

HÁZI FELADAT

Programozás alapjai 2.

Tervezés

Catlós Tomás

HM93XY

2025. április 5.

1.Feladat – A választott feladat:

Tervezze meg egy vonatjegy eladó rendszer egyszerűsített objektummodelljét, majd valósítsa azt meg! A vonatjegy a feladatban mindig jegyet és helyjegyet jelent együtt. Így egy jegyen minimum a következőket kell feltüntetni:

- vonatszám, kocsiszám, hely
- indulási állomás, indulási idő
- érkezési állomás, érkezési idő

A rendszerrel minimum a következő műveleteket kívánjuk elvégezni:

- vonatok felvétele
- jegy kiadása

A rendszer később lehet bővebb funkcionalitású (pl. késések kezelése, vonat törlése, menetrend stb.), ezért nagyon fontos, hogy jól határozza meg az objektumokat és azok felelősségét.

Valósítsa meg a jeggyel végezhető összes értelmes műveletet operátor átdefiniálással (overload), de nem kell ragaszkodni az összes operátor átdefiniálásához! A megoldáshoz ne használjon STL tárolót!

2.Feladatspecifikáció:

A rendszer célja, hogy egyszerűsítse a jegyek vásárlását és a felhasználók számára alapvető információkat biztosítson a vonatokról és a kért helyekről.

A rendszer ezeket az alapvető műveleteket végzi el:

- **Vonatok felvétele:** Az egyes vonatok adatait rögzítjük a rendszerben. Ezek az adatok a vonatszám, indulási és érkezési állomás, indulási és érkezési időpont, továbbá a kocsik számát és a szabad helyek számát.
- **Jegy kiadása:** A felhasználók képesek lesznek jegyet venni a vonatokra. A jegyek tartalmazzák az indulás és érkezés helyszíneit és időpontjait, a vonatszámot, a hely sorszámát, jegy azonosítót és esetleges késést is.
- **Egy adott vonat adatainak kiírása:** Egy adott vonat keresése a vonatszám alapján és az adott vonat adatainak kiírása.
- **Vonatok törlése és késés beállítása:** Vonatokat lehetséges lesz törölni és késést is lehetséges lesz beállítani.

Bemeneti és kimeneti adatok:

- Ha a felhasználó új vonatot akar létrehozni akkor a következő adatokat kell megadni: vonatszám, kocsiszám, helyek száma, indulási és érkezési állomás, indulási és érkezési időpont

- Ha a felhasználó jegyet akar venni, akkor megadja a keresett vonat számát és az alapján tud jegyet venni, ha még van hely. Tud választani a helyek közül, ha az adott hely nem foglalt.
- A rendszer miután vonat létrehozása megtörtént vagy jegyfoglalás megtörtént kiírja az adatokat, amiket a felhasználó megadott.
- A program két fájlban fogja tárolni az adatokat: egyikben a vonatok adatait a másikban pedig a foglalt jegyek adatait tárolja
- Ha valamilyen adat érvénytelen vagy esetleg a hely foglalt, akkor a program egy hibaüzenetet ad.

A rendszer működése:

- **Vonatok felvétele:** A vonatok felvétele automatikus történik, ha a programot elindítjuk, a program indításkor egy fájlból beolvassa a tárolt adatokat. Ezután a felhasználó létrehozhat vonatok, az egyes adatok megadásával. Ha rögzítettük a vonat adatait, akkor a felhasználó már tud a vonatra jegyet venni.
- **Jegy kiadása:** Ha a felhasználó jegyet akar venni, akkor a program kikeresi a keresett vonatot a felhasználó által megadott adatok szerint. A program felkínálja a felhasználónak, hogy válasszon a helyek közül, ha a felhasználó nem akar jegyet választani, akkor a program tud véletlenszerűen adni egy helyet. Ha a felhasználó kiválaszt egy helyet és az adott hely már foglalt, akkor hibakódot ad és felkínálja a felhasználónak, hogy válasszon egy másikat. Ha megtörtént a helyfoglalás, akkor a program kiírja a felhasználónak a jegy adatait.
- **Vonatok törlése és késés beállítása:** A felhasználó képes vonatokat törölni a vonatszám alapján. A program képes késést beállítani és miután a késést a felhasználó beállította, a program automatikusan az új jegy vásárlásakor kiírja az aktuális késést.
- **Adatok mentése:** Ha a felhasználó a programot bezárja, akkor a program automatikusan lementi az adatokat a két fájlba.

