

Python Assignment 1

Smart Home IoT Automation Simulator

Leírás:

Ebben a feladatban egy Python alapú IoT szimulátort kell készíteni egy okos otthoni automatizáló rendszernek. A szimulátornak emulálnia kell különböző IoT eszközök viselkedését, amik egy ilyen otthonban találhatóak, például okos lámpák, termosztátok vagy biztonsági kamerák. Egy központi automatizáló rendszerre is szükség van, ami ezeket az eszközöket kezeli. Készíteni kell még ezen felül egy ellenőrző irányítópultot, ami vizualizálja és kontrollálja az okos otthont. Ez a feladat segít abban, hogy a Python programozói képességeket használd objektum orientált fejlesztésre, adat kezelésre, valós idejű adat ellenőrzésre és grafikus felhasználói felület létrehozására (GUI).

Objective:

- ◆ Fejlessz egy Python alapú IoT eszköz szimulátort
- ◆ Valósíts meg alap szintű adat analízist és feldolgozást.
- ◆ Készíts egy figyelemfelkeltő ellenőrző felületet, hogy látszódnak a képernyőn a szenzorok által felvett adatok és az analitikai eredmények.
- ◆ Használj Python adat struktúrákat, osztályok, OOP objektumok, példányok és osztály változók, OOP metódusok, pip, static.
- ◆ metódusok, fájlok & standard könyvtár, exception kezelés, modulok és package-ek

Part 1: IoT Eszköz Emuláció (25 %)

- ◆ Eszköz Osztályok: készíts Python osztályokat minden típusú IoT eszköznek, amit szimulálni szeretnél, például: SmartLight, Thermostat, SecurityCamera. Minden osztálynak legyenek attribútumai, mint például: device ID, status(on/off), valamint az eszköznek megfelelő tulajdonságok (hőmérséklet a termosztátnak, fényesség a lámpának, mozgásérzékelés a kamerának stb.)
- ◆ Eszköz Viselkedés: Valósíts meg metódusokat mindegyik eszköz osztálynak, ami lehetővé teszi, hogy ki/be kapcsoljon, vagy megváltozzon bármilyen változója. Szimulálj realisztikus viselkedést, például fokozatosan változó fényességet a lámpának(amikor állítod a fényességét) vagy határok megszabását egy termosztátnak (ne lehessen 200 fokra feltenni például).
- ◆ Randomizálás: Legyen egy randomizáció mechanizmus, ami ha be van kapcsolva, véletlenszerűen állítja az eszközök értékeit folyamatosan

Part 2: Központi Automatizációs Rendszer (25 %)

- ◆ Automatizációs Rendszer Osztály: Készíts egy központi automatizációs rendszer osztályt, pl:AutomationSystem, ami felelős az eszközök kezeléséért és vezérléséért. Legyenek benne olyan metódusok, amik megtalálnak új eszközöket(ha hozzáadsz egy újat), hozzáadnak új eszközöket és végrehajtják az automatikus feladatokat.
- ◆ Szimulációs Ciklus: Valósíts meg egy szimulációs ciklust, ami periódikusan lefut(pl másodpercenként) és frissíti az eszközök státuszait, automatikus viselkedéseket indít el.

Part 3: Dokumentáció (30%)

- ◆ Dokumentáció: Készíts egy követhető dokumentációt a kódodhoz, amiben vannak osztály leírások, metódus elmagyarázások és instrukciók, hogy futtassuk a kódot, valamint hogy használjuk a grafikus felületet.
- ◆ Adj teszteseteket, amik biztosítják, hogy a szimulátor és az automatizáló rendszer jól működik.

Part 4: Ellenőrző irányítópult (20%)

- ◆ Grafikus Felhasználói Felület(GUI Graphical User Interface): Készíts egy grafikus felületet, ahol követni lehet az eszközök viselkedését, irányítani azokat. Ehhez használhatsz Python GUI könyvtárat, például Tkinter. A GUI mutassa az eszközök státuszát és különböző rájuk jellemző tulajdonságokat, mindegyiket legyen lehetőség irányítani, gombokkal vagy hasonló grafikus eszközökkel.
- ◆ Valós Idejű Adat Ellenőrzés: Legyenek láthatóak valós idejű adatok az eszközöktől. Például hőmérsékletm mozgásérzékelés a kemrának, fényesség szintje a lámpáknak.
- ◆ Felhasználói Interakció: Legyen lehetőség interakcióba lépni az eszközökkel a grafikus felület segítségével, például kikapcsolni a lámpát, változtatni a termosztát hőmérsékletét, ki/be kapcsolni kamerákat.



Segítség:

- ◆ Use Python's random library for generating sensor data.
- ◆ Utilize the statistics module for data analysis.
- ◆ Use a simulation loop for rule execution.
- ◆ Timestamp sensor data with Python's datetime module.
- ◆ Design a GUI (try to be creative) using Tkinter.
- ◆ Document class attributes and methods.
- ◆ Feel free to use ChatGPT, but make sure to be able understand and explain your solution!
- ◆ Do not hesitate to ask your instructor for help and asking for the template for the GUI!

PYTHON
ELTE

Példa GUI:

