

Kovács Antal Tamás (ERIP4O) A Programozás Alapjai 3 Házi Feladat Specifikáció

1: A házi feladatom egy olyan program, melyben egy autószervívz nyilvántarthatja az ügyfeleinek, hozzájuk tartozó autó(k)nak és azok javításaiinak adatait. A felhasználó hozzáadhat ügyfeleket, autókat, javításokat az adatbázishoz. Ha már létezik egy ügyfél akkor ahhoz még további autókat adhatunk majd hozzá, vagy ha nem létezik az adott ügyfél, akkor utólagosan is hozzá adhatjuk majd. Ezek mellett rákereshetünk egy ügyfélre vagy autóra, különböző információk alapján. Ha ügyfelet keresünk, a hozzá tartozó név, telefonszám, vagy valamelyik autója alapján tehetjük ezt meg. Autót kereshetünk majd rendszám alapján vagy a hozzá tartozó ügyfél neve alapján. Ügyfeleket és autókat távolíthatunk el az adatbázisból, javításokat szándékosan nem lehet törölni, visszaélések elkerülése végett. Kilstázhatjuk egy autó szervíztörténetét. Lekérdezhetjük azokat az autókat, amelyeknek adott időn belül lejár a műszaki vizsgájuk. Ügyfelekhez tartozó adatok: név, telefonszám, autó(k). Az autókhoz tartozó adatok: márka, típus, szín, rendszám, műszaki vizsga lejárt. Javításokhoz tartozó adatok: Dátum, a javítás leírása (mi lett cserélve, egyéb elvégzett munkák), ár, melyik autóhoz tartozik.

2: A programot megnyitva egy grafikus alkalmazást kap a felhasználó, ami a kezdő képernyővel indul. Ott a következő gombokat találja majd: ügyfél hozzáadása, ügyfél eltávolítása, ügyfél lekérdezése, autó hozzáadása, autó eltávolítása, autó lekérdezése, autó szervíztörténete, javítás hozzáadása, hamarosan lejáró műszaki vizsgák.

Az ügyfél hozzáadásánál a felhasználónak a képernyőn, a következő beviteli mezők jelennek meg: Ügyfél neve, Ügyfél telefonszáma, Autó márkája, Autó típusa, Autó színe, Autó rendszáma, Autó műszaki vizsgájának lejárat ideje (ÉÉÉÉ-HH-NN). Ezek után egy mentés gombot, mely menti a bevitt adatokat. Ha valamelyik mező nincs kitöltve, akkor a felhasználó egy hibaüzenetet fog kapni, hogy mely mezőket nem töltötte ki.

Az ügyfél eltávolítása gombra kattintva a képernyőn megjelenik két beviteli mező, ami az ügyfél nevét és telefonszámát kéri. Ezek mellett egy checkbox fog megjelenni, amit be kell pipálni ahhoz, hogy aktív legyen a törlés gomb. Ezzel erősítjük meg hogy tényleg törölni szeretnénk az adott ügyfelet. Az ügyfél törlésével, az összes hozzá tartozó autó és az autókhoz tartozó javítások is törlődnek! Ha az adott ügyfél nem található az adatbázisban, vagy nem töltöttünk ki minden mezőt, akkor a felhasználó egy hibaüzenetet kap.

Az ügyfél lekérdezése gombra kattintva egy bevitelimező jelenik meg, ahova a felhasználó beírhatja az ügyfél nevét, telefonszámát, vagy autójának rendszámát. A mehet gombra kattintva a program kiírja az ügyfél adatait és a hozzá tartozó autók adatait. Ellenkező esetben jelzést küldünk, hogy nem található az ügyfél az adatbázisban.

Autó hozzáadásánál az ugyan az az ablak jelenik meg, mint a felhasználó hozzáadásánál, viszont itt nem mentjük az adott felhasználót kivéve, ha nem található meg az adatbázisban, akihez csatolni akarjuk az autót, ebben az esetben a felhasználó egy figyelmeztetést kap, ahol eldöntheti, hogy szeretné hozzáadni az új felhasználót, visszalép módosítani, vagy szeretné eldobni a változtatásokat.

Az autó eltávolításánál a következő beviteli mezők jelennek meg: autó neve, autó márkája, autó rendszáma. Nem kitöltött mező esetén hibaüzenetet kap a felhasználó, hogy melyik beviteli mező nincs kitöltve. Az ügyfél eltávolítása módhoz hasonlóan, itt is egy megerősítő checkbox lesz a törlés gomb mellett. Az autó törlésével minden hozzá tartozó javítás törlődik!

Az autó lekérdezésénél a beviteli mezőbe írva az autó rendszámát vagy ügyfél nevét, a program kiírja az autó adatait. Ha nem található a keresett autó az adatbázisban, akkor jelzést adunk.

Az autó szervíztörténete módban az autó rendszámát beírva, a program kilistázza az adott autó összes javítását. Ha nincs találat jelzést adunk a felhasználónak.

A javítás hozzáadása módban a következő beviteli mezők lesznek: javítás dátuma (ÉÉÉÉ-HH-NN), javítás ára, autó rendszáma, javítás leírása. A mentés gombra kattintva mentésre kerül az adott autóhoz tartozó javítás. Hiányos kitöltés esetén hibaüzenetet kap a felhasználó.

A hamarosan lejáró műszaki vizsgák módban megadhatunk egy kezdeti dátumot, egy vég dátumot és a két idő közti intervallumban a program kilistázza az összes olyan ügyfél nevét, telefonszámát, autó márkáját, típusát, rendszámát, melyeknek lejár a vizsgája. Ha nem található ilyen autó, akkor ezt jelezzük.

A kilépés gombra kattintva bezárul a program.

Minden módban megtalálható egy vissza gomb, mely a főmenübe vezet vissza.

3: A program az Auto, Javitas, Ugyfel, osztályokkal lesz megvalósítva. A getterek mellett az ügyfelek osztályban még egy addAuto és egy removeAuto függvény is szerepel, mely segítségével menthetjük az autót a felhasználóhoz. Az AutoSzerviz osztály a következő adatokat tartalmazza majd: List<Ugyfel> ugyfelek, List<Auto> autok, List<Javitas> javitasok. Metódusai: load mely betölti a mentett adatokat a fájlokból. save ami menti az adatokat az autok.dat, javitasok.dat és ugyfelek.dat fájlokba, ezt szerializálással oldom meg. addUgyfel mellyel a felhasználót hozzáadjuk az adatbázishoz. removeUgyfel mellyel eltávolítjuk az ügyfelet az adatbázisból, addAuto ahol eltároljuk az adott ügyfélhez az új autót és hozzáadjuk az autók listához. A removeAuto metódussal töröljük az autót és a hozzá tartozó javításokat. Az addJavitas hozzáadja az új javítást a javítások listához. A lejarovizsgak metódus kilistázza a lejáró listákat. A grafikus felület megoldásánál felhasznált elemek: JButton, JCheckbox, JPasswordField, JList. A save metódus minden mentés és törlés után lefut, így biztosítjuk az automatikus mentést és az adatvesztést elkerüljük. A load metódus a preogram megnyitásakor fut le. A szerializálásnál természetesen figyelembe vesszük verzió azonosítást, mely majd a visszafelé kompatibilitás miatt lesz hasznos. A GUI Swing GUI-val készül majd.