

TASARIM DOKÜMANI

1.	GİRİŞ	1
2.	SİSTEM MİMARİSİ	1
3.	EKLER	3

1. Giriş

Coop Simulation projesi için teknik tasarımı detaylandırılmıştır.

1.1 Amaç

Bu belgenin amacı, Coop simülasyon yazılımının tasarımının ve uygulamasının genel bir görünümünü sağlamaktır.

1.2 Kapsam

Coop, kümesteki hayvan popülasyonunun değişiminin izlenmesine yönelik bir projedir. Proje içerisinde hayvanların yaşam süreleri, doğum, hamilelik, doğurganlık süreleri hesaplanarak kümesteki hayvan popülasyonu simule edilmektedir.

2. Sistem Mimarisi

2.1 Tasarım Detayları

Sistem kullanıcıdan herhangi bir bilgi istemeyecek. Sistem konfigürasyon dosyasından gerekli bilgileri okuyarak çalışmaya başlayacak.

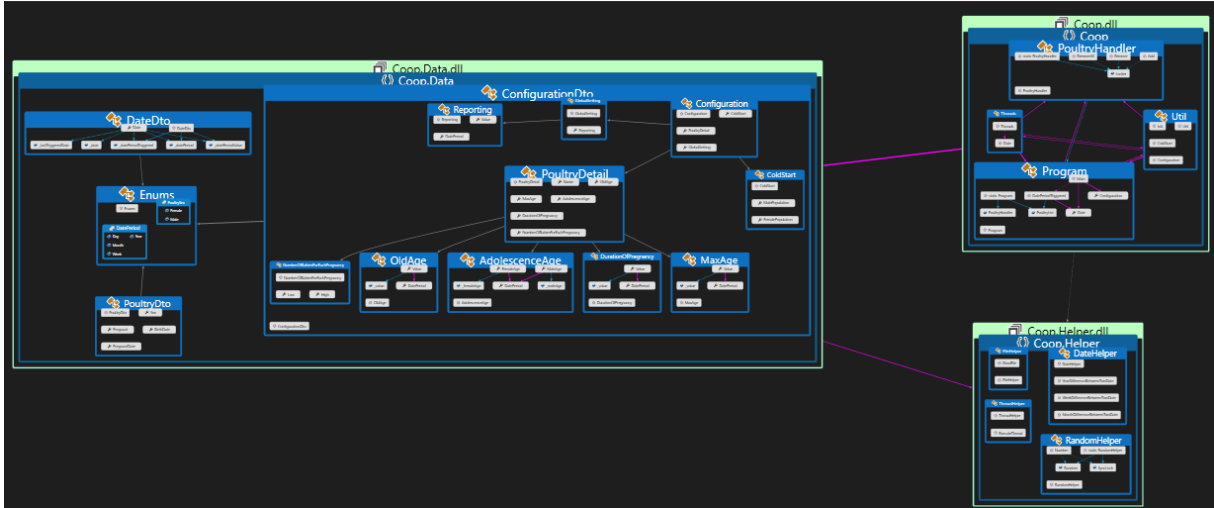
Küme başlangıç anında 1 dişi ve 1 erkek tavşan bulunacak ve konfigürasyon dosyasındaki belirtilen zamanda kümesteki tavşanların yaş ve cinsiyet dağılımları konsol ekranında görüntülenecek.

Uygulamanın başlarken okuyacağı gerekli parametreler;

- Dişi tavşanların bir doğumda doğurduğu tavşan adedi,
- Erkek ve dişi tavşanların ömür süreleri,
- Doğumda oluşan tavşanların erkek veya dişi olma olasılığı,
- Hamilelik süresi,
- Tavşanların yetişkinlik yaşı, dişilerin doğurganlıklarını yitirdikleri yaş.

Uygulama çalıştığındaki tarihi baz alacak ve bu tarihte kümes durumunu hesaplayarak kümesteki tavşan popülasyonunu ekranda görüntüleyecek.

2.2 Class Diyagram



Resim 1-Projenin Class Diyagramı

Resim-1 simülasyon içerisindeki sistemler arasındaki bağlantıları göstermektedir. Daha detaylı görüntüler belgenin ekler kısmında Ek-A'da yer almaktadır. Coop diğer sistemleri(Coop.Data ve Coop.Helper) içeren ana sistemdir. Coop'da Util sınıfı, Configuration.json dosyasını okuyarak kümes ve tavşanların gerekli bilgilerini okur ve başlangıç anının oluşmasını sağlar. Threads sınıfı, kümesteki tavşanların yaş durumlarını kontrol ederek, doğum, çiftleşme, ölüm durumlarını çözümleyerek popülasyonun kontrolünü sağlar. Poultry sınıfı, kümese yeni bireylerin eklenmesini veya çıkarılmasını kontrol etmektedir. Coop.Data'da Enums sınıfı, cinsiyet ve tarihe ait parametreleri(gün, ay...) belirlemektedir. PoultryDto sınıfı, kümesteki hayvanların doğum tarihi, cinsiyet, hamilelik gibi bilgilerini tutmaktadır. DateDto sınıfı programın tarihini alarak tavşanların doğum tarihi, gebelik süresi vb tarihlerin hesaplanması için kullanılmakta. ConfigurationDto sınıfı Configuration.json dosyasındaki kümes ve tavşanlarla ilgili bilgileri için kullanılmakta. Coop.Helper'da ise program içinde kullanılacak olan yardımcı sınıflar bulunmakta. ThreadHelper sınıfı da threadlerin çalışması için oluşturuldu. FileHelper sınıfı dosya okumak için oluşturuldu. DateHelper sınıfı, DateDto'daki iki tarih arasındaki ay, hafta ve yıl hesaplaması için oluşturuldu. RandomHelper sınıfı rastgele sayı üretimi yapılmasını sağlamakta ve program içerisinde doğumda oluşan tavşanların cinsiyetlerini belirlemek için kullanıldı.

