

Slimme woning simulatie - Projectopdracht

Python OOP

Inleiding

In deze opdracht bouw je in Python een **objectgeoriënteerde simulatie van een slimme woning**. Je modelleert bewoners, kamers en verbonden apparaten, en simuleert hoe deze met elkaar interageren. Je gebruikt de geziene **OOP-principes** zoals **compositie** en **overerving** om het systeem modulair en uitbreidbaar op te bouwen.

Je voorziet een **simulatiemodus** waarin bewoners zich verplaatsen en apparaten automatisch reageren. Daarnaast genereer je een **statische HTML-webpagina** die de actuele toestand van de woning en apparaten weergeeft.

Afhankelijk van je specialisatie (IoT, Cybersecurity of AI) werk je een **extra challenge** uit. Zie onderaan voor specifieke instructies.

Vereisten

Entiteiten

Je modelleert minstens de volgende klassen:

- **Woning**: bevat meerdere kamers en beheert de apparaten.
- **Kamer**: heeft een naam en bevat apparaten.
- **Apparaat** (*abstracte klasse*): basisklasse voor alle apparaten (status: aan/uit).
 - **Lamp**: kan aan/uit en helderheid aanpassen.
 - **Thermostaat**: kan temperatuur instellen.
 - **Deurslot**: kan vergrendelen/ontgrendelen.
 - **Bewegingssensor**: detecteert beweging.
 - **Rookmelder**: detecteert rook en kan alarm activeren.
 - **Gordijn**: kan automatisch openen/sluiten.
- **Bewoner**: beweegt door de woning.
- **SmartHub**: ontvangt meldingen van apparaten en voert regels uit.
- **Logger**: registreert gebeurtenissen.
- **HTMLGenerator**: maakt een statische site van toestanden per kamer en algemeen overzicht.

Structurering woning

- Je modelleert minstens **zes verschillende kamers** in de woning (bijvoorbeeld woonkamer, keuken, slaapkamer 1, slaapkamer 2, badkamer, gang).

Basisfunctionaliteit

- **Starttoestand:** Initialiseer woning met meerdere kamers, apparaten en bewoners.
- **Simulatie:**
 - Elke tijdstap beweegt een bewoner willekeurig naar een kamer.
 - Bewegingssensoren detecteren aanwezigheid.
 - De SmartHub activeert/deactiveert apparaten op basis van regels.
- **Logging:**
 - Alle gebeurtenissen worden chronologisch gelogd.
- **HTML-output:**
 - Toon actuele status per kamer en algemeen overzicht.
- **Terminal-interface:**
 - Start simulatie en/of manuele acties.

Geavanceerde optie (bonuspunten ;-))

- Voeg scenario's toe ("nachtmodus", "vakantiemodus") waarin gedrag van apparaten verschilt.

Extra Challenges per Profiel (je kiest je eigen traject)

IoT Studenten

- Implementeer eenvoudige "device-to-device" communicatie via de SmartHub (publish/subscribe model).
- Simuleer externe toegang: bewoner kan op afstand status opvragen en apparaten bedienen.
- Reflectie: hoe kan echte IoT-externe toegang beveiligd worden?

Cybersecurity Studenten

- Implementeer authenticatie (bv. pincode) voor kritieke acties (deur openen, externe toegang).
- Simuleer hackingpoging: brute force op pincode, illegale toegang.
- Reflectie: hoe kun je smart homes beter beschermen?

AI Studenten

- Laat apparaten gedrag leren: verzamel data over aanwezigheid en pas automatisch acties aan (bv. lamp aanzetten als bewoner meestal om 21u in woonkamer is).
- Genereer HTML-pagina over geleerde gewoontes.

- Reflectie: welke ethische risico's zijn er bij gedragspredictie?

Technische Richtlijnen

- Werk modulair: gebruik meerdere Python-bestanden/modules.
- Codeer duidelijk en volgens conventies.
- Werk met een virtual environment (`venv`) en maak een `requirements.txt` aan.
- Gebruik alleen zelfgeschreven code.

Inlevering

- Deadline: **vrijdag 30 mei 2025, 23u55**
- Bestand: zip-archief `python_oop_slimmewoning_AchternaamVoornaam.zip`
- Structuur:
 - Codebestanden (geen `venv`-folder meeleveren)
 - Reflectieverslag in Markdown (`reflectie.md`)
 - gegenereerde HTML-website in `_site/` folder
 - Demonstratievideo (maximaal 60 seconden, geformatteerd als MP4 of opgenomen via Panopto)

Reflectieverslag

Je reflectieverslag beschrijft:

- Opbouw en werking van jouw toepassing.
- Belangrijke keuzes en uitdagingen.
- Welke zaken moeilijk of niet opgelost zijn.
- Specifiek antwoord op de challenge voor jouw profiel.
- **Toevoeging:** een **korte demonstratievideo (max. 60 seconden)** waarin je de werking van jouw project toont. Upload je video in MP4-formaat of via Panopto.

Hulpmiddelen

- Vraag hulp tijdens de labo's.
- Gebruik betrouwbare bronnen voor achtergrondonderzoek.
- Geen gebruik van generative AI tools voor code (zelf schrijven verplicht).

Veel succes!

Gebruik deze opdracht om te tonen dat je complexe systemen helder kunt modelleren, simuleren en presenteren!