NHF – Vonatjegy

Programozás alapjai 2.

DORAU5

Toldi Lőrinc László

2024.05.19.

**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!

**Terv**

**A program célja**  
A feladat egy vonatjegy eladó rendszer megvalósítása, amely képes kezelni a vonatok rendszerbe történő felvételét és törlését, illetve a már mentett vonatokra szóló jegyek kiadását. A rendszerben regisztrált vonatok listájából menetrend lekérdezhető. A program képes a vonatokat és kiadott jegyeket fájl-ba kiírni és fájl-ból beolvasni. A program menüvezérelt, grafikus megjelenítést nem alkalmaz.

**menü megjelenése**  
A menü a konzolban jelenik meg szövegesen. A menüpontok egymást követő sorokba vannak írva, minden sor a megfelelő sorszámmal kezdődik. A választható menüpontok a következőek; (1) jegykiadás, (2) vonat felvétel, (3) vonat törlés, (4) adatok mentése, (5) adatok betöltése, (6) menetrend lekérdezés, (7) kilépés.

**menüpont választás**  
A menüpont választás az ahhoz tartozó szám beírásával majd enter-el történik. (pl. (1) - Jegykiadás esetében az ‘1’ konzolban való bevitelével.) Hibás bevitelével esetén a program a felhasználót tájékoztatja és újbóli választásra kéri. A menüpont választás után a választott menüpont jelenik meg.

**jegykiadás**  
A jegykiadás menüpontban egy jegy kiadására van lehetőség. A felhasználónak elsőnek az indulási és az érkezési állomás nevét kell megadnia, majd az indulási időpontot. A program arra a vonatra ad jegyet, mely az adott állomásra ahhoz leghamarabb érkezik. Amennyiben nem találtunk vonatot a kettő állomás között a felhasználót tájékoztatjuk és felajánljuk a folyamat újrakezdését. Ha találtunk vonatot az indulás időpontját kiírjuk. A felhasználónak a jegy típusát is kikell választania, amelynek ára a menetidőből ered (x Ft/perc alapon). A jegyek típusai a következők: első osztály - 48Ft/perc, másod osztály - 30Ft/perc, diák 12Ft/perc, nyugdíjas 12Ft/perc. Az adatok bekérése egyesével és egymást követően történik. A készített objektumot a rendszerben tároljuk. A ‘kiadott’ jegy a konzolban jelenik meg.

**vonat felvétel**  
A vonat felvétel menüpontban új vonat regisztrálására van lehetőség. A felhasználótól a következő adatokat kérjük be; vonatszám [egész], kocsi szám [egész], kocsikban elérhető helyek száma [egész], állomások száma [egész, minimum 2]. A megadott állomások számának megfelelően annyiszor jelenik meg az állomás bekérése, melyek sorrendje egymást követő kell, hogy legyen. Az állomás bekért attribútumai: állomás neve [szöveg], érkezési idő [óra: egész, perc: egész]. A készített objektumot a rendszerben tároljuk. Hiba esetén a felhasználót tájékoztatjuk és újbóli bevitelre kérjük. Sikeres esetben visszakerülünk a főmenübe.

**vonat törlés**  
A vonat törlése menüpontban létező vonat törlése elérhető. A felhasználótól a vonat számát kérjük be, majd az ahhoz tartozó vonatot töröljük a rendszerből. Hiba esetén a felhasználót tájékoztatjuk és újbóli bevitelre kérjük. Sikeres esetben visszakerülünk a főmenübe.

**adatok mentése**  
Az adatok mentése menüpontban fájl-ba tudjuk kiírni a vonatokat és a hozzájuk tartozó állomásokat, kocsikat és jegyeket. A felhasználótól a fájl nevét kérjük be. Hiba esetén a felhasználót tájékoztatjuk és újbóli bevitelre kérjük. Sikeres esetben visszakerülünk a főmenübe.

**adatok betöltése**  
Az adatok betöltése menüpontban fájl-ból olvassuk be a vonatokat és a hozzájuk tartozó állomásokat, kocsikat és jegyeket. A beolvasott adatokat a rendszerben tároljuk. A felhasználótól a fájl nevét kérjük be. Hiba esetén a felhasználót tájékoztatjuk és újbóli bevitelre kérjük. Sikeres esetben visszakerülünk a főmenübe.

**menetrend lekérdezés**  
A menetrend lekérdezésnél a következő paramétereket kérjük be és azokat szűrőként alkalmazzuk a menetrendre; Indulási állomás ["szöveg"], érkezési állomás ["szöveg"]. Az adatok bekérése egyesével és egymást követően történik. Az üresen hagyott paramétereket nem alkalmazzuk. Az adatok bevitelét követően kiírjuk a megfelelő vonatok számát és listázzuk őket. kilépés A kilépés menüpont leállítja a programot.

**Osztálydiagram**

A diagram of a computer

Description automatically generated

**Tesztelés**

**Ido osztály tesztjei**

Az Ido osztály tesztfüggvényei célja, hogy ellenőrizzék az osztály különböző funkcióinak helyességét és megbízhatóságát.

**IdoKiiras teszt:** Ennek a tesztnek az a célja, hogy ellenőrizze az Ido osztály kiírás funkcióját. Két Ido objektum jön létre különböző időpontokkal. A kiir metódus hívása után ellenőrizzük, hogy a kiírt értékek megegyeznek-e a várt értékekkel.

**IdoPerzisztencia teszt:** Ez a teszt azt vizsgálja, hogy az Ido osztály perzisztenciája megfelelően működik-e. Egy Ido objektumot kiírunk egy stringstream-be, majd ugyanebből a streamből egy másik Ido objektumot olvasunk vissza. Ellenőrizzük, hogy a két objektum azonos-e.

**IdoMuveletek teszt:** Az Ido osztály alapvető műveleteit ellenőrzi ez a teszt. Megvizsgáljuk az objektum létrehozásakor megadott értékek és a setterek működésének helyességét. Ezen kívül ellenőrizzük az aritmetikai műveletek (óra és perc hozzáadása) helyességét, valamint az egyenlőség és különbözőség operátorok működését.

**Allomas osztály tesztjei**

Az Allomas osztály tesztfüggvényei az osztály különböző funkcióit vizsgálják.

**AllomasKonstruktor teszt:** Ez a teszt az Allomas osztály konstruktorainak helyességét ellenőrzi. Teszteljük mind az alapértelmezett, mind a paraméteres konstruktor működését.

**AllomasModositas teszt:** Ennek a tesztnek a célja, hogy ellenőrizze az Allomas osztály attribútumainak módosítását. Vizsgáljuk az indulás és érkezés időpontjainak, valamint az állomás nevének módosítását és ezek helyességét.

**AllomasPerzisztencia teszt:** Ez a teszt az Allomas osztály perzisztenciáját vizsgálja. Egy Allomas objektumot kiírunk egy stringstream-be, majd ugyanebből a streamből egy másik Allomas objektumot olvasunk vissza. Ellenőrizzük, hogy a két objektum azonos-e.

**Utvonal osztály tesztjei**

Az Utvonal osztály tesztfüggvényei az osztály funkcióinak ellenőrzésére szolgálnak.

**UtvonalAllomasHozzaadas teszt:** Ez a teszt az Utvonal osztály állomás hozzáadás funkcióját vizsgálja. Új állomásokat adunk hozzá egy útvonalhoz, majd ellenőrizzük, hogy az állomások helyesen kerültek-e hozzáadásra és a várt sorrendben vannak-e.

**UtvonalAllomasCsere teszt:** Ennek a tesztnek az a célja, hogy ellenőrizze az Utvonal osztály állomás cseréjének helyességét. Kicseréljük az útvonal egyes állomásait és ellenőrizzük, hogy a csere helyesen történt-e meg.

**UtvonalAllomasHozzaadas teszt**: Ez a teszt az Utvonal osztály állomás hozzáadás funkcióját vizsgálja. Új állomásokat adunk hozzá egy útvonalhoz, majd ellenőrizzük, hogy az állomások helyesen kerültek-e hozzáadásra és a várt sorrendben vannak-e.

**UtvonalAllomasCsere teszt:** Ennek a tesztnek az a célja, hogy ellenőrizze az Utvonal osztály állomás cseréjének helyességét. Kicseréljük az útvonal egyes állomásait és ellenőrizzük, hogy a csere helyesen történt-e meg.

**UtvonalPerzisztencia teszt:** Ez a teszt azt vizsgálja, hogy az Utvonal osztály perzisztenciája megfelelően működik-e. Egy Utvonal objektumot kiírunk egy stringstream-be, majd ugyanebből a streamből egy másik Utvonal objektumot olvasunk vissza. Ellenőrizzük, hogy a két objektum azonos-e. Ez a teszt azt biztosítja, hogy az Utvonal objektumok helyesen menthetők és olvashatók vissza.

