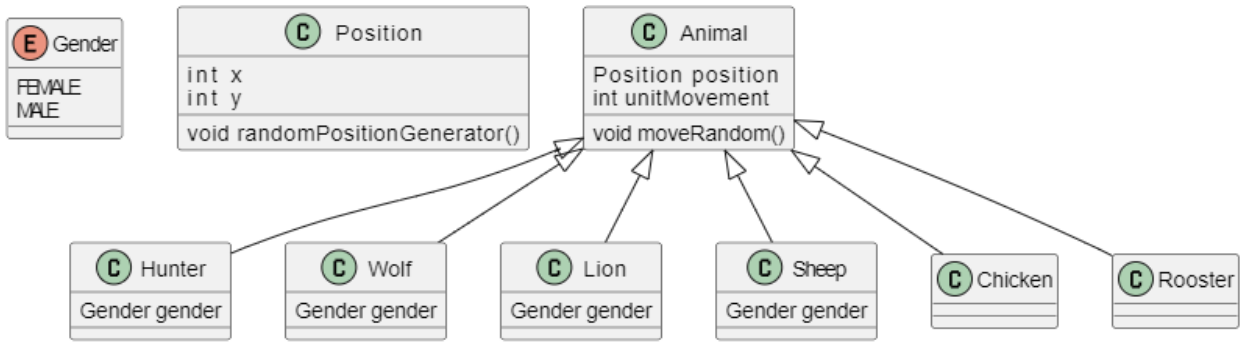
İçindekiler

KonsolTabanlı Hayvanat Bahçesi Uygulaması

Hayvanat Bahçesi Uygulaması, konsolda çalışacak şekilde Java dili üzerinden yazılmıştır. İsterlerde belirtildiği üzere hayvanların rastgele hareket etmesi, aynı türler arasında yeni hayvanların meydana gelmesi ve avlanma gibi durumlar kodlanmıştır.

Uygulamanın UML Diyagramı

Konsol uygulamasında sınıfların birbiriyle olan ilişkisini şematize etmek için Sınıf UML diyagramından yararlanıldı. Sınıflar için gerekli özellik ve yöntemler sınıf diyagramı aracılığıyla belirtildi.



Konsol uygulamasında Animal türünde ata sınıf (parent) ve ondan kalıtım alan alt (child) sınıflar bulunuyor. Animal sınıfının özellik ve metodları alt sınıflar aracılığıyla kullanılmakta. Buna ek olarak bazı sınıfların kendine has metodları bulunmakta.

Position Sınıfı

Sınıflardan üretilen nesnelere konum atamak için Position sınıfından yararlanıldı. Özellik olarak x ve y koordinatlarını belirten değişkenler kullanıldı. Ayrıca isterlerde belirtilen **"rastgele konumlarda doğması"** için void türünde bir fonksiyon tanımlandı.

Gender Enumu

Hayvanlara cinsiyet atamak için enumdan faydalanıldı. Klavyeden girilecek farklı değerlerinin meydana getireceği karışıklığı gidermek adına enum kullanıldı.

