

Documentatie proiect POO FixItNow

Aplicatia FixItNow reprezinta o simulare completa a functionarii unui service de reparatii pentru electrocasnice. Scopul principal al aplicatiei este gestionarea angajatilor, a electrocasnicelor care pot fi reparate si a cererilor de reparatie primite de la clienti, dar si modificarea datelor primite.

1 Arhitectura aplicatiei

Aplicatia este dezvoltata respectand principiile programarii orientate pe obiecte, avand o structura modulara. Fiecare clasa este implementata in fisiere separate de tip header si sursa, pentru a asigura lizibilitate si mentenanta usoara. Clasa Angajat este clasa de baza si contine attributele si functionalitatile comune tuturor tipurilor de angajati. Din aceasta clasa deriva clasele Receptioner, Tehnician si Supervizor, fiecare adaugand comportamente specifice.

Pentru gestiunea electrocasnicelor, aplicatia foloseste clasa Electrocasnic ca baza, din care deriva clasele TV, Frigider si MasinaSpalat. Aceste clase contin atat informatii comune, cat si date specifice fiecarui tip de aparat. Clasa Cerere modeleaza o solicitare de reparatie si contine o asociere catre un electrocasnic, fara a detine proprietatea acestuia.

Diagrama aferenta proiectului poate fi vazuta pe github sau in folderul proiectului.

2 Functionalitatea aplicatiei

Aplicatia permite adaugarea, modificarea si stergerea angajatilor din sistem. Pentru fiecare angajat se pot afisa datele personale si salariul curent calculat in functie de vechime, bonusuri si orasul de domiciliu. Sistemul verifica validitatea datelor introduse, precum varsta minima la angajare si corectitudinea CNP ului.

In ceea ce priveste electrocasnicele, utilizatorul poate administra lista de modele si marci care pot fi reparate in service. Se pot adauga sau elimina aparate, iar pentru fiecare electrocasnic sunt pastrate informatii precum anul de fabricatie si pretul de catalog. Aplicatia gestioneaza situatiile in care un electrocasnic nu poate fi reparat prin adaugarea lor intr-o lista specifica ce poate fi afisata.

3 Procesarea si simularea cererilor

Cererile de reparatie sunt citite din fisiere de test si sunt clasificate ca valide sau invalide in functie de posibilitatea service ului de a repara electrocasnicul respectiv.

4 Testare si validare

Pentru testarea aplicatiei au fost create fisiere dedicate in folderul tests, care contin date valide si invalide. Aceste teste urmaresc verificarea tuturor functionalitatilor implementate, inclusiv gestionarea erorilor de citire. Programul continua executia chiar daca sunt detectate date invalide, raportand in mod explicit fiecare problema aparuta. O descriere mai amanuntita a fisierelor de testare poate fi vazuta in readme-ul din folderul /tests.

5 Lucruri interesante:

Mi s-a parut interesant sa lucrez la o aplicatie care pare mai "reala". Adica sav ad cum trebuie sa arata o functie de verificare a cnp-ului si lucruri asemanatoare mi s-au parut super faine.

Complierea codului :)). Cand prime-am cate 100 de mesaje de eroare si nu intelegeam nimic aveam un gand ca ar trebui sa inchid si sa ma ocup alta data de ele, dar imi dadeam seama ca daca nu le rezolv acum nu mai scap de ele.