**UtvonalKiiras teszt:** Ennek a tesztnek az a célja, hogy ellenőrizze az Utvonal osztály kiírás funkcióját. Egy Utvonal objektumot hozunk létre több állomással, majd a kiírási metódust hívjuk. Ellenőrizzük, hogy a kiírt értékek megegyeznek-e a várt értékekkel.

**Vonat osztály tesztjei**

A Vonat osztály tesztfüggvényei célja, hogy biztosítsák az osztály minden funkciójának helyességét és megbízhatóságát.

**VonatKonstruktor teszt:** Ennek a tesztnek a célja, hogy ellenőrizze a Vonat osztály konstruktorainak helyességét. Teszteljük mind az alapértelmezett, mind a paraméteres konstruktor működését. Ellenőrizzük, hogy a megfelelő attribútumok helyesen inicializálódnak-e.

**VonatAllomasHozzaadas teszt:** Ez a teszt a vonathoz állomások hozzáadásának helyességét ellenőrzi. Új állomásokat adunk a vonat útvonalához, majd ellenőrizzük, hogy ezek az állomások megfelelő sorrendben és helyesen kerültek-e hozzáadásra.

**VonatAllomasEltavolitas teszt:** Ennek a tesztnek a célja, hogy ellenőrizze az állomások eltávolításának helyességét a vonat útvonalából. Egy állomást eltávolítunk, majd ellenőrizzük, hogy a többi állomás érintetlen maradt-e és az eltávolítás megfelelően történt-e meg.

**VonatKovetkezoAllomas teszt:** Ez a teszt azt vizsgálja, hogy a Vonat osztály helyesen kezeli-e a következő állomás lekérdezését. Megvizsgáljuk, hogy a vonat megfelelően tudja-e meghatározni a következő állomást az aktuális állomás alapján.

**Menetrend osztály tesztjei**

A Menetrend osztály tesztfüggvényei az osztály különböző funkcióinak ellenőrzésére szolgálnak.

**MenetrendKonstruktor teszt:** Ez a teszt a Menetrend osztály konstruktorainak helyességét ellenőrzi. Teszteljük az alapértelmezett és a paraméteres konstruktorokat, ellenőrizve, hogy a megfelelő attribútumok helyesen inicializálódnak-e.

**MenetrendVonatHozzaadas teszt:** Ennek a tesztnek a célja, hogy ellenőrizze a menetrendhez vonatok hozzáadásának helyességét. Új vonatokat adunk a menetrendhez, majd ellenőrizzük, hogy ezek a vonatok megfelelő sorrendben és helyesen kerültek-e hozzáadásra.

**MenetrendVonatEltavolitas teszt:** Ez a teszt a menetrendből vonatok eltávolításának helyességét vizsgálja. Egy vonatot eltávolítunk a menetrendből, majd ellenőrizzük, hogy a többi vonat érintetlen maradt-e és az eltávolítás megfelelően történt-e meg.

**MenetrendKereses teszt:** Ennek a tesztnek az a célja, hogy ellenőrizze a menetrend keresési funkcióját. Különböző kritériumok alapján (pl. indulási idő, érkezési idő, állomás neve) kereséseket végzünk a menetrendben, és ellenőrizzük, hogy a visszaadott eredmények helyesek-e és megfelelnek a keresési feltételeknek.

**MenetrendPerzisztencia teszt:** Ez a teszt a Menetrend osztály perzisztenciáját vizsgálja. Egy menetrend objektumot kiírunk egy stringstream-be, majd ugyanebből a streamből egy másik menetrend objektumot olvasunk vissza. Ellenőrizzük, hogy a két objektum azonos-e.

A program biztos működését úgy biztosítom, hogy minden osztály minden függvényét és perzisztenciáját alaposan tesztelem, mind önállóan, mind pedig beágyazott környezetben. Az alapos tesztelés során minden lehetséges műveletet, változót és állapotot ellenőrzök, hogy az esetleges hibákat és rendellenességeket kiszűrjem. A memória szivárgás elkerülése érdekében különös figyelmet fordítok az objektumok létrehozására és felszabadítására, biztosítva, hogy minden erőforrás megfelelően kezelve legyen. A perzisztencia tesztelése során biztosítom, hogy az objektumok állapota helyesen menthető és visszaolvasható legyen, így garantálva az adat integritását.

NHF Dokumentáció

(Doxygen)

1. [**Hierarchikus mutató**](#_bookmark0) **1**
   1. [Osztályhierarchia](#_bookmark1) 1
2. [**Osztálymutató**](#_bookmark2) **3**
   1. [Osztálylista](#_bookmark3) 3
3. [**Fájlmutató**](#_bookmark4) **5**
   1. [Fájllista](#_bookmark5) 5
4. [**Osztályok dokumentációja**](#_bookmark6) **7**
   1. [Allomas osztályreferencia](#_bookmark8) 7
      1. [Részletes leírás](#_bookmark11) 8
      2. [Konstruktorok és destruktorok dokumentációja](#_bookmark12) 8
         1. [Allomas() [1/3]](#_bookmark13) 8
         2. [Allomas() [2/3]](#_bookmark14) 8
         3. [Allomas() [3/3]](#_bookmark16) 8
         4. [*∼*Allomas()](#_bookmark19) 10
      3. [Tagfüggvények dokumentációja](#_bookmark21) 10
         1. [changeErkezes()](#_bookmark22) 10
         2. [changeIndulas()](#_bookmark23) 10
         3. [changeNev()](#_bookmark26) 11
         4. [getErkezes()](#_bookmark28) 11
         5. [getErkezesOra()](#_bookmark30) 11
         6. [getErkezesPerc()](#_bookmark32) 11
         7. [getIndulas()](#_bookmark34) 11
         8. [getIndulasOra()](#_bookmark37) 12
         9. [getIndulasPerc()](#_bookmark39) 12
         10. [getNev()](#_bookmark41) 12
         11. [operator=()](#_bookmark43) 12
         12. [read()](#_bookmark46) 12
         13. [write()](#_bookmark49) 13
      4. [Adattagok dokumentációja](#_bookmark50) 13
         1. [allomas\_azonosito](#_bookmark51) 13
         2. [erkezes](#_bookmark53) 13
         3. [indulas](#_bookmark55) 13
         4. [nev](#_bookmark57) 13
   2. [FelarasJegy osztályreferencia](#_bookmark58) 14
      1. [Részletes leírás](#_bookmark62) 15
      2. [Konstruktorok és destruktorok dokumentációja](#_bookmark63) 15
         1. [FelarasJegy() [1/3]](#_bookmark64) 15
         2. [FelarasJegy() [2/3]](#_bookmark65) 15
         3. [FelarasJegy() [3/3]](#_bookmark68) 16
         4. [*∼*FelarasJegy()](#_bookmark70) 16

|  |  |  |
| --- | --- | --- |
|  | [4.2.3. Tagfüggvények dokumentációja](#_bookmark72) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 16 |
| [4.2.3.1. clone()](#_bookmark73) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 16 |
| [4.2.3.2. kiir()](#_bookmark75) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 17 |
| [4.2.3.3. operator=()](#_bookmark77) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 17 |
| [4.2.3.4. read()](#_bookmark79) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 17 |
| [4.2.3.5. write()](#_bookmark82) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 17 |
| [4.2.4. Adattagok dokumentációja](#_bookmark84) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 18 |
| [4.2.4.1. felar](#_bookmark85) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 18 |
| [4.2.4.2. tipus](#_bookmark87) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 18 |
| [4.3.](#_bookmark88) | [Ido osztályreferencia](#_bookmark88) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 18 |
|  | [4.3.1. Részletes leírás](#_bookmark91) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 19 |
|  | [4.3.2. Konstruktorok és destruktorok dokumentációja](#_bookmark92) . . . . . . . . . . . . . . . . . . . . . . . . | 19 |
|  | [4.3.2.1. Ido() [1/3]](#_bookmark93) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 19 |
|  | [4.3.2.2. Ido() [2/3]](#_bookmark94) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 19 |
|  | [4.3.2.3. Ido() [3/3]](#_bookmark95) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 19 |
|  | [4.3.2.4. *∼*Ido()](#_bookmark99) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 20 |
|  | [4.3.3. Tagfüggvények dokumentációja](#_bookmark101) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 20 |
|  | [4.3.3.1. addOra()](#_bookmark102) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 20 |
|  | [4.3.3.2. addPerc()](#_bookmark103) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 20 |
|  | [4.3.3.3. getOra()](#_bookmark105) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 20 |
|  | [4.3.3.4. getPerc()](#_bookmark107) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 20 |
|  | [4.3.3.5. kiir()](#_bookmark110) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 21 |
|  | [4.3.3.6. operator"!=()](#_bookmark112) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 21 |
|  | [4.3.3.7. operator==()](#_bookmark114) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 21 |
|  | [4.3.3.8. read()](#_bookmark116) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 21 |
|  | [4.3.3.9. setIdo()](#_bookmark120) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 22 |
|  | [4.3.3.10. setOra()](#_bookmark121) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 22 |
|  | [4.3.3.11. setPerc()](#_bookmark123) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 22 |
|  | [4.3.3.12. write()](#_bookmark125) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 22 |
|  | [4.3.4. Adattagok dokumentációja](#_bookmark128) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 24 |
|  | [4.3.4.1. ora](#_bookmark129) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 24 |
|  | [4.3.4.2. perc](#_bookmark131) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 24 |
| [4.4.](#_bookmark132) | [Jegy osztályreferencia](#_bookmark132) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 24 |
|  | [4.4.1. Részletes leírás](#_bookmark134) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 25 |
|  | [4.4.2. Konstruktorok és destruktorok dokumentációja](#_bookmark135) . . . . . . . . . . . . . . . . . . . . . . . . | 26 |
|  | [4.4.2.1. Jegy() [1/4]](#_bookmark137) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 26 |
|  | [4.4.2.2. Jegy() [2/4]](#_bookmark138) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 26 |
|  | [4.4.2.3. Jegy() [3/4]](#_bookmark140) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 26 |
|  | [4.4.2.4. Jegy() [4/4]](#_bookmark143) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 27 |
|  | [4.4.2.5. *∼*Jegy()](#_bookmark145) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 27 |
|  | [4.4.3. Tagfüggvények dokumentációja](#_bookmark147) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 27 |
|  | [4.4.3.1. clone()](#_bookmark148) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 27 |