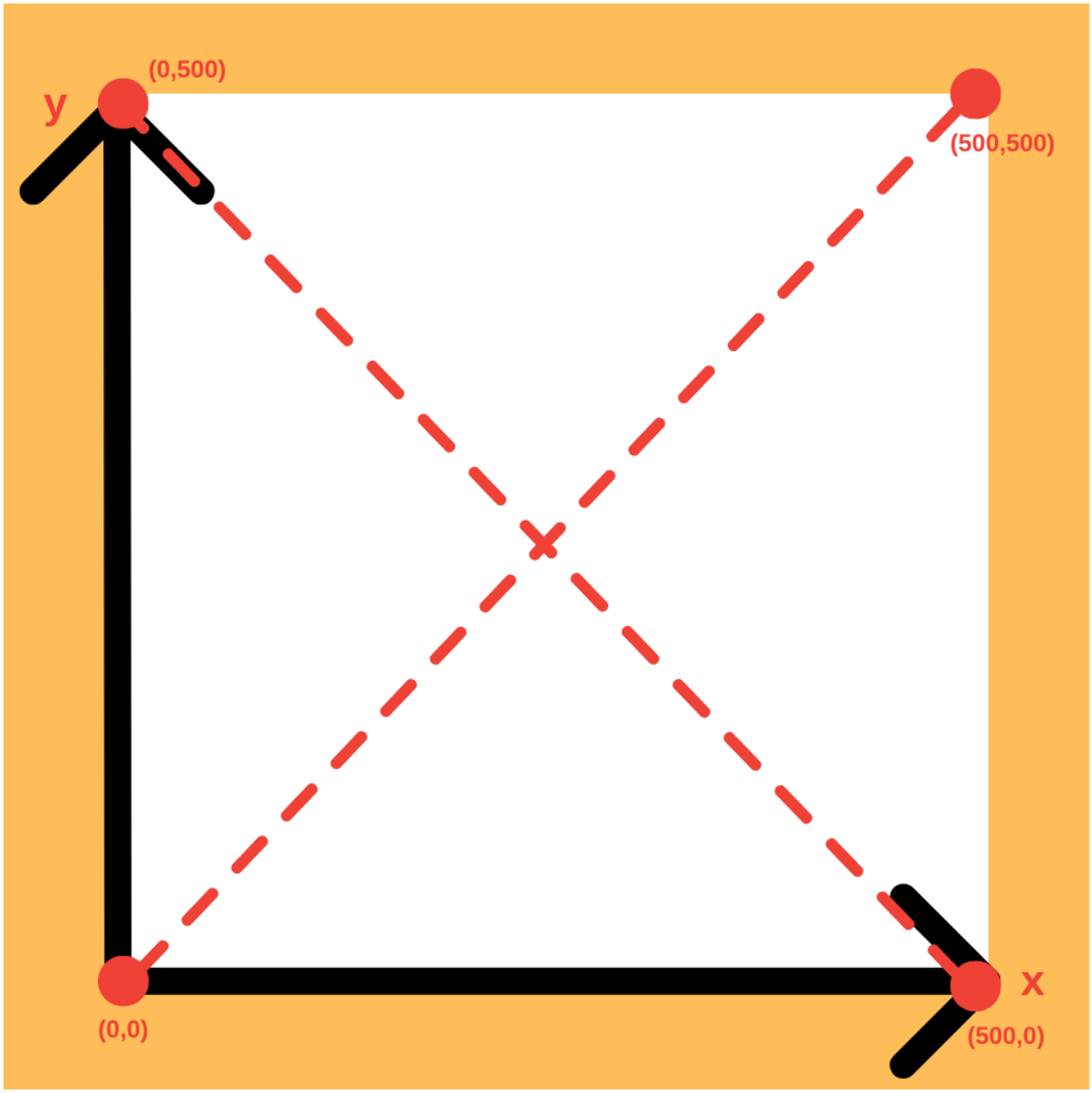
Animal Sınıfı

Özellik olarak birim hareket değerini belirten **unitMovement** ve lokasyon değerlerini belirten Position türündeki **position** değişkenleri kullanıldı. Metod olarak rastgele hareketi sağlayayan void türündeki **moveRandom** ve cinsiyet üzerindeki işlemler için getter/setter kullanıldı.

Alt Sınıflar

Hunter, Wolf, Lion , Sheep , Chicken ve Rooster sınıfları Animal sınıfından kalıtım alacak şekilde düzenlendi. Chicken ve Rooster sınıflarına cinsiyet değişkeni atanmadı.

Uygulamanın Çalışma Mantığı



Uygulama 500 x 500 boyutundaki bir kare üzerinden ilerlemektedir. Belirlenen x ve y koordinatlarının **maksimum** değeri olan 500 birim , hayvanat bahçesindeki tüm elemanların bu sınırlar içerisinde hareket etmesini sağlar.

