Utazási Iroda Fejlesztői dokumentáció

Fejlesztők:

Kulcsár Dénes

Glonczi Tamás

**Fejlesztői dokumentáció C# konzolprogramhoz**

**Bevezetés**

Ez a dokumentáció a fejlesztők számára készült, akik a C# konzolprogram továbbfejlesztésén szeretnének dolgozni. A dokumentum bemutatja a program felépítését, az adatok kezelését és a menürendszer működését.

**1. Adatstruktúrák**

A program két fő adatstruktúrát használ:

* **Traveler (Utas):**
  + name (név): Az utas neve (string)
  + home\_address (lakcím): Az utas lakcíme (string)
  + phone\_number (telefonszám): Az utas telefonszáma (string)
  + ChangeName(name): Az utas nevének módosítása.
  + ChangeHomeAddress(home\_address): Az utas lakcímének módosítása.
  + ChangePhoneNumber(phone\_number): Az utas telefonszámának módosítása. A képen szöveg, képernyőkép, szoftver, Operációs rendszer látható

    Automatikusan generált leírás
* **Path (Útvonal):**
  + destination (célpont): Az útvonal célpontja (string)
  + price (ár): Az útvonal ára (int)
  + down\_payment (előleg): Az útvonal előlege (int)
  + travelers (utasok): Az útvonalhoz csatlakozott utasok indexei tömbben tárolva (int[])
  + paid\_down\_payment (fizetett előlegek): Az utasok által az útvonalhoz fizetett előlegek indexei tömbben tárolva (int[])
  + ChangeDestination(destination): Az útvonal célpontjának módosítása.
  + ChangePrice(price): Az útvonal árának módosítása.
  + ChangeDownPayment(down\_payment): Az útvonal előlegének módosítása.
  + AddTraveler(index): Új utas hozzáadása az útvonalhoz a travelers tömb indexére.
  + AddPaidDownPayment(index): Új befizetett előleg hozzáadása az útvonalhoz a paid\_down\_payment tömb indexére.
  + RemoveTraveler(index): Utas eltávolítása az útvonalról a travelers tömb indexéről.
  + A képen szöveg, képernyőkép, szoftver látható

    Automatikusan generált leírás
  + A képen szöveg, képernyőkép, szoftver látható

    Automatikusan generált leírás

**2. Adatkezelés**

A program két szöveges fájlt használ az adatok tárolására:

* **travelers.txt**: Minden sor egy utast reprezentál, az adatok "|" karakterrel elválasztva: index | name | home\_address | phone\_number
* **paths.txt**: Minden sor egy útvonalat reprezentál, az adatok "|" karakterrel elválasztva: index | destination | price | down\_payment | travelers (opcionális) | paid\_down\_payment (opcionális)

A program indulásakor a LoadData metódus beolvassa a fájlokat, és az adatokat a travelers és paths listákba tölti. A program futása közben a módosításokat a memóriában tárolja, a program kilépésekor a MainMenu végtelen ciklusából való kilépéskor a travelers és paths listák adatai visszaíródnak a megfelelő fájlokba.

**3. Menürendszer**

A program főmenüje a MainMenu metódusban található. A menü az alábbi opciókat kínálja:

* **1. Add new traveler (Új utas hozzáadása)**: Lehetővé teszi új utas létrehozását, és opcionálisan hozzáadható egy létező útvonalhoz.
* **2. Add new path (Új útvonal hozzáadása)**: Lehetővé teszi új útvonal létrehozását.
* **3. Travelers (Utasok)**: Listázza az összes utast, lehetőséget biztosít az utasok adatainak módosítására.
* **4. Paths (Útvonalak)**: Listázza az összes útvonalat, lehetőséget biztosít az útvonalak adatainak módosítására.
* **5. Exit (Kilépés)**: Kiléptet a programból.