2.3 Kullanıcı Arayüzü

Kullanıcılar program çıktısını konsol ekranında görüntüleyecekler.

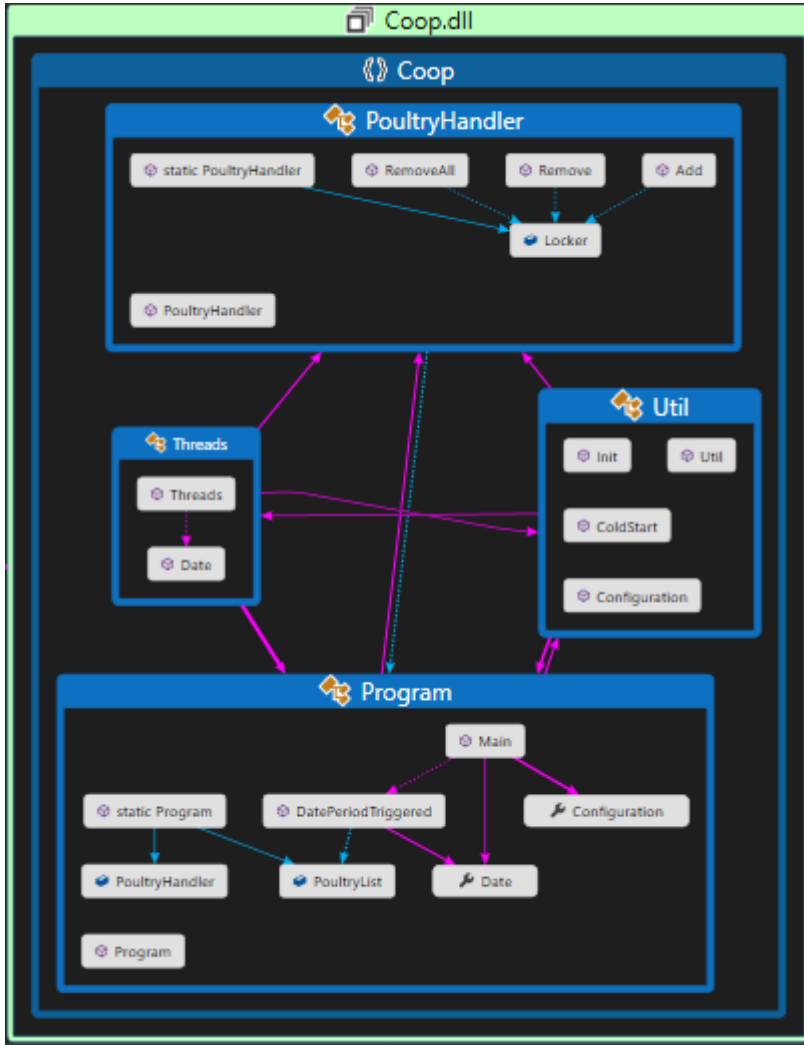
```
C:\Users\gozde\Desktop\Yeni klasör\Coop\Coop\bin\Debug\netcoreapp3.1\Coop.exe

Coop Simulator of Rabbit
01-01-2000 | F: 1 | M: 1 | AVG: 0
01-01-2001 | F: 9 | M: 20 | AVG: 89,41379310344827
01-01-2002 | F: 1175 | M: 813 | AVG: 73,32293762575453
01-01-2003 | F: 133046 | M: 89180 | AVG: 74,76161205259511
```

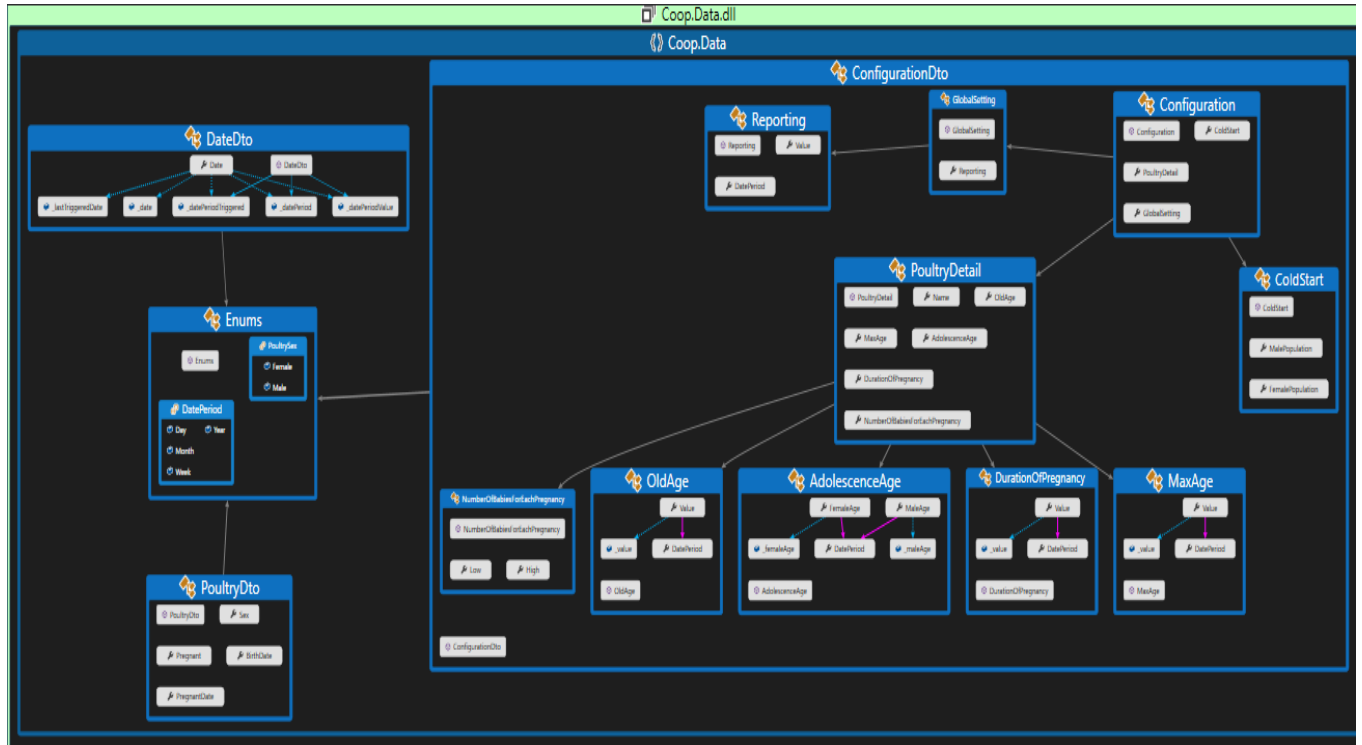
Şekil 2- Konsolda sonuç görüntüsü

3. Ekler

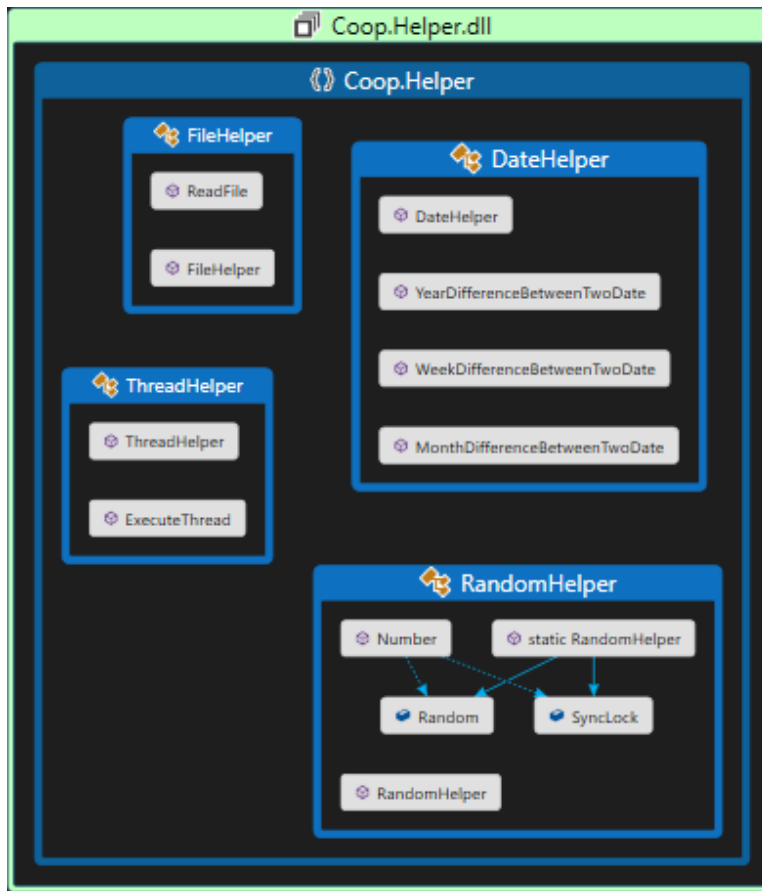
3.1 Ek-A



Şekil 3-Coop Sınıf Diyagramı



Şekil 4-Coop.Data Sınıf Diyagramı



Şekil 5- Coop.Helper Sınıf Diyagramı