- Hayvanların position değişkeninin değerleri 500 x 500 birimlik alanın x ve y değerleridir.
- Hayvanlara atanan birim hareket değerleri, hayvanın bu sınırlar içerisinde gideceği mesafeyi ifade eder.
- Alan içerisinde x veya y değerleri arasındaki mesafe en fazla 3 birim olan aynı cinsteki hayvanlar çiftleşebilir.
- Hunter, Wolf ve Lion sınıfının üyeleri , ilgili hayvanları ilgili uzaklık şartını sağlaması durumunda avlar.
- Çiftleşme ve avlanma işlemleri **animals** isimli **ArrayList** üzerinden yapılır. İlgili koşullar sağlandığında listeye ekleme çıkarma işlemleri yapılır.
- Hayvanların hareket ettikten sonra hayvanların birim hareket değerleri 1000 birimden düşer.

Uygulama Kodları

Uygulamada 11 sınıf ve 1 adet enum kullanıldı. Temel işlemler **ZooGeneralOperations** sınıfında yapıldı. Bu sınıf ise Main sınıfındaki **main** fonksiyonu üzerinden çağrıldı.

Position Sınıfı

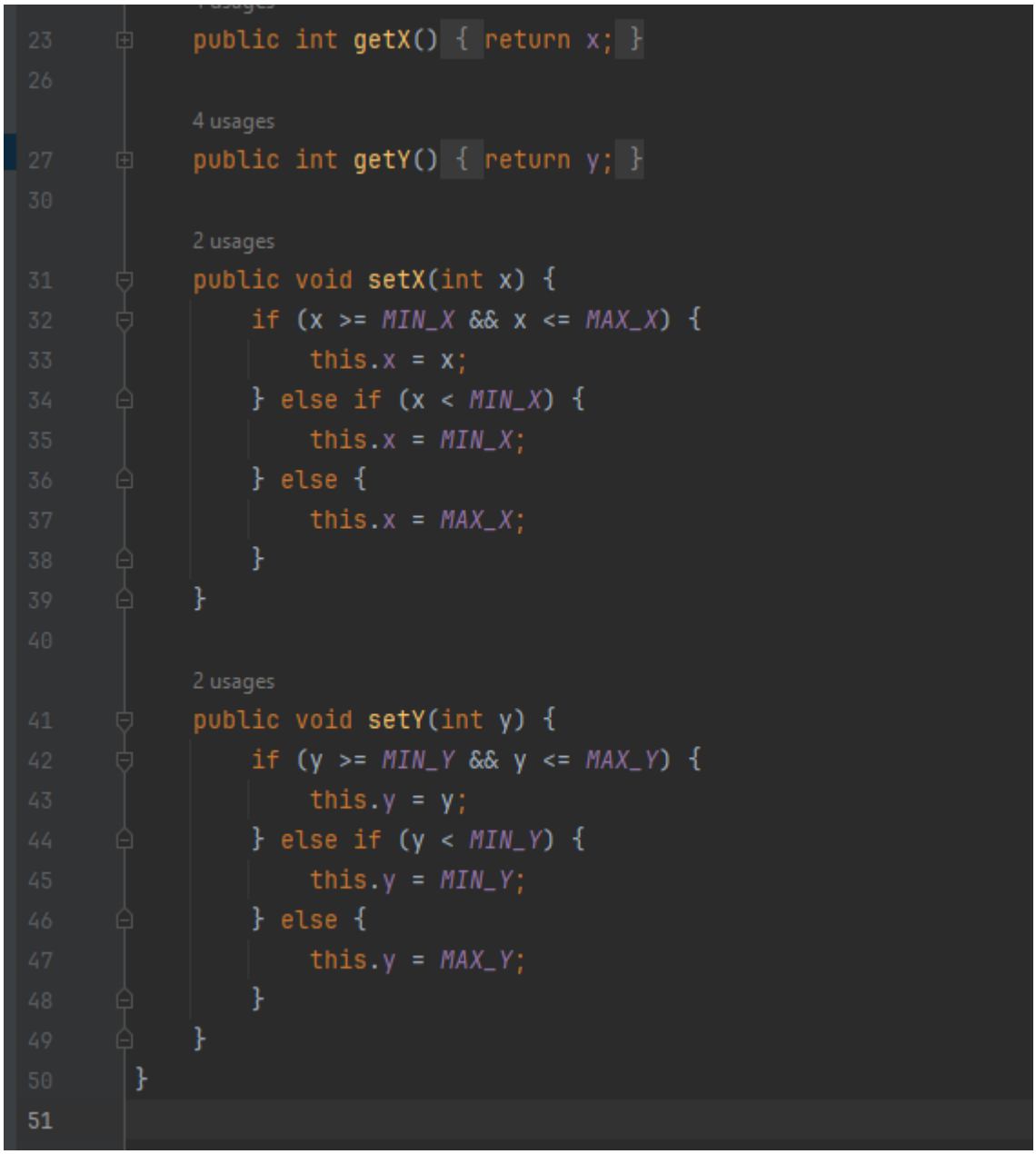
```java
import java.util.Random;

31 usages
public class Position {
    2 usages
    private static final int MAX_X = 500;
    3 usages
    private static final int MIN_X = 0;
    2 usages
    private static final int MAX_Y = 500;
    3 usages
    private static final int MIN_Y = 0;
    5 usages
    private int x,y;


    1 usage
    public Position(int x, int y) {
        this.x = x;
        this.y = y;
    }
    17 usages
    public static Position RandomPositionGenerator(){
        Random random=new Random();
        int random_startX_pos= random.nextInt( bound: 501);
        int random_startY_pos=random.nextInt( bound: 501);
        Position position = new Position(random_startX_pos,random_startY_pos);
        return position;
    }
```

