# **Fejlesztői Dokumentáció**

## **Projekt áttekintés**

A program egy szenzorhálózat szimulációját valósítja meg. A szenzorok adatok generálásával méréseket végeznek, melyeket eltárol a rendszer, majd ezek az adatok több formátumban menthetők (XML, JSON). LINQ-lekérdezések segítségével elemzések is végezhetők az adatokon.

## **Mappa Struktúra**

* **Szenzorhalozat** (Főprogram)
* Program.cs
* **SzenzorhalozatLibrary** (Class Library)
* Sensor.cs
* SensorData.cs
* SensorDataEventArgs.cs
* SensorNetwork.cs

## **Fő funkciók**

1. **Szenzorok kezelése**: A program egy dinamikus szenzorhálózatot kezel, ahol szenzorokat lehet hozzáadni.
2. **Adat generálás**: Minden szenzor mérési adatokat generál, amelyeket naplóz a rendszer.
3. **Adatmentés**: Az adatok menthetők JSON és XML formátumban.
4. **Adatelemzés LINQ-lekérdezésekkel**: Elemzés történik, pl. átlagértékek, adott küszöbérték feletti adatok.
5. **Eseménykezelés**: Az adatok fogadásához események használhatók.

## **Kód Részletek**

### **1. Főprogram (Program.cs)**

A program belépési pontja.

#### **Funkciók:**

* **Main()**:
* Szenzorhálózat inicializálása.
* Szenzorok hozzáadása.
* Adatok generálása és naplózása.
* Eredmények közlése LINQ-lekérdezések segítségével.
* Az adatok mentése JSON és XML formátumba.
* A felhasználó tájékoztatása a kimeneten keresztül.
* **OnDataReceived()**: Eseménykezelő metódus, amely a szenzor adatok fogadásáért felelős. Megjeleníti a konzolon az érkezett adatokat.

#### **Kódrészlet:**

static void OnDataReceived(object sender, SensorDataEventArgs e)

{

Console.WriteLine($"Adat érkezett: Szenzor={e.SensorName}, Érték={e.Value}");

}

### **2. Class Library (SzenzorhalozatLibrary)**

#### **a) Sensor osztály**

* **Felelőssége**:
* Egy szenzor működését modellezi.
* Adatokat generál.
* **Tulajdonságok:**
* Name: A szenzor neve.
* **Metódusok:**
* GenerateData(): 0 és 100 közötti véletlenszerű értéket generál.

#### **b) SensorData osztály**

* **Felelőssége**:
* Egy mérési adatot tárol.
* **Tulajdonságok:**
* SensorName: A szenzor neve.
* Value: A mért érték.

#### **c) SensorDataEventArgs osztály**

* **Felelőssége**:
* Az eseménykezeléshez szükséges adatokat tartalmazza.
* **Tulajdonságok:**
* SensorName: A szenzor neve.
* Value: A mért érték.

#### **d) SensorNetwork osztály**

* **Felelőssége**:
* A szenzorhálózat működését irányítja.
* **Tulajdonságok:**
* sensors: A szenzorok listája.
* dataLog: A mérési adatok naplója.
* **Metódusok:**
* AddSensor(Sensor sensor): Szenzor hozzáadása a hálózathoz.
* GenerateData(): Adatok generálása az összes szenzorral.
* SaveToXml(string fileName): Adatok mentése XML formátumban.
* SaveToJson(string fileName): Adatok mentése JSON formátumban.
* GetHighValues(int threshold): LINQ-lekérdezés, amely visszaadja a küszöbérték feletti adatokat.
* GetAverageValue(): LINQ-lekérdezés, amely visszaadja az átlagos mérési értéket.
* GetSensorsWithData(): Azokat a szenzorokat adja vissza, amelyekhez tartozik adat.

## **Használati útmutató**

### **1. Futtatás**

1. Hozzáadj egy referenciát a SzenzorhalozatLibrary-hez a Szenzorhalozat projektben.
2. Fordítsd le és futtasd a Szenzorhalozat projektet.
3. A program konzolon jeleníti meg az eredményeket.

### **2. Mentett adatok elérése**

* A program a futtatható fájl mappájába menti az adatokat:
* **adatok.xml**: XML formátumban.
* **adatok.json**: JSON formátumban.

### **3. Elemzések**

* LINQ-lekérdezések eredményeit a konzolon látod.
* **Küszöbérték feletti adatok:** 50 feletti mért értékek.
* **Átlagos érték:** Az összes mérés átlaga.
* **Adattal rendelkező szenzorok:** Azon szenzorok nevei, amelyek generáltak adatot.

## **Hibakezelés**

1. **Fájlok elérése:** Győződj meg róla, hogy a programnak van írási jogosultsága a mentési mappában.
2. **Adat generálás hiba:** Ha egyetlen szenzor sincs hozzáadva, az GenerateData() metódus nem fog adatot generálni.
3. **LINQ-lekérdezés hiba:** Ha nincs mért adat, a lekérdezések üres eredményt adnak vagy kivételt dobhatnak.

## **Fejlesztési lehetőségek**

1. **GUI környezet fejlesztése:** Az eredmények vizuális megjelenítése.
2. **Adatbázis integráció:** Adatok mentése és betöltése adatbázisból.
3. **Konfigurálható küszöbérték:** A felhasználó adhasson meg egyedi küszöböt az elemzésekhez.