

SZENZORHALÓZAT - DOKUMENTÁCIÓ

I. FELHASZNÁLÓI DOKUMENTÁCIÓ

1. FELADAT MEGFOGALMAZÁSA

A Szenzorhalózat projekt egy szenzoradatgyűjtési alkalmazás, amely:

- Különböző típusú szenzorok (hőmérséklet, forgás, vibráció, CO₂, nyomás) szimulációját valósítja meg
- Automatikusan gyűjti és tároja a mérési adatokat egy beágyazott adatbázisban
- Interaktív menürendszer biztosít az adatok szűréséhez és exportálásához
- Időpecsétel-alapú gyűjteményeket használ az adatok elkülönítésére

Cél: Egy rugalmas, bővíthető szenzoradatgyűjtési platform fejlesztése, amely valós szenzorműködést szimulál és az adatokat persistent módon tároja.

2. BEMENET ÉS VÁRT KIMENET

BEMENET:

- Szenzortípusok: 6 szenzorobjektum (2×TEMP, 1×ROT, 1×VIB, 1×CO2, 1×PRES)
- Mérési ciklus: 5 futtatás
- Interaktív menü választások (1-4 opció)

VÁRT KIMENET:

A) Alkalmazás indítása után:

- Szenzor inicializálás lista (6 szenzor CompID-val)
- Mérési adatok (5 ciklus × 6 szenzor = 30 mérés)
- Adatbázis statisztika (T20251210155924 - 6 elem, T20251210155924_Adatok - 30 elem)

B) Interaktív menü:

Main Menu opcióinak kijelzése:

- 1. List Sensors (szenzor listázása típus szerint LINQ szűréssel)
- 2. Export JSON (sensors_export.json fájl létrehozása)
- 3. List Database (mérési adatok megjelenítése)
- 4. Exit (alkalmazás bezárása)

3. RÖVID KEZELÉSI ÚTMUTATÓ

TELEPÍTÉS:

- Projekt klónozása/letöltése
- cd /home/dev/mnt/szofi/beadando/Szenzorhalozat
- dotnet restore
- dotnet build

FUTTATÁS:

```
dotnet run --project Szenzorhalozat.Console/Szenzorhalozat.Console.csproj
```

MENÜ KEZELÉSE:

- Válassz egy számot (1-4) a kérdésre válaszolva

- 1: Szenzorokat típus szerint szűrhetsz (OfType<T>() LINQ)
- 2: JSON exportálás (sensors_export.json)
- 3: Adatbázis lekérdezés
- 4: Kilépés (adatbázis korrekt lezárása)

II. FEJLESZTŐI DOKUMENTÁCIÓ

1. PROJEKT FELÉPÍTÉSE

KÖNYVTÁRSTRUKTÚRA:

```

Szenzorhalozat/
  ├── Szenzorhalozat.sln (Visual Studio Solution)
  ├── Szenzorhalozat.Core/ (Üzleti logika)
    |   ├── Sensors.cs (szenzor osztályok)
    |   ├── Szenzorhalozat.cs (rendszer vezérlő)
    |   ├── database.cs (LiteDB adatbázis kezelés)
    |   ├── AdatgyujtoAllomas.cs (mérési adat gyűjtés)
    |   └── MeresiAdat.cs (mérési adat model)
  ├── Szenzorhalozat.Console/ (Konzol alkalmazás)
    |   └── Program.cs (belépési pont, interaktív menü)
  └── Meres.db (LiteDB adatbázisfájl, futáskor jön létre)

```

2. SAJÁT FEJLESZTÉSŰ OSZTÁLYOK LEÍRÁSA

SENSOR - Absztrakt alap osztály

Cél: minden szenzortípus közös viselkedésének definiálása

Tagváltozók:

- [BsonId] int Id - LiteDB automatikus ID
- string Name - szenzor neve
- string Type - szenzor típusa (TEMP, ROT, stb.)
- string Unit - mértékegység
- double CurrentValue - aktuális mért érték
- double[] MinMax - min/max tartomány [0]=min, [1]=max

Metódusok:

- void Meres() - mérés triggere (ValueUpd, StatusUpdate, event hívás)
- abstract string StatusUpdate() - státusz frissítés
- event Action<MeresiAdat>? MeresiAdatKeszult

DATABASE - LiteDB adatbázis kezelés

Cél: Persistent tárolás és lekérdezés

Kollekciónevezés:

- T[YYYYMMDDHHmmss] - szenzor adatok
- T[YYYYMMDDHHmmss]_Adatok - mérési adatok

Metódusok:

- AddSensor(Sensor) - szenzor INSERT/UPDATE
- AddMeresiAdat(MeresiAdat) - mérési adat INSERT
- GetAllMeresiAdatok() - összes mérési adat lekérdezése
- GetAllTables() - adatbázis statisztika

ADATGYUJTTOALLOMAS - Mérési adat gyűjtés

Cél: Event alapú mérési adat kezelés

- List<MeresiAdat> Adatok - memóriabeli gyűjtés
- Database? Database - null-safe adatbázis referencia
- MeresiAdatFogadas(MeresiAdat) - event handler

3. VÁLASZTOTT ADATSZERKEZETEK

ADATBÁZIS: LiteDB

Miért LiteDB?

- ✓ Beágyazott - nincs külön szerver
- ✓ C# natív integráció
- ✓ BSON formátum - flexibilis dokumentumok
- ✓ Automatikus ID kezelés ([BsonId])
- ✓ ACID tranzakciók

SZENZOR ÉRTÉKEK: Random szám generálás

Metódus: Random.Shared.NextDouble() * (Max - Min) + Min

Előny: Valós szenzor viselkedés szimulációja, egyenletes eloszlás

EVENT-DRIVEN ARCHITEKTÚRA

Az alkalmazás event-alapú kommunikációt használ:

- Sensor.MeresiAdatKeszult event - mérési adat rendelkezésre állása
- Szenzorhalozat.MeresTrigger event - mérés indítása
- AdatgyujtoAllomas.MeresiAdatFogadas() event handler