|  |  |  |  |
| --- | --- | --- | --- |
|  | [4.4.3.2. getAr()](#_bookmark150) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 27 |
| [4.4.3.3. kiir()](#_bookmark152) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 27 |
| [4.4.3.4. operator"!=()](#_bookmark154) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 28 |
| [4.4.3.5. operator=()](#_bookmark156) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 28 |
| [4.4.3.6. operator==()](#_bookmark158) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 28 |
| [4.4.3.7. read()](#_bookmark161) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 29 |
| [4.4.3.8. write()](#_bookmark164) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 29 |
| [4.4.4. Adattagok dokumentációja](#_bookmark165) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 29 |
| [4.4.4.1. ar](#_bookmark166) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 29 |
| [4.4.4.2. cel\_ido](#_bookmark168) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 29 |
| [4.4.4.3. cel\_nev](#_bookmark170) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 30 |
| [4.4.4.4. helyszam](#_bookmark173) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 30 |
| [4.4.4.5. indulo\_ido](#_bookmark175) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 30 |
| [4.4.4.6. indulo\_nev](#_bookmark177) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 30 |
| [4.4.4.7. jegy\_azonosito](#_bookmark179) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 30 |
| [4.4.4.8. kocsi\_szam](#_bookmark181) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 30 |
| [4.4.4.9. vonat\_szam](#_bookmark183) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 30 |
| [4.5.](#_bookmark184) | [KedvezmenyesJegy osztályreferencia](#_bookmark184) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 31 |
|  | [4.5.1. Részletes leírás](#_bookmark188) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 32 |
|  | [4.5.2. Konstruktorok és destruktorok dokumentációja](#_bookmark189) . . . . . . . . . . . . . . . . . . . . . . . . | | 32 |
|  | [4.5.2.1. KedvezmenyesJegy() [1/3]](#_bookmark190) . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 32 |
|  | [4.5.2.2. KedvezmenyesJegy() [2/3]](#_bookmark191) . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 32 |
|  | [4.5.2.3. KedvezmenyesJegy() [3/3]](#_bookmark194) . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 33 |
|  | [4.5.2.4. *∼*KedvezmenyesJegy()](#_bookmark196) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 33 |
|  | [4.5.3. Tagfüggvények dokumentációja](#_bookmark198) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 33 |
|  | [4.5.3.1. clone()](#_bookmark199) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 33 |
|  | [4.5.3.2. kiir()](#_bookmark201) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 34 |
|  | [4.5.3.3. operator=()](#_bookmark203) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 34 |
|  | [4.5.3.4. read()](#_bookmark205) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 34 |
|  | [4.5.3.5. write()](#_bookmark208) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 34 |
| [4.5.4.](#_bookmark210) | | [Adattagok dokumentációja](#_bookmark210) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 35 |
|  | | [4.5.4.1. kedvezmeny](#_bookmark211) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 35 |
|  | | [4.5.4.2. tipus](#_bookmark213) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 35 |
| [4.6. Kocsi](#_bookmark214) | | [osztályreferencia](#_bookmark214) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 35 |
| [4.6.1.](#_bookmark217) | | [Részletes leírás](#_bookmark217) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 36 |
| [4.6.2.](#_bookmark218) | | [Konstruktorok és destruktorok dokumentációja](#_bookmark218) . . . . . . . . . . . . . . . . . . . . . . . . | 36 |
|  | | [4.6.2.1. Kocsi() [1/3]](#_bookmark219) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 36 |
|  | | [4.6.2.2. Kocsi() [2/3]](#_bookmark220) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 36 |
|  | | [4.6.2.3. Kocsi() [3/3]](#_bookmark222) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 36 |
|  | | [4.6.2.4. *∼*Kocsi()](#_bookmark224) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 37 |
| [4.6.3.](#_bookmark226) | | [Tagfüggvények dokumentációja](#_bookmark226) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 37 |
|  | | [4.6.3.1. getAzonosito()](#_bookmark227) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 37 |

|  |  |  |
| --- | --- | --- |
|  | [4.6.3.2. getSzekek\_szama()](#_bookmark228) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 37 |
| [4.6.3.3. read()](#_bookmark230) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 37 |
| [4.6.3.4. reserveHely()](#_bookmark233) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 37 |
| [4.6.3.5. tele()](#_bookmark235) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 38 |
| [4.6.3.6. write()](#_bookmark237) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 38 |
| [4.6.4. Adattagok dokumentációja](#_bookmark239) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 38 |
| [4.6.4.1. foglalt\_szekek\_szama](#_bookmark240) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 38 |
| [4.6.4.2. kocsi\_azonosito](#_bookmark242) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 38 |
| [4.6.4.3. szekek\_szama](#_bookmark244) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 38 |
| [4.7.](#_bookmark245) | [Menetrend osztályreferencia](#_bookmark245) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 39 |
|  | [4.7.1. Részletes leírás](#_bookmark248) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 39 |
|  | [4.7.2. Konstruktorok és destruktorok dokumentációja](#_bookmark249) . . . . . . . . . . . . . . . . . . . . . . . . | 40 |
|  | [4.7.2.1. Menetrend() [1/4]](#_bookmark251) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 40 |
|  | [4.7.2.2. Menetrend() [2/4]](#_bookmark252) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 40 |
|  | [4.7.2.3. Menetrend() [3/4]](#_bookmark253) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 40 |
|  | [4.7.2.4. Menetrend() [4/4]](#_bookmark255) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 40 |
|  | [4.7.2.5. *∼*Menetrend()](#_bookmark259) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 41 |
|  | [4.7.3. Tagfüggvények dokumentációja](#_bookmark261) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 41 |
|  | [4.7.3.1. addVonat()](#_bookmark262) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 41 |
|  | [4.7.3.2. changeVonat()](#_bookmark263) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 41 |
|  | [4.7.3.3. clear()](#_bookmark265) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 41 |
|  | [4.7.3.4. createJegy()](#_bookmark268) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 41 |
|  | [4.7.3.5. getVonat()](#_bookmark270) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 42 |
|  | [4.7.3.6. getVonatokSzama()](#_bookmark272) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 42 |
|  | [4.7.3.7. kiir()](#_bookmark274) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 42 |
|  | [4.7.3.8. read()](#_bookmark277) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 43 |
|  | [4.7.3.9. removeVonat()](#_bookmark280) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 43 |
|  | [4.7.3.10. write()](#_bookmark281) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 43 |
|  | [4.7.4. Adattagok dokumentációja](#_bookmark283) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 44 |
|  | [4.7.4.1. vonatok](#_bookmark285) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 44 |
|  | [4.7.4.2. vonatok\_szama](#_bookmark287) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 44 |
| [4.8.](#_bookmark288) | [Serializable osztályreferencia](#_bookmark288) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 44 |
|  | [4.8.1. Részletes leírás](#_bookmark290) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 44 |
|  | [4.8.2. Tagfüggvények dokumentációja](#_bookmark291) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 44 |
|  | [4.8.2.1. read()](#_bookmark292) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 44 |
|  | [4.8.2.2. write()](#_bookmark295) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 45 |
| [4.9.](#_bookmark296) | [Utvonal osztályreferencia](#_bookmark296) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 45 |
|  | [4.9.1. Részletes leírás](#_bookmark299) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 46 |
|  | [4.9.2. Konstruktorok és destruktorok dokumentációja](#_bookmark300) . . . . . . . . . . . . . . . . . . . . . . . . | 46 |
|  | [4.9.2.1. Utvonal() [1/2]](#_bookmark301) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 46 |
|  | [4.9.2.2. Utvonal() [2/2]](#_bookmark302) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 46 |
|  | [4.9.2.3. *∼*Utvonal()](#_bookmark305) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 47 |

