2 import random
3
4 Hayvanlar = []
5 Avlanma_sayisi = 0
6 Ureme_sayisi = 0
7
8 class Hayvan:
9     def __init__(self, Tur, Cinsiyet, x, y, Hareket_kabilyeti):
10         self.Tur = Tur
11         self.Cinsiyet = Cinsiyet
12         self.x = x
13         self.y = y
14         self.Hareket_kabilyeti = Hareket_kabilyeti
15
```

Çiftliğimiz için hayvan sınıfı oluşturuyoruz.

```
18 def Yeni_Hayvan_Ekle(cls, Tur, Cinsiyet, Adet, Hareket_kabilyeti):
19     Yeni_Hayvan = []
20     Hareket_kabilyeti = Hareket_kabilyeti
21     for i in range(Adet):
22         x = random.randint(0, 500)
23         y = random.randint(0, 500)
24         Yeni_Hayvan.append(cls(Tur, Cinsiyet, x, y, Hareket_kabilyeti))
25     return Yeni_Hayvan
26
```

Oluşturduğumuz hayvan sınıfı içerisinde Yeni_Hayvan_Ekle methodu ile yeni hayvanları çiftliğimize ekliyoruz.

```
27 def Hareket(self):
28     dx = random.randint(-self.Hareket_kabilyeti, self.Hareket_kabilyeti)
29     dy = random.randint(-self.Hareket_kabilyeti, self.Hareket_kabilyeti)
30     self.x += dx
31     self.y += dy
32     # Sınırları kontrol etme
33     self.x = max(0, min(self.x, 500))
34     self.y = max(0, min(self.y, 500))
35
```

Oluşturduğumuz hayvan sınıfı içerisinde Hareket methodu ile çiftliğimizdeki hayvanları hareket ettirme işlemlerini gerçekleştiriyoruz.

```

36     def Üreme(self, Hayvanlar):
37         for Hayvan in Hayvanlar:
38             if self == Hayvan: # Kendisiyle eşleşmeye çalışmamak için
39                 return False
40
41             if (Hayvan.Tur == self.Tur) and (Hayvan.Cinsiyet != self.Cinsiyet):
42                 Mesafe = ((self.x - Hayvan.x) ** 2 + (self.y - Hayvan.y) ** 2) ** 0.5
43                 if Mesafe >= 0 and Mesafe <= 3:
44                     return True
45                 else:
46                     return False
47
48             elif (self.Tur == "Tavuk" and Hayvan.Tur == "Horoz") or (self.Tur == "Horoz" and Hayvan.Tur == "Tavuk"):
49                 Mesafe = ((self.x - Hayvan.x) ** 2 + (self.y - Hayvan.y) ** 2) ** 0.5
50                 if Mesafe >= 0 and Mesafe <= 3:
51                     return True
52                 else:
53                     return False
54

```

Oluşturduğumuz hayvan sınıfı içerisinde Üreme methodu ile çiftliğimizdeki hayvanlar üreme için koşulları sağlıyor ise üreme işlemlerini gerçekleştirerek çiftlik nüfusumuzu arttırıyoruz.

```

55     def Avlanma(self):
56         Avlanma_Kosul = {
57             'Kurt': ['Koyun', 'Tavuk', 'Horoz'],
58             'Aslan': ['İnek', 'Koyun'],
59             'Avcı': ['Koyun', 'Tavuk', 'Horoz', 'İnek']}
60
61         Avlanma_Kosul_ = Avlanma_Kosul.get(self.Tur, [])
62         for Hayvan in Hayvanlar:
63             if Hayvan.Tur in Avlanma_Kosul_:
64                 Mesafe = ((self.x - Hayvan.x) ** 2 + (self.y - Hayvan.y) ** 2) ** 0.5
65                 if (self.Tur == 'Kurt' and (Mesafe >= 0 and Mesafe <= 4)) or \
66                     (self.Tur == 'Aslan' and (Mesafe >= 0 and Mesafe <= 5)) or \
67                     (self.Tur == 'Avcı' and (Mesafe >= 0 and Mesafe <= 8)):
68                     Hayvanlar.remove(Hayvan) # Avlanılan hayvanı silme
69                     global Avlanma_sayisi
70                     Avlanma_sayisi += 1
71                     print(f"{self.Tur} {self.Cinsiyet} avladı: {Hayvan.Tur} {Hayvan.Cinsiyet}")
72

```

Oluşturduğumuz hayvan sınıfı içerisinde Avlanma methodu ile çiftliğimizdeki avlanma özelliği olan hayvanlarımız avlanma koşullarını sağlamakta ise avlanılan hayvanımızı çiftliğimizden siliyoruz. Bilgilendirme için Avlanan ve avlanılan hayvanı ekrana yazdırıyoruz.