**4. Fejlesztés**

A program jelenleg nem teljesen működőképes. A további fejlesztéshez a következőkre lehet figyelni:

* A AddTraveler és AddPath metódusokban a fájlba írás nincs befejezve. A hiányzó rész a travelers.txt és paths.txt fájlokba való írást valósítja meg a File.AppendAllText metódus helyett. A metódusnak az aktuális utast/útvonalat kell karakterlánccá alakítani a megfelelő formátumban, majd ezt a karakterláncot kell a fájlhoz hozzáfűzni.

**4.1. AddTraveler metódus fejlesztése**

A jelenlegi AddTraveler metódus hozzáadja az utast a travelers listához, de elmulasztja a frissítést a travelers.txt fájlban.

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás

* A metódus a hozzáadott utas indexét, nevét, lakcímét és telefonszámát egyetlen karakterláncba fűzi "|" karakterekkel elválasztva.
* A File.AppendAllText metódus ezt a karakterláncot hozzáfűzi a travelers.txt fájlhoz új sorral együtt.

**4.2. AddPath metódus fejlesztése**

Hasonlóan az AddTraveler metódushoz, az AddPath metódus sem frissíti a paths.txt fájlt az új útvonallal.

* A metódus hasonlóan működik, mint az AddTraveler metódus, de figyelembe veszi a travelers és paid\_down\_payment tömböket is.
* A ciklusok végigiterálnak a tömbökön, és csak akkor fűzi hozzá az utas/előleg indexét a karakterlánchoz, ha az érték nagyobb, mint 0 (érvényes index).
* A metódus eltávolítja a felesleges vesszőt a karakterlánc végéről, ha hozzáadódtak indexek.

**5. Hibakezelés**

A program jelenleg hiányos hibakezeléssel rendelkezik a felhasználói bemenetnél. A fejlesztés során fontos kiterjeszteni az ellenőrzéseket, hogy kezelje a hibás bemeneteket.

* A GetInput metódust ki kellene bővíteni, hogy ellenőrizze a bevitt adat típusát (szám vagy string) a megadott input\_type paraméter alapján.
* A menüopciók kiválasztásánál is érdemes hibakezelést alkalmazni. Ha a felhasználó olyan opciót választ, amely nem létezik, a program jelenleg lefagy. A hibakezeléssel a program egy üzenetet jelenít meg a felhasználónak, és újra kéri a bemenetet.

**5.1. GetInput metódus fejlesztése**

A GetInput metódus kiegészíthető a bemenő adat típusának ellenőrzésével:

A képen szöveg, képernyőkép, szoftver, képernyő látható

Automatikusan generált leírás

A metódus reguláris kifejezéseket használ a bemenő adat típusának ellenőrzésére.

Ha a bemenet nem felel meg a várt típusnak, a program hibaüzenetet jelenít meg, és az input értéke "0" lesz.

**5.2. Menürendszer hibakezelése**

A MainMenu és az almenük metódusai kiegészíthetők a felhasználói választás ellenőrzésével:

A képen szöveg, képernyőkép látható

Automatikusan generált leírás

* A ciklus addig fut, amíg a felhasználó érvényes opciót nem választ.
* Ha a felhasználó olyan számot ad meg, amely nem szerepel a menüben, a program hibaüzenetet jelenít meg, és a ciklus újraindul.

**6. További fejlesztési lehetőségek**

A program számos módon bővíthető a funkcionalitás és a felhasználói élmény javítása érdekében. Néhány ötlet:

* **Utasok és útvonalak keresése/szűrése** - Lehetővé kellene tenni a felhasználók számára, hogy név, célállomás vagy ár alapján keressenek utast/útvonalat.
* **Adatok exportálása/importálása** - A program kiegészíthető olyan funkciókkal, amelyek lehetővé teszik az adatok exportálását CSV vagy más formátumokba, valamint az adatok importálását fájlból.
* **Felhasználói felület (UI) fejlesztése** - A konzol felhasználói felület helyett egy grafikus felhasználói felület (GUI) létrehozása javíthatja a program használhatóságát.

**7. Tesztelés**

A fejlesztés során fontos a folyamatos tesztelés. Különböző teszteseteket kell futtatni a program különböző funkcióinak lefedésére