* + 1. [Tagfüggvények dokumentációja](#_bookmark307) 47
       1. [addAllomas()](#_bookmark308) 47
       2. [changeAllomas()](#_bookmark309) 47
       3. [createAllomas()](#_bookmark311) 47
       4. [getAllomas()](#_bookmark313) 48
       5. [getAllomasokSzama()](#_bookmark315) 48
       6. [kiir()](#_bookmark318) 48
       7. [operator=()](#_bookmark319) 48
       8. [read()](#_bookmark321) 48
       9. [removeAllomas()](#_bookmark325) 49
       10. [write()](#_bookmark326) 49
    2. [Adattagok dokumentációja](#_bookmark328) 49
       1. [allomasok](#_bookmark329) 49
       2. [allomasok\_szama](#_bookmark331) 49
       3. [utvonal\_azonosito](#_bookmark333) 50
  1. [Vonat osztályreferencia](#_bookmark335) 50
     1. [Részletes leírás](#_bookmark338) 51
     2. [Konstruktorok és destruktorok dokumentációja](#_bookmark339) 51

|  |  |  |  |
| --- | --- | --- | --- |
| [4.10.2.1.](#_bookmark340) | [Vonat() [1/3]](#_bookmark340) | . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 51 |
| [4.10.2.2.](#_bookmark341) | [Vonat() [2/3]](#_bookmark341) | . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 51 |
| [4.10.2.3.](#_bookmark343) | [Vonat() [3/3]](#_bookmark343) | . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 51 |
| [4.10.2.4.](#_bookmark346) | [*∼*Vonat()](#_bookmark346) . . . | . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 52 |

* + 1. [Tagfüggvények dokumentációja](#_bookmark348) 52
       1. [addJegy()](#_bookmark349) 52
       2. [addKocsi()](#_bookmark350) 52
       3. [createJegy()](#_bookmark353) 53
       4. [createKocsi()](#_bookmark355) 53
       5. [findAllomas()](#_bookmark357) 53
       6. [getAzonosito()](#_bookmark360) 54
       7. [getJegy()](#_bookmark362) 54
       8. [getUtvonal()](#_bookmark364) 54
       9. [indulasiIdoKulonbseg()](#_bookmark366) 54
       10. [operator=()](#_bookmark369) 55
       11. [read()](#_bookmark371) 55
       12. [routeExists()](#_bookmark374) 55
       13. [setAzonosito()](#_bookmark376) 56
       14. [setUtvonal()](#_bookmark378) 56
       15. [write()](#_bookmark380) 56
    2. [Adattagok dokumentációja](#_bookmark382) 57
       1. [jegyek](#_bookmark384) 57
       2. [jegyek\_szama](#_bookmark386) 57
       3. [kocsik](#_bookmark388) 57
       4. [kocsik\_szama](#_bookmark390) 57
       5. [utvonal](#_bookmark392) 57
       6. [vonat\_azonosito](#_bookmark394) 57

1. [Fájlok dokumentációja](#_bookmark395) 59
   1. [allomas.cpp fájlreferencia](#_bookmark397) 59
   2. [allomas.h fájlreferencia](#_bookmark399) 59
   3. [allomas.h](#_bookmark401) 59
   4. [ido.cpp fájlreferencia](#_bookmark403) 60
   5. [ido.h fájlreferencia](#_bookmark405) 60
   6. [ido.h](#_bookmark408) 61
   7. [jegy.cpp fájlreferencia](#_bookmark409) 61
   8. [jegy.h fájlreferencia](#_bookmark411) 61
   9. [jegy.h](#_bookmark413) 62
   10. [kocsi.cpp fájlreferencia](#_bookmark415) 63
   11. [kocsi.h fájlreferencia](#_bookmark417) 63
   12. [kocsi.h](#_bookmark419) 64
   13. [main.cpp fájlreferencia](#_bookmark421) 64
       1. [Függvények dokumentációja](#_bookmark423) 64
          1. [main()](#_bookmark424) 64
   14. [menetrend.cpp fájlreferencia](#_bookmark426) 65
   15. [menetrend.h fájlreferencia](#_bookmark428) 65
   16. [menetrend.h](#_bookmark430) 65
   17. [serializable.h fájlreferencia](#_bookmark431) 66
   18. [serializable.h](#_bookmark434) 66
   19. [test.cpp fájlreferencia](#_bookmark435) 66
       1. [Függvények dokumentációja](#_bookmark437) 66
          1. [test()](#_bookmark438) 66
   20. [test.h fájlreferencia](#_bookmark439) 73
       1. [Függvények dokumentációja](#_bookmark442) 73
          1. [test()](#_bookmark443) 73
   21. [test.h](#_bookmark444) 80
   22. [utvonal.cpp fájlreferencia](#_bookmark446) 80
   23. [utvonal.h fájlreferencia](#_bookmark448) 80
   24. [utvonal.h](#_bookmark450) 80
   25. [vonat.cpp fájlreferencia](#_bookmark452) 81
   26. [vonat.h fájlreferencia](#_bookmark454) 81
   27. [vonat.h](#_bookmark456) 81

[Tárgymutató](#_bookmark457) 83

**Hierarchikus mutató**

## Osztályhierarchia

Majdnem (de nem teljesen) betu˝ rendbe szedett leszármazási lista:

|  |  |
| --- | --- |
| Serializable . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | [44](#_bookmark288) |
| Allomas . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | [7](#_bookmark8) |
| Ido . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | [18](#_bookmark88) |
| Jegy . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | [24](#_bookmark132) |
| FelarasJegy . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | [14](#_bookmark58) |
| KedvezmenyesJegy . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | [31](#_bookmark184) |
| Kocsi . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | [35](#_bookmark214) |
| Menetrend . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | [39](#_bookmark245) |
| Utvonal . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | [45](#_bookmark296) |
| Vonat . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | [50](#_bookmark335) |

**Osztálymutató**

## Osztálylista

Az összes osztály, struktúra, unió és interfész listája rövid leírásokkal:

[Allomas](#_bookmark9)

Időtárolásához [7](#_bookmark8)

[FelarasJegy](#_bookmark60) [14](#_bookmark58)

[Ido](#_bookmark89)

[Jegy](#_bookmark133)

Szabványos bemenet és kimenet kezeléséhez [18](#_bookmark88)

Szabványos bemenet és kimenet kezeléséhez [24](#_bookmark132)

[KedvezmenyesJegy](#_bookmark186) [31](#_bookmark184)

[Kocsi](#_bookmark215)

Szabványos bemenet és kimenet kezeléséhez [35](#_bookmark214)

[Menetrend](#_bookmark247)

Szabványos bemenet és kimenet kezeléséhez [39](#_bookmark245)

[Serializable](#_bookmark289)

Ostream és istream használatához [44](#_bookmark288)

[Utvonal](#_bookmark297)

Interfész az objektumok sorosításához és deszerializálásához [45](#_bookmark296)

[Vonat](#_bookmark336) [50](#_bookmark335)

**Fájlmutató**

## Fájllista

Az összes fájl listája rövid leírásokkal:

[allomas.cpp](#_bookmark398) [59](#_bookmark397)

[allomas.h](#_bookmark400) [59](#_bookmark399)

[ido.cpp](#_bookmark404) [60](#_bookmark403)

[ido.h](#_bookmark406) [60](#_bookmark405)

[jegy.cpp](#_bookmark410) [61](#_bookmark409)

[jegy.h](#_bookmark412) [61](#_bookmark411)

[kocsi.cpp](#_bookmark416) [63](#_bookmark415)

[kocsi.h](#_bookmark418) [63](#_bookmark417)

[main.cpp](#_bookmark422) [64](#_bookmark421)

[menetrend.cpp](#_bookmark427) [65](#_bookmark426)

[menetrend.h](#_bookmark429) [65](#_bookmark428)

[serializable.h](#_bookmark433) [66](#_bookmark431)

[test.cpp](#_bookmark436) [66](#_bookmark435)

[test.h](#_bookmark441) [73](#_bookmark439)

[utvonal.cpp](#_bookmark447) [80](#_bookmark446)

[utvonal.h](#_bookmark449) [80](#_bookmark448)

[vonat.cpp](#_bookmark453) [81](#_bookmark452)

[vonat.h](#_bookmark455) [81](#_bookmark454)

**Osztályok dokumentációja**

## Allomas osztályreferencia

Időtárolásához.