```

74     Hayvanlar.extend(Hayvan.Yeni_Hayvan_Ekle("Koyun", "Erkek", 10, 2))
75     Hayvanlar.extend(Hayvan.Yeni_Hayvan_Ekle("Koyun", "Dişi", 15, 2))
76
77     Hayvanlar.extend(Hayvan.Yeni_Hayvan_Ekle("İnek", "Erkek", 5, 2))
78     Hayvanlar.extend(Hayvan.Yeni_Hayvan_Ekle("İnek", "Dişi", 5, 2))
79
80     Hayvanlar.extend(Hayvan.Yeni_Hayvan_Ekle("Tavuk", "Dişi", 10, 1))
81     Hayvanlar.extend(Hayvan.Yeni_Hayvan_Ekle("Horoz", "Erkek", 10, 1))
82
83     Hayvanlar.extend(Hayvan.Yeni_Hayvan_Ekle("Kurt", "Erkek", 5, 3))
84     Hayvanlar.extend(Hayvan.Yeni_Hayvan_Ekle("Kurt", "Dişi", 5, 3))
85
86     Hayvanlar.extend(Hayvan.Yeni_Hayvan_Ekle("Aslan", "Erkek", 4, 4))
87     Hayvanlar.extend(Hayvan.Yeni_Hayvan_Ekle("Aslan", "Dişi", 4, 4))
88
89     Hayvanlar.append(Hayvan("Avcı", "İnsan", 0, 0, 1))
90
91     print("Hayvan Çiftliğinin İlk Hali\n")
92     for Hayvan in Hayvanlar:
93         print(f"{Hayvan.Cinsiyet} {Hayvan.Tur} Konumu: ({Hayvan.x}, {Hayvan.y}) Hareket Kabiliyeti: {Hayvan.Hareket_kabiliyeti}")
94

```

Çiftliğimize yeni hayvan kayıt ediyoruz ve çiftliğimizdeki tüm hayvanların bilgilerini ekrana yazdırıyoruz.

```

94     print("-----")
95     print("1000 Adımlık Döngü Başlangıcı\n")
96
97     for i in range(1000):
98         for Hayvan in Hayvanlar:
99             Hayvan.Hareket()
100             Hayvan.Avlama()
101             if Hayvan.Ureme(Hayvanlar):
102                 Cinsiyet = random.choice(["Erkek", "Dişi"])
103                 Yeni_hayvanlar = Hayvan.Yeni_Hayvan_Ekle(Hayvan.Tur, Cinsiyet, 1, Hayvan.Hareket_kabiliyeti)
104                 Hayvanlar.extend(Yeni_hayvanlar)
105                 Yeni_dogan = Yeni_hayvanlar[-1]
106                 print(f"Yeni Doğan {Yeni_dogan.Cinsiyet} {Yeni_dogan.Tur} Konumu ({Yeni_dogan.x}, {Yeni_dogan.y}) Hareket Kabiliyeti {Yeni_dogan.Hareket_kabiliyeti}")
107                 Ureme_sayisi += 1
108             print(f"{i}. döngü")
109

```

Çiftliğimizin 1000 adımlık döngüsünü gerçekleştiriyoruz.

```

111     Toplam_hayvan_sayisi = len(Hayvanlar)
112     print("\n\n1000 adımlık döngü sonucu :\n")
113     print(f"Toplam Ureme Sayisi: {Ureme_sayisi}")
114     print(f"Toplam Avlanma Sayisi: {Avlanma_sayisi}")
115     print(f"Kalan Toplam Hayvan Sayisi: {Toplam_hayvan_sayisi}")
116     print("\n\nKalan Hayvan Türleri ve Adetleri :")
117
118     Hayvan_sayilari = {}
119     for Hayvan in Hayvanlar:
120         Tur = Hayvan.Tur
121         Hayvan_sayilari[Tur] = Hayvan_sayilari.get(Tur, 0) + 1
122
123     for Tur, count in Hayvan_sayilari.items():
124         print(f"{Tur}: {count} adet")
125

```

Çiftliğimizin 1000 adımlık döngü sonucundaki son halini ekrana yazdırıyoruz.

Çiftlik simülasyonumuzun çıktı örnekleri.

```

↑
↓
Dişi Kurt Konumu (460, 190) Hareket Kabiliyeti: 3)
Dişi Kurt Konumu (257, 358) Hareket Kabiliyeti: 3)
Dişi Kurt Konumu (101, 481) Hareket Kabiliyeti: 3)
Dişi Kurt Konumu (481, 69) Hareket Kabiliyeti: 3)
Dişi Kurt Konumu (8, 461) Hareket Kabiliyeti: 3)
Erkek Aslan Konumu (16, 465) Hareket Kabiliyeti: 4)
Erkek Aslan Konumu (256, 194) Hareket Kabiliyeti: 4)
Erkek Aslan Konumu (134, 358) Hareket Kabiliyeti: 4)
Erkek Aslan Konumu (350, 332) Hareket Kabiliyeti: 4)
Dişi Aslan Konumu (222, 39) Hareket Kabiliyeti: 4)
Dişi Aslan Konumu (227, 274) Hareket Kabiliyeti: 4)
Dişi Aslan Konumu (63, 234) Hareket Kabiliyeti: 4)
Dişi Aslan Konumu (327, 347) Hareket Kabiliyeti: 4)
İnsan Avcı Konumu (0, 0) Hareket Kabiliyeti: 1)
-----
1000 Adımlık Döngü Başlangıcı

0. döngü
1. döngü
2. döngü
3. döngü
4. döngü
5. döngü
6. döngü
7. döngü
8. döngü
9. döngü

```

```

954. döngü
Kurt Dişi avladı: Koyun Dişi
955. döngü
Yeni Doğan Dişi Koyun Konumu (239, 167) Hareket Kabiliyeti 2)
956. döngü
Yeni Doğan Erkek Koyun Konumu (280, 26) Hareket Kabiliyeti 2)
957. döngü
958. döngü

```

998. döngü
999. döngü

1000 adımlık döngü sonucu :

Toplam Üreme Sayısı: 28
Toplam Avlanma Sayısı: 27
Kalan Toplam Hayvan Sayısı: 80

Kalan Hayvan Türleri ve Adetleri :
Koyun: 35 adet
İnek: 7 adet
Tavuk: 5 adet
Horoz: 9 adet
Kurt: 15 adet
Aslan: 8 adet
Avcı: 1 adet