Alanın maksimum değerlerini belirtmek için değişkenler final yapıldı. Constructor kısmında x ve y değişkenleri kullanıldı.

Hayvanları rastgele konumlandırmak için RandomPositionGenerator metodu eklendi. Bu metotla birlikte Random türünde x ve y değerleri oluşturulur. Maksimum değerleri ifade etmek için nextInt kısmındaki sınır kısmına **[0,501)** aralığındaki tam sayıları ifade etmek için 501 yazıldı. Ardından Position sınıfı, rastgele atanan x ve y değerleri üzerinden çağrılır. Çağrılan bu değer metotta döndürülür.



```
23      public int getX() { return x; }
26
        4 usages
27      public int getY() { return y; }
30
        2 usages
31      public void setX(int x) {
32          if (x >= MIN_X && x <= MAX_X) {
33              this.x = x;
34          } else if (x < MIN_X) {
35              this.x = MIN_X;
36          } else {
37              this.x = MAX_X;
38          }
39      }
40
        2 usages
41      public void setY(int y) {
42          if (y >= MIN_Y && y <= MAX_Y) {
43              this.y = y;
44          } else if (y < MIN_Y) {
45              this.y = MIN_Y;
46          } else {
47              this.y = MAX_Y;
48          }
49      }
50  }
51
```

Getter ve setter metotları kullanıldı. Set metotlarında belirtilen değerleri sağladığı takdirde bu kodda x ve y değerleri atanır.

Animal Sınıfı

```java
public class Animal {
    11 usages
    private Position position;
    3 usages
    private int unitMovement;
    //Gender gender;


    public Position getPosition() { return position; }

    public void setPosition(Position position) { this.position = position; }

    5 usages
    public int getUnitMovement() { return unitMovement; }

    no usages
    public void setUnitMovement(int unitMovement) { this.unitMovement = unitMovement; }



    7 usages
    public Animal(Position position,int unitMovement) {
        this.position=position;
        this.unitMovement=unitMovement;
    }
```

- Position ve unitMovement değişkenleriyle constructor oluşturuldu. Geter setter metotları kullanıldı

```java
    protected void moveRandom(){
        Random random=new Random();
        int direction=random.nextInt( bound: 4);

        //0 down
        // 1 up
        // 2 right
        // 3 left represented

        switch (direction){
            case(0):
                position.setY(position.getY()-getUnitMovement());
                break;
            case(1):
                position.setY(position.getY()+getUnitMovement());
                break;
            case(2):
                position.setX(position.getX()+getUnitMovement());
                break;
            case(3):
                position.setX(position.getX()-getUnitMovement());
                break;

        }

    }


}
```

- moveRandom metoduyla rastgele hareket sağlandı. 0,1,2 ve 3 tam sayılarına göre hareket etme özelliği kazandırıldı. Randomdan çıkan sayıya göre hareket sağlanır.