#include *<*allomas.h*>*

Az Allomas osztály származási diagramja:

|  |  |
| --- | --- |
| Serializable | |
|  |  |
| Allomas | |

**Publikus tagfüggvények**

* [Allomas](#_bookmark13) ()

*Az érkezési időpont tárolója.*

* [Allomas](#_bookmark15) (const [Allomas](#_bookmark9) &other)
* [Allomas](#_bookmark9) & [operator=](#_bookmark44) (const [Allomas](#_bookmark9) &other)

*Értékadó operátor.*

* [Allomas](#_bookmark17) (int azonosito, const std::string &[nev](#_bookmark56), int erkezes\_ora, int erkezes\_perc, int indulas\_ora, int indulas*←›*

\_perc)

* void [changeErkezes](#_bookmark22) (int ora, int perc)
* void [changeIndulas](#_bookmark24) (int ora, int perc)
* void [changeNev](#_bookmark25) (const std::string &other)
* int [getIndulasOra](#_bookmark36) ()
* int [getIndulasPerc](#_bookmark40) ()
* int [getErkezesOra](#_bookmark31) ()
* int [getErkezesPerc](#_bookmark33) ()
* [Ido](#_bookmark89) & [getIndulas](#_bookmark35) ()
* [Ido](#_bookmark89) & [getErkezes](#_bookmark29) ()
* std::string & [getNev](#_bookmark42) ()
* void [write](#_bookmark47) (std::ostream &os) const
* void [read](#_bookmark45) (std::istream &is)
* [*∼*Allomas](#_bookmark20) ()

*Az osztály destruktora.*

**Privát attribútumok**

* int [allomas\_azonosito](#_bookmark51)
* std::string [nev](#_bookmark56)

*Az állomás azonosítója.*

* [Ido](#_bookmark89) [indulas](#_bookmark54)

*Az állomás neve.*

* [Ido](#_bookmark89) [erkezes](#_bookmark52)

*Az indulási időpont tárolója.*

### Részletes leírás

Időtárolásához.

Szabványos bemenet és kimenet kezeléséhez. Memória helyes használatának ellőrzéséhez. Interfész az objektu- mok sorosításához és deszerializálásához. Az [Allomas](#_bookmark9) osztály definíciója.

### Konstruktorok és destruktorok dokumentációja

* + - 1. **Allomas() [1/3]**

Allomas::Allomas ( )

Az érkezési időpont tárolója.

Az osztály alapértelmezett konstruktora.

* + - 1. **Allomas() [2/3]**

Allomas::Allomas (

const [Allomas](#_bookmark9) & *other* )

Másoló konstruktor.

**Paraméterek**

- Másolni kívánt [Allomas](#_bookmark9) objektum.

*other*

* + - 1. **Allomas() [3/3]**

Allomas::Allomas (

int *azonosito,*

const std::string & *nev,*

int *erkezes\_ora,* int *erkezes\_perc,* int *indulas\_ora,* int *indulas\_perc* )

Egyedi konstruktor.

**Paraméterek**

|  |  |
| --- | --- |
| *azonosito* | - Az állomás azonosítója. |
| *nev* | - Az állomás neve. |
| *erkezes\_ora* | - Az érkezés órája. |
| *erkezes\_perc* | - Az érkezés perce. |
| *indulas\_ora* | - Az indulás órája. |
| *indulas\_perc* | - Az indulás perce. |

#### *∼*Allomas()

Allomas::*∼*Allomas ( )

Az osztály destruktora.

### Tagfüggvények dokumentációja

#### changeErkezes()

void Allomas::changeErkezes (

int *ora,*

int *perc* )

Az érkezés idejének módosítása.

**Paraméterek**

|  |  |
| --- | --- |
| *ora* | - Az új érkezés órája. |
| *perc* | - Az új érkezés perce. |

#### changeIndulas()

void Allomas::changeIndulas (

int *ora,*

int *perc* )

Az indulás idejének módosítása.

**Paraméterek**

|  |  |
| --- | --- |
| *ora* | - Az új indulás órája. |
| *perc* | - Az új indulás perce. |

#### changeNev()

void Allomas::changeNev (

const std::string & *other* )

Az állomás nevének módosítása.

**Paraméterek**

- Az új név.

*other*

#### getErkezes()

[Ido](#_bookmark89) & Allomas::getErkezes ( )

Az érkezési időreferenciájának lekérdezése.

**Visszatérési érték**

Az érkezés idejének referenciája.

#### getErkezesOra()

int Allomas::getErkezesOra ( )

Az érkezés órájának lekérdezése.

**Visszatérési érték**

Az érkezés órája.

#### getErkezesPerc()

int Allomas::getErkezesPerc ( )

Az érkezés percének lekérdezése.

**Visszatérési érték**

Az érkezés perce.

#### getIndulas()

[Ido](#_bookmark89) & Allomas::getIndulas ( )

Az indulási időreferenciájának lekérdezése.

**Visszatérési érték**

Az indulás idejének referenciája.

#### getIndulasOra()

int Allomas::getIndulasOra ( )

Az indulás órájának lekérdezése.

**Visszatérési érték**

Az indulás órája.

#### getIndulasPerc()

int Allomas::getIndulasPerc ( )

Az indulás percének lekérdezése.

**Visszatérési érték**

Az indulás perce.

#### getNev()

std::string & Allomas::getNev ( )

Az állomás névének lekérdezése.

**Visszatérési érték**

Az állomás neve.

#### operator=()

[Allomas](#_bookmark9) & Allomas::operator= (

const [Allomas](#_bookmark9) & *other* )

Értékadó operátor.

#### read()

void Allomas::read (

std::istream & *is* ) [virtual]

[Serializable](#_bookmark289) interfész implementációja: objektum olvasása adatfolyamból.

**Paraméterek**

- A beolvasásra használt bemeneti adatfolyam.

*is*

Megvalósítja a következőket: [Serializable](#_bookmark292).

#### write()

void Allomas::write (

std::ostream & *os* ) const [virtual]

[Serializable](#_bookmark289) interfész implementációja: objektum írása adatfolyamba.

**Paraméterek**

- Az írásra használt kimeneti adatfolyam.

*os*

Megvalósítja a következőket: [Serializable](#_bookmark294).

### Adattagok dokumentációja

#### allomas\_azonosito

int Allomas::allomas\_azonosito [private]

#### erkezes

[Ido](#_bookmark89) Allomas::erkezes [private]

Az indulási időpont tárolója.

#### indulas

[Ido](#_bookmark89) Allomas::indulas [private]

Az állomás neve.

#### nev

std::string Allomas::nev [private]

Az állomás azonosítója.

Ez a dokumentáció az osztályról a következőfájlok alapján készült:

* + - * + [allomas.h](#_bookmark400)
        + [allomas.cpp](#_bookmark398)

## FelarasJegy osztályreferencia

#include *<*jegy.h*>*

A FelarasJegy osztály származási diagramja:

Serializable

|  |  |
| --- | --- |
| Jegy | |
|  |  |
| FelarasJegy | |

**Publikus tagfüggvények**

* [FelarasJegy](#_bookmark64) ()

*A felár értéket.*

* [FelarasJegy](#_bookmark66) (const [FelarasJegy](#_bookmark60) &other)
* [FelarasJegy](#_bookmark69) (int azonosito, int hely, int kocsi, int vonat, const std::string &indulo, [Ido](#_bookmark89) indulo\_idopont, const std::string &cel, [Ido](#_bookmark89) cel\_idopont, double [felar](#_bookmark85), std::string [tipus](#_bookmark86))
* [FelarasJegy](#_bookmark60) & [operator=](#_bookmark78) (const [FelarasJegy](#_bookmark60) &other)
* void [kiir](#_bookmark74) (std::ostream &os) const
* void [write](#_bookmark81) (std::ostream &os) const
* void [read](#_bookmark80) (std::istream &is)
* [Jegy](#_bookmark133) *∗* [clone](#_bookmark73) () const
* [*∼*FelarasJegy](#_bookmark71) ()

*Az osztály destruktora.*

**Publikus tagfüggvények a(z)** [**Jegy**](#_bookmark133) **osztályból származnak**

* [Jegy](#_bookmark137) ()

*A jegy árát számíto függvény.*

* [Jegy](#_bookmark139) (int azonosito, int hely, int kocsi, int vonat, const std::string &indulo, [Ido](#_bookmark89) indulo\_idopont, const std::string &cel, [Ido](#_bookmark89) cel\_idopont)
* [Jegy](#_bookmark141) (int azonosito, int hely, int kocsi, int vonat, const std::string &indulo, [Ido](#_bookmark89) indulo\_idopont, const std::string &cel, [Ido](#_bookmark89) cel\_idopont, double pred)
* [Jegy](#_bookmark144) (const [Jegy](#_bookmark133) &other)
* [Jegy](#_bookmark133) & [operator=](#_bookmark157) (const [Jegy](#_bookmark133) &other)
* bool [operator==](#_bookmark159) (const [Jegy](#_bookmark133) &other) const
* bool [operator!=](#_bookmark155) (const [Jegy](#_bookmark133) &other) const
* void [kiir](#_bookmark151) (std::ostream &os) const
* void [write](#_bookmark163) (std::ostream &os) const
* void [read](#_bookmark162) (std::istream &is)
* virtual [*∼*Jegy](#_bookmark146) ()

