

PROJEKT: WinterOps

System zarządzania zimową infrastrukturą i bezpieczeństwem

Motyw przewodni:

Miasto / ośrodek narciarski zarządza zimą:

-odśnieżanie

-sprzęt

-alerty pogodowe

-interwencje

Struktura projektu (bazowa)

WinterOps/

|— data/

| |— zones.txt

| |— vehicles.txt

| |— alerts.txt

|

|— loader.py

|— operations.py

|— alerts.py

|— stats.py

|— admin.py

|— main_console.py

|— gui.py

Etap 1 — Start projektu (bardzo prosty)

Cel: odpalenie, brak stresu

uruchomienie programu

menu tekstowe

komunikaty w konsoli

pętle

input / print

Etap 2 — Dane z pliku

Temat: strefy miasta / tras narciarskich

zones.txt

CENTRUM;wysoki

OSIEDLE_PÓŁNOC;średni

TRASA_NARCIARSKA;niski

wczytanie danych

lista stref

priorytet odśnieżania

-listy

-split

-funkcje

Etap 3 — Sprzęt zimowy

vehicles.txt

PŁUG_01;dostępny

PŁUG_02;w_trakcie

POSYPYWARKA;dostępny

wyświetlanie sprzętu

statusy

słowniki

warunki

Etap 4 — Operacja „Odśnieżanie”

przypisanie sprzętu do strefy

zmiana statusu

komunikat sukcesu

tu większość klasy jest już „w grze”

ETAPY ŚREDNIE

Etap 5 — Alerty pogodowe

alerts.txt

-15;SILNY_MRÓZ

ŚNIEG;INTENSYWNE_OPADY

analiza danych

komunikaty ostrzegawcze

-if / elif

-logika biznesowa

Etap 6 — Statystyki zimowe

liczba interwencji

najbardziej obciążona strefa

wykorzystanie sprzętu

-liczniki

-agregacja danych

Etap 7 — Tryb ADMIN

osobne menu

reset danych

podgląd całości

ETAPY wyzwanie

Etap 8 — Symulacja czasu

„godziny zimy”

zmiana statusów w czasie

automatyczne alerty

- pętle czasowe

- stan aplikacji

Etap 9 — Algorytm priorytetów

które strefy pierwsze?

wpływ pogody + ruchu

punktacja

pierwsze realne algorytmy decyzyjne

Etap 10 — Aplikacja desktopowa (GUI)

lista stref

przyciski operacji

status sprzętu

alerty wizualne

(Tkinter / PySide)