

Kestirimle Hedefi Bul

Projenin Amacı: Bu projenin amacı, belirli bir alan içerisinde iki sensör yardımıyla hedefin konumunu tespit eden bir sistem geliştirmektir.

Bu proje, 1000x1000 boyutlarındaki bir alanda herhangi bir yerde konumlanabilen iki sensör ve bir hedef içermektedir. Sensörler, hedefin kerterizini belirleyerek bu bilgiyi TCP/IP protokolü aracılığıyla işleyici birime iletmektedir. İşleyici birim, sensörlerden gelen verileri kullanarak hedefin kesin Kartezyen koordinatlarını hesaplamaktadır. Proje Java programla dili ile geliştirilmiştir.

Bu projenin geliştirilmesi ile aşağıdaki hedeflere ulaşılması amaçlanmaktadır:

- Sensör verileri kullanarak bir hedefin konumunu yüksek doğrulukla tespit edebilme.
- Dağıtık bir sistemde veri toplama ve işleme süreçlerinin entegrasyonu ve yönetimi.
- TCP/IP protokolü kullanarak cihazlar arası iletişim sağlama.
- SOLID prensiplerine uygunluk.

Projenin Çalıştırılması

1. **applications** paketinin içinde yer alan **'ProcessorApplication'** uygulaması çalıştırılır.
2. **applications** paketinin içinde yer alan **'SensorApplication'** uygulaması çalıştırılır.
3. **applications** paketinin içinde yer alan **'TargetApplication'** uygulaması çalıştırılır.
4. Bu işlemler uygulandığında Processor Application konsol çıktısında tespit edilen hedefin konumu yazdırılmaktadır.

Paketler ve Sınıfların Açıklaması

Bu yapı, çeşitli bileşenlerin belirli işlevleri yerine getirdiği modüler bir sistem sunar. Her paketin ve sınıfın işlevselliği aşağıda detaylandırılmıştır.

analyticalgeometry Paketi

Bu paket, analitik geometri ile ilgili hesaplamalar ve veri yapıları içerir.

- **BearingCalculator:** Kerteriz hesaplamasını yapar.
- **IntersectionFinder:** İki doğrunun kesişim noktasını bulur.
- **Line:** Bir doğruyu temsil eder, eğim ve y-eksenini kestiği noktayı hesaplar.
- **Point:** İki boyutlu bir noktayı temsil eder.
- **RandomPointGenerator:** Rastgele noktalar oluşturan bir sınıftır,

applications Paketi

Bu paket, uygulama senaryolarını içerir ve ana çalışma fonksiyonlarını sağlar.

- **ProcessorApplication:** İşlemci ile ilgili uygulamaları temsil eder, verileri işleyerek hedef konumunu belirler.
- **SensorApplication:** Hedeften veri alır ve bu verileri işleyerek kerterizi hesaplar.
- **TargetApplication:** Rastgele bir hedef konumu oluşturur ve bu veriyi sensör uygulamasına gönderir.

communication Paketi

Bu paket, veri iletişimi ile ilgili sınıfları içerir. DataReceiver adlı class abstract tanımlanmıştır. ProcessorDataReceiver ve SensorDataReceiver classları ilgili classı miras almıştır. Tek bir class tanımlayıp, bu classta şart ifadeleri kullanılarak işlemleri yapmak SOLID'e aykırı olacağı için böyle bir yaklaşım benimsenmiştir.

- **DataReceiver:** Verileri almak için kullanılan soyut bir sınıftır.
- **DataSender:** Verileri göndermek için kullanılır.
- **ProcessorDataReceiver:** İşlemci verilerini almak için özelleştirilmiş bir sınıftır.
- **SensorDataReceiver:** Sensör verilerini almak için özelleştirilmiş bir sınıftır.

components Paketi

Bu paket, sistemin 3 ana bileşenini içerir.

- **Processor:** Verileri işleyen bileşendir, sensör verilerini kullanarak kesişim noktalarını hesaplar.
- **Sensor:** Belirli bir konumda bulunan ve verileri toplayan sensörleri temsil eder.
- **Target:** İzlenmesi gereken hedefi temsil eder.

exceptions Paketi

Bu paket, özel durumlar ve hatalar için sınıflar içerir.

- **SensorPositionException:** Sensörlerin konumlarıyla ilgili hataları temsil eden özel bir error handling classıdır.

Örnek Program Çalıştırılması

1. applications paketinin içinde yer alan 'ProcessorApplication' uygulaması çalıştırılır.

```
Processor application is waiting for client(s) on port 12346
```

2. applications paketinin içinde yer alan 'SensorApplication' uygulaması çalıştırılır.

```
Sensor application is waiting for client(s) on port 12345
```

3. applications paketinin içinde yer alan 'TargetApplication' uygulaması çalıştırılır.

```
Target Location: x:242.2 y:-316.3
```

4. SensorApplication uygulamasında Sensörlerin koordinatları ve kerteriz değerleri yazdırılır.

```
Sensor application is waiting for client(s) on port 12345
Client connected: /127.0.0.1
Received Data from Target Application: x:242.2 y:-316.3
Sensor 1 Location : x:-381.5 y:392.93 Bearing: 138.6714597294155
Sensor 2 Location : x:455.81 y:-105.57 Bearing: 225.38886108166272
```

5. ProcessorApplication uygulamasında tespit edilen hedefin koordinatları yazdırılır.

```
Processor application is waiting for client(s) on port 12346
Client connected: /127.0.0.1
Client connected: /127.0.0.1
Detected Target Location: x: 242.2 y: -316.3
```

CLASS DIAGRAM