Hibakezelés:

- Ha a két fájl a program nem tudja megnyitni indításkor, akkor a program hibaüzenetet fog kiírni.
- Ha a felhasználó olyan helyet próbál foglalni, ami már foglalt, akkor a program hibát fog kiírni.
- Ha a felhasználó helytelen információt próbál megadni, akkor a program hibát fog kiírni.

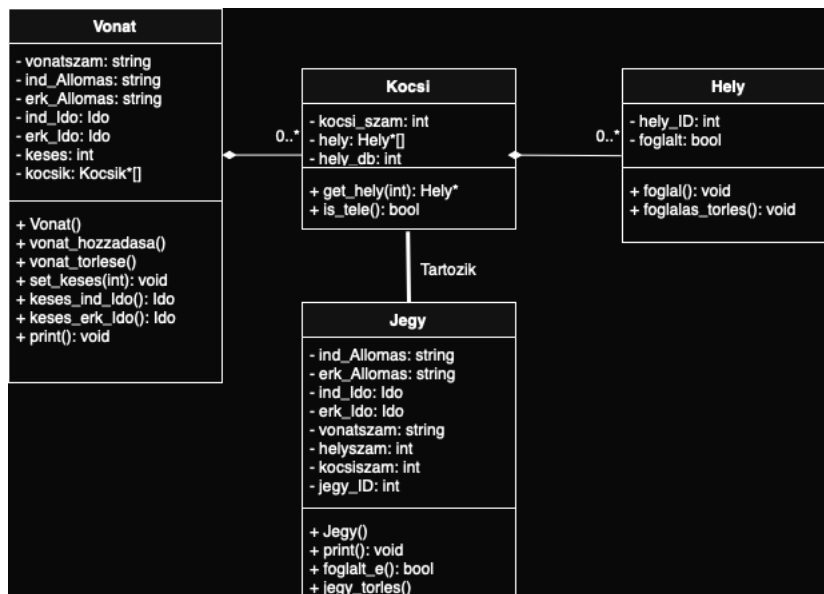
A program indítás után egy menüt fog kiírni és az egyes funkciókba a sorszámuk beírásával lehet belépni. Miután beléptünk egy adott funkcióba, akkor a program fogja kérni a felhasználótól a szükséges adatokat, amiket a funkció követel. A program a funkciók végén kiírja az összes adatot és miután a felhasználó úgy gondolja, hogy minden helyes, akkor a program lementi az adatokat.

4. Terv:

Objektumok:

- **Vonat osztály:** Egy konkrét vonat teljes leírását tartalmazza, valamint a vonathoz tartozó kocsikat. Attribútumai: vonatszám, ind_Allomas, erk_Allomas, ind_Ido, erk_Ido, keses, továbbá kapcsolatban van a kocsi osztállyal, mert egy vonat több kocsit tartalmaz, ezért tartalmaz még egy kocsi tömböt.
- **Kocsi osztály:** Egy vonatkocsi adatait és az abban található üléseket kezeli. Attribútumai: kocsiszám, továbbá kapcsolatban van a hely osztállyal, ezért tartalmaz még egy hely tömböt.
- **Hely osztály:** Egy adott hely foglaltságának nyilvántartása. Attribútumai: hely_ID és foglalt változók.
- **Jegy osztály:** Egy kiadott jegy információit tárolja. Attribútumai: vonatszám, helyszám, kocsiszám, jegyID, ind_Allomas, erk_Allomas, ind_Ido, erk_Ido. A jegyID változó véletlenszerűen jön létre.
- **Ido segédosztály:** Egy időpontot tartalmaz, ora, perc formátumban.
- **VonatRendszer osztály:** A vonatokat és jegyeket kezeli, tárolja a vonatokat és a jegyeket, vonatok és jegyek adatait képes kezelni, frissíti a helyek elfoglaltságát, továbbá vonatokat képes létrehozni, törölni és késéseket beállítani.

UML diagram:



Operátor átdefiniálás: A Jegy osztály esetén átdefiniáljuk ezt a két operátort:

- operator==: Két jegy egyenlő, ha a minden releváns adatuk megegyezik (pl. Ugyanarra a vonaton ugyanarra a helyre szólnak.)
- operator<<: Lehetővé teszi a jegy formázott szöveges kiírását.

- További operátorok is implementálhatók szükség szerint.

Hibakezelés:

- A program hibát dob, ha olyan helyet próbálunk lefoglalni, ami már foglalt, nem található az adott hely, nem található a keresett vonat, nem található a keresett kocsi vagy ha túlindexelni próbálunk.

Tesztprogram:

- A tesztprogram kipróbálja a program egyes funkciót sorban úgy, hogy a tesztelőtől kér adatot és azokat az adatokat dolgozza fel.