*Az osztály virtuális destruktora.*

**Privát attribútumok**

* std::string [tipus](#_bookmark86)
* double [felar](#_bookmark85)

*A féláras típus neve.*

**További örökölt tagok**

**Védett tagfüggvények a(z)** [**Jegy**](#_bookmark133) **osztályból származnak**

* virtual int [getAr](#_bookmark149) (double pred)

*A jegy ára.*

**Védett attribútumok a(z)** [**Jegy**](#_bookmark133) **osztályból származnak**

* int [jegy\_azonosito](#_bookmark178)
* int [helyszam](#_bookmark172)

*A jegy azonosítója.*

* int [kocsi\_szam](#_bookmark180)

*A jegyhez tartozó helyszám.*

* int [vonat\_szam](#_bookmark182)

*A kocsi azonositója, ahol a hely található.*

* std::string [indulo\_nev](#_bookmark176)

*A vonat azonosítója, amelyre a jegy szól.*

* [Ido](#_bookmark89) [indulo\_ido](#_bookmark174)

*Az indulási állomas neve.*

* std::string [cel\_nev](#_bookmark169)

*Az indulási időpont.*

* [Ido](#_bookmark89) [cel\_ido](#_bookmark167)

*A célállomás neve.*

* int [ar](#_bookmark166)

*A célállomás időpontja.*

### Részletes leírás

A FélárasJegy osztály a [Jegy](#_bookmark133) osztály leszármazottja, és reprezentálja a féláras jegyeket. Tartalmazza a típust és a felárat.

### Konstruktorok és destruktorok dokumentációja

* + - 1. **FelarasJegy() [1/3]**

FelarasJegy::FelarasJegy ( )

A felár értéket.

Az osztály alapértelmezett konstruktora.

* + - 1. **FelarasJegy() [2/3]**

FelarasJegy::FelarasJegy (

const [FelarasJegy](#_bookmark60) & *other* )

Másoló konstruktor.

**Paraméterek**

- másolni kívánt FélárasJegy objektum

*other*

* + - 1. **FelarasJegy() [3/3]**

FelarasJegy::FelarasJegy (

int *azonosito,*

int *hely,* int *kocsi,* int *vonat,*

const std::string & *indulo,*

[Ido](#_bookmark89) *indulo\_idopont,* const std::string & *cel,* [Ido](#_bookmark89) *cel\_idopont,*

double *felar = 0.6,*

std::string *tipus = "felaras jegy"* )

Paraméteres konstruktor.

**Paraméterek**

|  |  |
| --- | --- |
| *azonosito* | - a jegy azonosítója |
| *hely* | - a helyszám |
| *kocsi* | - a kocsi azonosítója |
| *vonat* | - a vonat azonosítója |
| *indulo* | - az indulási állomás |
| *cel* | - a célállomás |
| *felar* | - a felár értéket |
| *tipus\_nev* | - a féláras típus neve |

#### *∼*FelarasJegy()

FelarasJegy::*∼*FelarasJegy ( )

Az osztály destruktora.

* + 1. **Tagfüggvények dokumentációja**
       1. **clone()**

[Jegy](#_bookmark133) *∗* FelarasJegy::clone ( ) const [virtual]

Létrehoz egy új [Jegy](#_bookmark133) objektum másolatot az aktuális objektumról.

**Visszatérési érték**

[Jegy](#_bookmark133) objektumra mutató pointer, amely az új másolatot tartalmazza.

Újraimplementált ősök: [Jegy](#_bookmark148).

* + - 1. **kiir()**

void FelarasJegy::kiir (

std::ostream & *os* ) const

Objektum kiírása adatfolyamba esztétikus formában.

**Paraméterek**

- Az írásra használt kimeneti adatfolyam.

*os*

* + - 1. **operator=()**

[FelarasJegy](#_bookmark60) & FelarasJegy::operator= (

const [FelarasJegy](#_bookmark60) & *other* )

Értékadó operátor.

**Paraméterek**

- másolni kívánt FélárasJegy objektum

*other*

**Visszatérési érték**

A másolt FélárasJegy objektum referenciája.

* + - 1. **read()**

void FelarasJegy::read (

std::istream & *is* ) [virtual]

[Serializable](#_bookmark289) interfész implementációja: objektum olvasása adatfolyamból.

**Paraméterek**

- A beolvasásra használt bemeneti adatfolyam.

*is*

Megvalósítja a következőket: [Serializable](#_bookmark292).

* + - 1. **write()**

void FelarasJegy::write (

std::ostream & *os* ) const [virtual]

[Serializable](#_bookmark289) interfész implementációja: objektum írása adatfolyamba.

**Paraméterek**

- Az írásra használt kimeneti adatfolyam.

*os*

Megvalósítja a következőket: [Serializable](#_bookmark294).

* + 1. **Adattagok dokumentációja**
       1. **felar**

double FelarasJegy::felar [private]

A féláras típus neve.

* + - 1. **tipus**

std::string FelarasJegy::tipus [private]

Ez a dokumentáció az osztályról a következőfájlok alapján készült:

* + - * + [jegy.h](#_bookmark412)
        + [jegy.cpp](#_bookmark410)
  1. **Ido osztályreferencia**

Szabványos bemenet és kimenet kezeléséhez.

#include *<*ido.h*>*

Az Ido osztály származási diagramja:

|  |  |
| --- | --- |
| Serializable | |
|  |  |
| Ido | |

**Publikus tagfüggvények**

* [Ido](#_bookmark93) ()

*Az időobjektumban tárolt perc. (0-59)*

* [Ido](#_bookmark98) (int o, int p)
* [Ido](#_bookmark96) (const [Ido](#_bookmark89) &other)
* int [getOra](#_bookmark106) () const
* int [getPerc](#_bookmark108) () const
* void [setOra](#_bookmark122) (int o)
* void [setPerc](#_bookmark124) (int p)
* void [setIdo](#_bookmark119) (int o, int p)
* void [addPerc](#_bookmark104) (int p)
* void [addOra](#_bookmark102) (int o)
* bool [operator==](#_bookmark115) (const [Ido](#_bookmark89) &other) const
* bool [operator!=](#_bookmark113) (const [Ido](#_bookmark89) &other) const
* void [kiir](#_bookmark109) (std::ostream &os) const
* void [write](#_bookmark126) (std::ostream &os) const
* void [read](#_bookmark117) (std::istream &is)
* [*∼*Ido](#_bookmark97) ()

*Az osztály destruktora.*

**Privát attribútumok**

* int [ora](#_bookmark129)
* int [perc](#_bookmark130)

*Az időobjektumban tárolt óra. (0-23)*

* + 1. **Részletes leírás**

Szabványos bemenet és kimenet kezeléséhez.

Interfész az objektumok sorosításához és deszerializálásához. Memória helyes használatának ellőrzéséhez. Az Időosztály reprezentál egy időpontot. Tartalmazza az időpontot alkotó órát és percet.

* + 1. **Konstruktorok és destruktorok dokumentációja**
       1. **Ido() [1/3]**

Ido::Ido ( )

Az időobjektumban tárolt perc. (0-59) Az osztály alapértelmezett konstruktora.

* + - 1. **Ido() [2/3]**

Ido::Ido (

int *o,*

int *p* )

Paraméteres konstruktor.

**Paraméterek**

|  |  |
| --- | --- |
| *o* | - A tárolni kívánt óra. |
| *p* | - A tárolni kívánt perc. |

* + - 1. **Ido() [3/3]**

Ido::Ido (

const [Ido](#_bookmark89) & *other* )

Az osztály másoló konstruktora.

**Paraméterek**

- Másolni kívánt [Ido](#_bookmark89) objektum.

*other*

* + - 1. *∼***Ido()**

Ido::*∼*Ido ( )

Az osztály destruktora.

* + 1. **Tagfüggvények dokumentációja**
       1. **addOra()**

void Ido::addOra (

int *o* )

Óra hozzáadása az aktuális időhöz.

**Paraméterek**

- A hozzáadandó óra értéke.

*o*

* + - 1. **addPerc()**

void Ido::addPerc (

int *p* )

Perc hozzáadása az aktuális időhöz.

**Paraméterek**

- A hozzáadandó perc értéke.

*p*

* + - 1. **getOra()**

int Ido::getOra ( ) const

Getter függvény az óra lekérdezéséhez.