Alt Sınıflar (Hunter,Chicken,Lion,Rooster,Sheep,Wolf)

Animal sınıfından miras alan bazı sınıflarda farklı olarak Gender enumu kullanıldı. Rooster ve Chicken haricinde Gender kullanıldı.

```java
public class Cow extends Animal{
    1 usage
    Gender gender;
    3 usages
    public Cow(Position position, int unitMovement, Gender gender) {
        super(position, unitMovement);
        this.gender=gender;
    }
}
```

Örnek olarak Cow sınıfında Gender kullanıldı. Böylelikle nesne üretilirken cinsiyet özelliği de parametre olarak kullanılmış oldu. Ayrıca her alt sınıfta super() ile ata sınıf olan Animaldaki constructor tanımı kalıtıldı.

```java
public class Chicken extends Animal{
    public Chicken(Position position, int unitMovement) {
        super(position, unitMovement);
    }
}
```

```java
public class Cow extends Animal{
    Gender gender;
    public Cow(Position position, int unitMovement, Gender gender) {
        super(position, unitMovement);
        this.gender=gender;
    }
}
```

```java
public class Hunter extends  Animal {
    public Hunter(Position position, int unitMovement) {
        super(position, unitMovement);
    }

}
```

```java
public class Lion extends Animal{
    Gender gender;
    public Lion(Position position, int unitMovement, Gender gender) {
        super(position, unitMovement);
        this.gender=gender;
    }
}
```

```java
public class Rooster extends Animal{
    public Rooster(Position position, int unitMovement) {
        super(position, unitMovement);
    }
}
```

```java
public class Sheep  extends Animal{
    private Gender gender;
    public Sheep(Position position, int unitMovement, Gender gender) {
        super(position, unitMovement);
        this.gender=gender;
    }
}
```

```java
public class Wolf extends Animal{
    private  Gender gender;
    public Wolf(Position position, int unitMovement, Gender gender) {
        super(position, unitMovement);
        this.gender=gender;
    }
```

```
}
```

İşlemlerin Yapıldığı ZooGeneralOperaitons Sınıfı

İsterlere göre gerekli işlemlerin yapıldığı bu sınıfta birden fazla metot kullanıldı. Yapılacak işler metotlara ayrılarak program modüler hale getirildi.

```
public static List<Animal> animals=new ArrayList<>();
```

Hayvanlar arasında yapılacak işlemler için Animal türündeki bir ArrayListten faydanılındı. Hayvanlar bu liste içinde tutuldu.

```
public ZooGeneralOperaitons() {
    createZoo();
    simulateMovement(1000);
    displayRemainingAnimalCount();
}
```

nesne üretildiği anda programın çalışması amacıyla Constructorun içine metotlar yazıldı.

CreateZoo Metodu

```
public void createZoo(){
    for (int i=0;i<15;i++){
        animals.add(new
Sheep(Position.RandomPositionGenerator(),2,Gender.MALE));
        animals.add(new
Sheep(Position.RandomPositionGenerator(),2,Gender.FEMALE));

    }

    for (int i=0;i<5;i++){
        animals.add(new
Cow(Position.RandomPositionGenerator(),2,Gender.MALE));
        animals.add(new
Cow(Position.RandomPositionGenerator(),2,Gender.FEMALE));
        animals.add(new
Wolf(Position.RandomPositionGenerator(),3,Gender.MALE));
        animals.add(new
Wolf(Position.RandomPositionGenerator(),3,Gender.FEMALE));
    }

    for (int i=0;i<10;i++){
        animals.add(new Chicken(Position.RandomPositionGenerator(),1));
        animals.add(new Rooster(Position.RandomPositionGenerator(),1));
    }
    for (int i=0;i<4;i++){
        animals.add(new
Lion(Position.RandomPositionGenerator(),4,Gender.MALE));
        animals.add(new
Lion(Position.RandomPositionGenerator(),4,Gender.FEMALE));
```

```
    }
    animals.add(new Hunter(Position.RandomPositionGenerator(),1));
}
```

Hayvanat bahçesine eklenecek hayvanların istenilen koşulları sağlayacak şekilde üretimi için işlemlerin yapıldığı metottur. Bu metotta temel olarak:

- istenilen adette ve cinsiyette hayvanlar for döngüsüyle üretilir
- Hayvan sınıflarından nesne üretimi için elle değişken atamak yerine new Anahtar kelimesiyle listeye elemanlar eklenir.
- Hayvan üretimi esnasında constructorun içine Position sınfıında statik olarak tanımlanan RandomPositionGenerator metodu yazılır. Bu sayede konum değişkeni rastgele atanmış olur. Birim hareket parametresine de her hayvanın kendi birim hareket değeri yazılır.

simulateMovement Metodu

```
private void simulateMovement(int remainUnit) {
    Random random = new Random();
    while (remainUnit > 0 && !animals.isEmpty()) {
        for (Animal animal:animals){
            animal.moveRandom();
            remainUnit-=animal.getUnitMovement();
        }
        checkReproduce();
        checkHunt();

    }
}
```

Metodun parametresi int üründeki adım sayısıdır.

- While döngüsü ile kalan adım sayısı kontrol edilir
- Döngüye girildiyse for döngüsü ile animals listesindeki elemanlar gezilir. Her eleman rastgele hareket eder ve her elemanın birim hareket değeri kalan adım sayısından düşer. Örneğin hareket eden avcı ise adım sayısından 1 birim düşülür.
- Üreme ve Avlanma işlemlerini kontrol etmek için metotlar çağrılır.

checkReproduce Metodu

```
public void checkReproduce(){
    List<Animal> newAnimals = new ArrayList<>();

    for (int i=0; i<animals.size();i++){
        Animal firstAnimal=animals.get(i);
        for (int j=i+1;j<animals.size();j++){
            Animal secondAnimal=animals.get(j);
            if (firstAnimal.getClass().equals(secondAnimal.getClass()) &&
distance(firstAnimal,secondAnimal,3))
            {
                if (firstAnimal instanceof Sheep) {
                    newAnimals.add(new
```

```
Sheep(Position.RandomPositionGenerator(), 2, getRandomGender()));
                } else if (firstAnimal instanceof Cow) {
                    newAnimals.add(new Cow(Position.RandomPositionGenerator(),
2, getRandomGender()));
                } else if (firstAnimal instanceof Chicken) {
                    newAnimals.add(new
Chicken(Position.RandomPositionGenerator(), 1));
                } else if (firstAnimal instanceof Rooster) {
                    newAnimals.add(new
Rooster(Position.RandomPositionGenerator(), 1));
                } else if (firstAnimal instanceof Wolf) {
                    newAnimals.add(new
Wolf(Position.RandomPositionGenerator(), 3, getRandomGender()));
                } else if (firstAnimal instanceof Lion) {
                    newAnimals.add(new
Lion(Position.RandomPositionGenerator(), 4, getRandomGender()));
                } else {
                }
            }
        }
    }


    animals.addAll(newAnimals);
}
```

Bu metot temel olarak çiftleşme durumlarını kontrol eder.

- İç içe for döngüleriyle animals listesindeki elemanlar kontrol edilir. Eğer iki eleman aynı sınfıın üyesiyse (aynı sınıftan instance edilmişse) ve aralarındaki uzaklık -distance fonksiyonu ile kontrol edilen- 3 birim veya 3 birimden azsa aynı cinsten yeni bir hayvan animals listesine eklenir.
- Cinsiyet ataması yaparken getRandomGender() metodu ile rastgele cinsiyet atanır.

NOT: animals listesine döngüler aracılığıyla doğrudan ulaşıldığında sonsuz döngüye gireceği için ayrı liste oluşturuldu. Animals listesine eklenecek elemanlar önce ara listeye eklendi.

distance Metodu

```
public boolean distance(Animal animal1,Animal animal2, int targetDistance){
    int X1=animal1.getPosition().getX();
    int Y1=animal1.getPosition().getY();
    int X2=animal2.getPosition().getX();
    int Y2=animal2.getPosition().getY();

    return Math.abs(X1-X2)<=targetDistance || Math.abs(Y1-Y2)<=targetDistance;
}
```

