**Felhasználói dokumentáció**

Egy szenzorhálózat térben szétszórt autonóm szenzor-csomópontokból áll, amelyek figyelemmel kísérik és rögzítik a környezet fizikai állapotát, és képesek arra, hogy egymással együttműködve eljuttassák a mért adatokat egy központi adatgyűjtő állomáshoz. Egy-egy szenzorhálózatnál a monitorozott paraméterek lehetnek: hőmérséklet, páratartalom, túlfolyó tartályok vízszintje, gépek állapotjellemzői, folyó vízszintje stb.

Készítsen C# (.NET Framework) konzolalkalmazást, amely egy hőmérsékltetet mérő és dokumentáló szenzorhálózat szimulációját valósítja meg. A mérési adatokat véletlenszám-generálással állítsák elő. Az adatokon végezze el a következő feladatokat.

[DLL fájl létrehozása és használata]

1. Hozzon létre egy új C# (.NET Framework) projektet a Visual Studio-ban és adjon hozzá a projekthez egy új SzenzorHalozat.dll-t egy Szenzor nevű osztállyal! Az osztály legyen képes tárolni és kezelni a szenzorok azonosítójának értékét és egy AdatGeneralas nevű függvény visszatérési értékeként hozzon létre a szenzorokhoz két tizedes jegyes lebegőpontos random értékeket 15 és 29 között! Adja hozzá referenciaként a főprogramhoz az elkészített DLL fájlt! *(Tóth Krisztina)*
2. A screenshot of a computer program

   Description automatically generatedA főprogramban hozza létre a szenzorok objektumait egyedi azonosítókkal! (lásd példa) *(Tóth Krisztina)*
3. A black screen with white text

   Description automatically generatedIrassa ki a keletkezett adatokat a konzolra, amelyhez hozzon létre egy SzenzorErtekek osztalyt amely tulajdonságként tartalmaz: egy új azonosítót, a szenzorok azonosítóját, a mért értéket és az aktuális időpontot. A főprogram képes legyen ezeknek paramétert átadni! *(Tóth Krisztina)*

[Adatok elhelyezése adatbázisban]

1. A létrehozott adatokat helyezze el egy ,,szenzorhalozat.db” nevű LiteDB adatbázisban és irassa ki annak annak adatait! *(Tóth Krisztina)*

A screenshot of a computer code

Description automatically generated

1. [LINQ LEKÉRDESÉSES FELADATOK]
   1. LINQ lekérdezéssel kérje le azokat a mérési adatokat, ahol 25°C fölötti a mért hőmérséklet. *(Nagy Máté Iván)*
   2. Kérje le a mérési adatok segitségével az átlag hőmérsékletet. *(Nagy Máté Iván)*
   3. Csoportosítsd az adatokat a szenzorok azonosítói szerint, és jelenítsd meg, hány mérési adat tartozik egy-egy szenzorhoz. *(Nagy Máté Iván)*
2. [Delegált használata]
   1. Készíts egy SzenzorAdatfeldolgozo nevű osztályt, amely tartalmaz egy delegáltat SzenzorAdatKezelo néven. A delegált olyan metódusokat ír le, amelyek két paramétert fogadnak: egy string típusú szenzorazonosítót és egy double típusú mért értéket, visszatérési érték nélkül. Az osztálynak tartalmaznia kell egy AdatErkezett nevű eseményt, amely ezen delegált típusával van definiálva. *(Nagy Máté Iván)*
   2. Hozz létre egy AdatFeldolgozas nevű metódust az osztályban, amely végigiterál egy List<Szenzor> típusú listán, és minden egyes szenzorra kiváltja az AdatErkezett eseményt a szenzor azonosítójával és az általa generált véletlenszerű mért értékkel. *(Nagy Máté Iván)*
3. [Legalább 1 eseménykezelés]
   1. Készitsen egy C# konzol applikációt, amely egy egyszerű eseménykezelési mechanizmust valósit meg. Hozzon létre egy EsemenyKezeles osztályt, amely tartalmaz egy eseményt. Készítsen egy Kezelo metódust, amely a lefutás után írja ki a képernyőre, hogy „A lefutós megtörtént”. *(Nagy Máté Iván)*

[JSON fájl létrehozása]

1. A black background with white text

   Description automatically generatedA már beolvasott szenzorok adatait irassa ki egy ,,jsonadatok.json” nevű JSON fájlba is, adjon felhasználói visszajelzést a fájl elkészültéről! *(Tóth Krisztina)*

**Fejlesztői dokumentáció**

Ez a dokumentáció a „beadando\_LHPSQE.sln” nevű C# projektünkhöz készült, amelynek lényege, hogy egy hőmérsékletet mérő szenzorhálózat értékeit hozza létre random szám generátorral és ezeket rendeli egy előre meghatározott azonosítóhoz értékpárként a futtatáskor aktuális idővel együtt kiírva. A program kezel eseményt, használ delegáltat, az adatokat képes LiteDatabase adatbázisban elhelyezni, kiiratni, hozzá kapcsolódó LINQ lekérdezéseket kezelni és ugyanezen adatokat JSON fájlba exportálni is. A program ad felhasználói visszajelzéseket futtatáskor az adott eredményekkel kapcsolatban.

**Használt program:** Microsoft Visual Studio 2022

**Projekt típusa, verziószáma:** Console App (.Net Framework) 4.7.2

**Használt programnyelvek:** C#, LINQ, JSON

**Használt könyvtárak:**

* Newtonsoft.Json (hozzáadott NuGet Package)
* LiteDB (hozzáadott NuGet Package)
* Szenzorkonyvtar (saját kezűleg készített DLL Szenzor osztállyal) (referenciaként hozzáadva)

**Elérési útja:** beadando\_LHPSQE\Szenzorkonyvtar\bin\Debug\Szenzorkonyvtar.dll

**Porgramkódja:** beadando\_LHPSQE\Szenzorkonyvtar\SzenzorHalozat.cs

**Egyéb futtatáskor létrehozott fájlok:**

* **Fő programkód:** beadando\_LHPSQE\beadando\_LHPSQE\Program.cs
* **Szenzorok értékadó osztálya:** beadando\_LHPSQE\beadando\_LHPSQE\SzenzorErtekek.cs
* **A screenshot of a computer

  Description automatically generatedA screenshot of a computer

  Description automatically generatedLétrehozott dokumentumok:**

**A screenshot of a computer

Description automatically generated**

**További információért látogasson el a projektünk GitHub-oldalára:** https://github.com/Tinaa1001/Szofi\_beadando