- Parametre olarak Animal türünden iki nesne ve hedeflenen uzaklık değerini alır. Hayvanların x ve y koordinatlarını get metotlarıyla alır.
- İki hayvanın x veya y değerleri arasındaki uzaklık hedeflenen uzaklıktan küçük ya da eşitse true değer döner.

getRandomGender Metodu

```
public static Gender getRandomGender(){
    return Math.random()<0.5 ? Gender.MALE: Gender.FEMALE;
    }
```

- rastgele seçilen değere göre erkek veya dişi cinsiyetleri atanır.

checkHunter Metodu

```
public void checkHunt() {
    List<Animal> removeToAnimals = new ArrayList<>();

    for (int i=0;i<animals.size();i++){
        Animal animal=animals.get(i);
        for (int j=i+1;j<animals.size();j++){
            Animal animal2=animals.get(j);
            if (animal instanceof Hunter && distance(animal, animal2,8)) {
                removeToAnimals.add(animal2);
            } else if (animal2 instanceof Hunter && distance(animal2,
animal,8)) {
                removeToAnimals.add(animal);
            } else if ((animal instanceof Wolf) && (animal2 instanceof Sheep
|| animal2 instanceof Chicken || animal2 instanceof Rooster) &&
distance(animal, animal2,4) ) {
                removeToAnimals.add(animal2);
            } else if ((animal2 instanceof Wolf) && (animal instanceof Sheep
|| animal instanceof Chicken || animal instanceof Rooster) &&
distance(animal2, animal,4)) {
                removeToAnimals.add(animal);
            } else if ((animal instanceof Lion) && (animal2 instanceof Sheep
|| animal2 instanceof Cow) && distance(animal, animal2,5) ) {
                removeToAnimals.add(animal2);
            } else if ((animal2 instanceof Lion) && (animal instanceof Sheep
|| animal instanceof Cow) && distance(animal2, animal,5) ) {
                removeToAnimals.add(animal);
            }


        }
    }
    animals.removeAll(removeToAnimals);
}
```

Bu metotta ise temel olarak avlanma şartları kontrol edilir ve buna göre işlemler yapılır.

- Döngüler aracılığıyla iki hayvan seçilir. Seçilen bu iki hayvanın birbiriyle olan ilişkisi incelenir.
- İsterlerde belirtilen avlanma şartlarına uyan hayvanlara göre listeden kaldırma işlemi uygulanır. Örneğin avcı ile diğer hayvanlar arasındaki mesafe 8 birim veya daha az ise avcının haricindeki hayvan listeden kaldırılır.

NOT: Animals listesinden ayrı liste oluşturuldu. Animals listesine eklenecek elemanlar önce ara listeye eklendi. Sonrasında silinecek elemanların bulunduğu listedeki tüm elemanlar animals listesinden kaldırılır.

displayRemainingAnimalCount Metodu

```
public void displayRemainingAnimalCount(){
        int hunterCount = 0,
                chickenCount = 0,
                lionCount = 0,
                wolfCount = 0,
                roosterCount = 0,
                sheepCount = 0,
                cowCount=0 ;

        for (Animal animal: animals){
            if (animal instanceof Hunter)
                hunterCount++;
            else if (animal instanceof Chicken )
                chickenCount++;
            else if (animal instanceof Lion )
                lionCount++;
            else if (animal instanceof Wolf )
                wolfCount++;
            else if (animal instanceof Rooster )
                roosterCount++;
            else if (animal instanceof Sheep )
                sheepCount++;
            else if (animal instanceof Cow )
                cowCount++;
        }

        System.out.println("Remaining animal counts:");
        System.out.println("Hunter: " + hunterCount);
        System.out.println("Chicken: " + chickenCount);
        System.out.println("Lion: " + lionCount);
        System.out.println("Wolf: " + wolfCount);
        System.out.println("Rooster: " + roosterCount);
        System.out.println("Cow: " + cowCount);
        System.out.println("Sheep: " + sheepCount);
    }
```

- Animals listesi for döngüsü ile dolaşılır. Nesneler hangi sınıftan üretilmişse o sınıftaki eleman sayısını ölçen değişkenin değeri artar. Ardından print fonksiyonlarıyla kalan hayvanların sayısı yazdırılır.