**Visszatérési érték**

Az aktuális óra értéke.

* + - 1. **getPerc()**

int Ido::getPerc ( ) const

Getter függvény a perc lekérdezéséhez.

**Visszatérési érték**

Az aktuális perc értéke.

* + - 1. **kiir()**

void Ido::kiir (

std::ostream & *os* ) const

Objektum kiírása adatfolyamba esztétikus formában.

**Paraméterek**

- Az írásra használt kimeneti adatfolyam.

*os*

* + - 1. **operator"!=()**

bool Ido::operator!= (

const [Ido](#_bookmark89) & *other* ) const

Nem egyenlőség operátor.

**Paraméterek**

- A másik idő, amivel a nem egyenlőség vizsgálat történik.

*other*

**Visszatérési érték**

igaz, ha az idők nem egyeznek meg, különben hamis.

* + - 1. **operator==()**

bool Ido::operator== (

const [Ido](#_bookmark89) & *other* ) const

Egyenlőség operátor.

**Paraméterek**

- A másik idő, amivel az egyenlőség vizsgálat történik.

*other*

**Visszatérési érték**

igaz, ha az idők megegyeznek, különben hamis.

* + - 1. **read()**

void Ido::read (

std::istream & *is* ) [virtual]

[Serializable](#_bookmark289) interfész implementációja: objektum olvasása adatfolyamból.

**Paraméterek**

- A beolvasásra használt bemeneti adatfolyam.

*is*

Megvalósítja a következőket: [Serializable](#_bookmark292).

* + - 1. **setIdo()**

void Ido::setIdo (

int *o,*

int *p* )

Időbeállítása adott óra és perc értékekkel.

**Paraméterek**

|  |  |
| --- | --- |
| *o* | - A beállítandó óra értéke. |
| *p* | - A beállítandó perc értéke. |

* + - 1. **setOra()**

void Ido::setOra (

int *o* )

Setter függvény az óra beállításához.

**Paraméterek**

- A beállítandó óra értéke.

*o*

* + - 1. **setPerc()**

void Ido::setPerc (

int *p* )

Setter függvény a perc beállításához.

**Paraméterek**

- A beállítandó perc értéke.

*p*

* + - 1. **write()**

void Ido::write (

std::ostream & *os* ) const [virtual]

[Serializable](#_bookmark289) interfész implementációja: objektum írása adatfolyamba.

**Paraméterek**

- Az írásra használt kimeneti adatfolyam.

*os*

Megvalósítja a következőket: [Serializable](#_bookmark294).

* + 1. **Adattagok dokumentációja**
       1. **ora**

int Ido::ora [private]

* + - 1. **perc**

int Ido::perc [private]

Az időobjektumban tárolt óra. (0-23)

Ez a dokumentáció az osztályról a következőfájlok alapján készült:

* + - * + [ido.h](#_bookmark406)
        + [ido.cpp](#_bookmark404)
  1. **Jegy osztályreferencia**

Szabványos bemenet és kimenet kezeléséhez.

#include *<*jegy.h*>*

A Jegy osztály származási diagramja:

Serializable

KedvezmenyesJegy

FelarasJegy

Jegy

**Publikus tagfüggvények**

* [Jegy](#_bookmark137) ()

*A jegy árát számíto függvény.*

* [Jegy](#_bookmark139) (int azonosito, int hely, int kocsi, int vonat, const std::string &indulo, [Ido](#_bookmark89) indulo\_idopont, const std::string &cel, [Ido](#_bookmark89) cel\_idopont)
* [Jegy](#_bookmark141) (int azonosito, int hely, int kocsi, int vonat, const std::string &indulo, [Ido](#_bookmark89) indulo\_idopont, const std::string &cel, [Ido](#_bookmark89) cel\_idopont, double pred)
* [Jegy](#_bookmark144) (const [Jegy](#_bookmark133) &other)
* [Jegy](#_bookmark133) & [operator=](#_bookmark157) (const [Jegy](#_bookmark133) &other)
* bool [operator==](#_bookmark159) (const [Jegy](#_bookmark133) &other) const
* bool [operator!=](#_bookmark155) (const [Jegy](#_bookmark133) &other) const
* void [kiir](#_bookmark151) (std::ostream &os) const
* void [write](#_bookmark163) (std::ostream &os) const
* void [read](#_bookmark162) (std::istream &is)
* virtual [Jegy](#_bookmark133) *∗* [clone](#_bookmark148) () const
* virtual [*∼*Jegy](#_bookmark146) ()

*Az osztály virtuális destruktora.*

**Védett tagfüggvények**

* virtual int [getAr](#_bookmark149) (double pred)

*A jegy ára.*

**Védett attribútumok**

* int [jegy\_azonosito](#_bookmark178)
* int [helyszam](#_bookmark172)

*A jegy azonosítója.*

* int [kocsi\_szam](#_bookmark180)

*A jegyhez tartozó helyszám.*

* int [vonat\_szam](#_bookmark182)

*A kocsi azonositója, ahol a hely található.*

* std::string [indulo\_nev](#_bookmark176)

*A vonat azonosítója, amelyre a jegy szól.*

* [Ido](#_bookmark89) [indulo\_ido](#_bookmark174)

*Az indulási állomas neve.*

* std::string [cel\_nev](#_bookmark169)

*Az indulási időpont.*

* [Ido](#_bookmark89) [cel\_ido](#_bookmark167)

*A célállomás neve.*

* int [ar](#_bookmark166)

*A célállomás időpontja.*

* + 1. **Részletes leírás**

Szabványos bemenet és kimenet kezeléséhez.

Interfész az objektumok sorosításához és deszerializálásához. Memória helyes használatának ellőrzéséhez. Időtárolásához. A [Jegy](#_bookmark133) osztaly reprezentalja a vonatjegyeket. Tartalmazza a jegy azonositojat, a helyszamot, a kocsi szamat, a vonat szamat, az indulo es a cel allomast, valamint az arat.

* + 1. **Konstruktorok és destruktorok dokumentációja**
       1. **Jegy() [1/4]**

Jegy::Jegy ( )

A jegy árát számíto függvény.

Az osztaly alapertelmezett konstruktora.

* + - 1. **Jegy() [2/4]**

Jegy::Jegy (

int *azonosito,*

int *hely,* int *kocsi,* int *vonat,*

const std::string & *indulo,*

[Ido](#_bookmark89) *indulo\_idopont,* const std::string & *cel,* [Ido](#_bookmark89) *cel\_idopont* )

Paraméteres konstruktor.

**Paraméterek**

|  |  |
| --- | --- |
| *azonosito* | - a jegy azonosítója |
| *hely* | - a helyszám |
| *kocsi* | - a kocsi azonosítója |
| *vonat* | - a vonat azonosítója |
| *indulo* | - az indulási állomás |
| *cel* | - a célállomás |

* + - 1. **Jegy() [3/4]**

Jegy::Jegy (

int *azonosito,*

int *hely,* int *kocsi,* int *vonat,*

const std::string & *indulo,*

[Ido](#_bookmark89) *indulo\_idopont,* const std::string & *cel,* [Ido](#_bookmark89) *cel\_idopont,*

double *pred* )

Paraméteres konstruktor (alosztályhoz).

**Paraméterek**

- a jegy azonosítója

*azonosito*

**Paraméterek**

|  |  |
| --- | --- |
| *hely* | - a helyszám |
| *kocsi* | - a kocsi azonosítója |
| *vonat* | - a vonat azonosítója |
| *indulo* | - az indulási állomás |
| *cel* | - a célállomás |
| *pred* | - az árat meghatározó kedvezmény vagy felár értéke |

* + - 1. **Jegy() [4/4]**

Jegy::Jegy (

const [Jegy](#_bookmark133) & *other* )

Az osztály másoló konstruktora.

**Paraméterek**

- Másolni kívánt [Jegy](#_bookmark133) objektum.

*other*

* + - 1. *∼***Jegy()**

virtual Jegy::*∼*Jegy ( ) [inline], [virtual]

Az osztály virtuális destruktora.

* + 1. **Tagfüggvények dokumentációja**
       1. **clone()**

[Jegy](#_bookmark133) *∗* Jegy::clone ( ) const [virtual]

Létrehoz egy új [Jegy](#_bookmark133) objektum másolatot az aktuális objektumról.

**Visszatérési érték**

[Jegy](#_bookmark133) objektumra mutató pointer, amely az új másolatot tartalmazza. Újraimplementáló leszármazottak: [KedvezmenyesJegy](#_bookmark199) és [FelarasJegy](#_bookmark73).

* + - 1. **getAr()**

int Jegy::getAr (

double *pred = 1* ) [protected], [virtual]

A jegy ára.

* + - 1. **kiir()**

void Jegy::kiir (

std::ostream & *os* ) const

Objektum kiírása adatfolyamba esztétikus formában.

**Paraméterek**

- Az írásra használt kimeneti adatfolyam.

*os*

* + - 1. **operator"!=()**

bool Jegy::operator!= (

const [Jegy](#_bookmark133) & *other* ) const

Nem egyenlőség operátor.

**Paraméterek**

- A másik jegy, amivel a nem egyenlőség vizsgálat történik.

*other*

**Visszatérési érték**

igaz, ha a jegyek nem egyeznek meg, különben hamis.

* + - 1. **operator=()**

[Jegy](#_bookmark133) & Jegy::operator= (

const [Jegy](#_bookmark133) & *other* )

Értékadó operátor.

**Paraméterek**

- Másolni kívánt [Jegy](#_bookmark133) objektum.

*other*

**Visszatérési érték**

A másolt [Jegy](#_bookmark133) objektum referenciája.

* + - 1. **operator==()**

bool Jegy::operator== (

const [Jegy](#_bookmark133) & *other* ) const

Egyenlőség operátor.

**Paraméterek**

- A másik jegy, amivel az egyenlőség vizsgálat történik.

*other*

**Visszatérési érték**

igaz, ha a jegyek megegyeznek, különben hamis.

* + - 1. **read()**

void Jegy::read (

std::istream & *is* ) [virtual]

[Serializable](#_bookmark289) interfész implementációja: objektum olvasása adatfolyamból.

**Paraméterek**

- A beolvasásra használt bemeneti adatfolyam.

*is*

Megvalósítja a következőket: [Serializable](#_bookmark292). Újraimplementáló leszármazottak: [KedvezmenyesJegy](#_bookmark206).

* + - 1. **write()**

void Jegy::write (

std::ostream & *os* ) const [virtual]

[Serializable](#_bookmark289) interfész implementációja: objektum írása adatfolyamba.

**Paraméterek**

- Az írásra használt kimeneti adatfolyam.

*os*

Megvalósítja a következőket: [Serializable](#_bookmark294). Újraimplementáló leszármazottak: [KedvezmenyesJegy](#_bookmark207).

* + 1. **Adattagok dokumentációja**
       1. **ar**

int Jegy::ar [protected]

A célállomás időpontja.

* + - 1. **cel\_ido**

[Ido](#_bookmark89) Jegy::cel\_ido [protected]

A célállomás neve.

* + - 1. **cel\_nev**

std::string Jegy::cel\_nev [protected]

Az indulási időpont.

* + - 1. **helyszam**

int Jegy::helyszam [protected]

A jegy azonosítója.

* + - 1. **indulo\_ido**

[Ido](#_bookmark89) Jegy::indulo\_ido [protected]

Az indulási állomas neve.

* + - 1. **indulo\_nev**

std::string Jegy::indulo\_nev [protected]

A vonat azonosítója, amelyre a jegy szól.

* + - 1. **jegy\_azonosito**

int Jegy::jegy\_azonosito [protected]

* + - 1. **kocsi\_szam**

int Jegy::kocsi\_szam [protected]

A jegyhez tartozó helyszám.

* + - 1. **vonat\_szam**

int Jegy::vonat\_szam [protected]

A kocsi azonositója, ahol a hely található.

Ez a dokumentáció az osztályról a következőfájlok alapján készült:

* + - * + [jegy.h](#_bookmark412)
        + [jegy.cpp](#_bookmark410)
  1. **KedvezmenyesJegy osztályreferencia**

#include *<*jegy.h*>*

A KedvezmenyesJegy osztály származási diagramja:

|  |  |
| --- | --- |
| Serializable | |
|  |  |
| Jegy | |
|  |  |
| KedvezmenyesJegy | |

**Publikus tagfüggvények**

* [KedvezmenyesJegy](#_bookmark190) ()

*A kedvezmény mértékét százalékban kifejezőérték.*

* [KedvezmenyesJegy](#_bookmark192) (const [KedvezmenyesJegy](#_bookmark186) &other)
* [KedvezmenyesJegy](#_bookmark195) (int azonosito, int hely, int kocsi, int vonat, const std::string &indulo, [Ido](#_bookmark89) indulo\_idopont, const std::string &cel, [Ido](#_bookmark89) cel\_idopont, double [kedvezmeny](#_bookmark211), std::string [tipus](#_bookmark212))
* [KedvezmenyesJegy](#_bookmark186) & [operator=](#_bookmark204) (const [KedvezmenyesJegy](#_bookmark186) &other)
* void [kiir](#_bookmark200) (std::ostream &os) const
* void [write](#_bookmark207) (std::ostream &os) const
* void [read](#_bookmark206) (std::istream &is)
* [Jegy](#_bookmark133) *∗* [clone](#_bookmark199) () const
* [*∼*KedvezmenyesJegy](#_bookmark197) ()

*Az osztály destruktora.*

**Publikus tagfüggvények a(z)** [**Jegy**](#_bookmark133) **osztályból származnak**

* [Jegy](#_bookmark137) ()

*A jegy árát számíto függvény.*

* [Jegy](#_bookmark139) (int azonosito, int hely, int kocsi, int vonat, const std::string &indulo, [Ido](#_bookmark89) indulo\_idopont, const std::string &cel, [Ido](#_bookmark89) cel\_idopont)
* [Jegy](#_bookmark141) (int azonosito, int hely, int kocsi, int vonat, const std::string &indulo, [Ido](#_bookmark89) indulo\_idopont, const std::string &cel, [Ido](#_bookmark89) cel\_idopont, double pred)
* [Jegy](#_bookmark144) (const [Jegy](#_bookmark133) &other)
* [Jegy](#_bookmark133) & [operator=](#_bookmark157) (const [Jegy](#_bookmark133) &other)
* bool [operator==](#_bookmark159) (const [Jegy](#_bookmark133) &other) const
* bool [operator!=](#_bookmark155) (const [Jegy](#_bookmark133) &other) const
* void [kiir](#_bookmark151) (std::ostream &os) const
* virtual [*∼*Jegy](#_bookmark146) ()

*Az osztály virtuális destruktora.*

**Privát attribútumok**

* std::string [tipus](#_bookmark212)
* double [kedvezmeny](#_bookmark211)

*A kedvezményes típus neve.*

**További örökölt tagok**

**Védett tagfüggvények a(z)** [**Jegy**](#_bookmark133) **osztályból származnak**

* virtual int [getAr](#_bookmark149) (double pred)

*A jegy ára.*

**Védett attribútumok a(z)** [**Jegy**](#_bookmark133) **osztályból származnak**

* int [jegy\_azonosito](#_bookmark178)
* int [helyszam](#_bookmark172)

*A jegy azonosítója.*

* int [kocsi\_szam](#_bookmark180)

*A jegyhez tartozó helyszám.*

* int [vonat\_szam](#_bookmark182)

*A kocsi azonositója, ahol a hely található.*

* std::string [indulo\_nev](#_bookmark176)

*A vonat azonosítója, amelyre a jegy szól.*

* [Ido](#_bookmark89) [indulo\_ido](#_bookmark174)

*Az indulási állomas neve.*

* std::string [cel\_nev](#_bookmark169)

*Az indulási időpont.*

* [Ido](#_bookmark89) [cel\_ido](#_bookmark167)

*A célállomás neve.*

* int [ar](#_bookmark166)

*A célállomás időpontja.*

* + 1. **Részletes leírás**

A KedvezményesJegy osztály a [Jegy](#_bookmark133) osztály leszármazottja, és reprezentálja a kedvezményes jegyeket. Tartal- mazza a típust (pl. diák, nyugdíjas) és a kedvezmény mértékét.

* + 1. **Konstruktorok és destruktorok dokumentációja**
       1. **KedvezmenyesJegy() [1/3]**

KedvezmenyesJegy::KedvezmenyesJegy ( )

A kedvezmény mértékét százalékban kifejezőérték. Az osztály alapértelmezett konstruktora.

* + - 1. **KedvezmenyesJegy() [2/3]**

KedvezmenyesJegy::KedvezmenyesJegy (

const [KedvezmenyesJegy](#_bookmark186) & *other* )

Másoló konstruktor.

**Paraméterek**

- másolni kívánt [KedvezmenyesJegy](#_bookmark186) objektum

*other*

* + - 1. **KedvezmenyesJegy() [3/3]**

KedvezmenyesJegy::KedvezmenyesJegy (

int *azonosito,*

int *hely,* int *kocsi,* int *vonat,*

const std::string & *indulo,*

[Ido](#_bookmark89) *indulo\_idopont,* const std::string & *cel,* [Ido](#_bookmark89) *cel\_idopont,*

double *kedvezmeny = -0.6,*

std::string *tipus = "kedvezmenyes jegy"* )

Paraméteres konstruktor.

**Paraméterek**

|  |  |
| --- | --- |
| *azonosito* | - a jegy azonosítója |
| *hely* | - a helyszám |
| *kocsi* | - a kocsi azonosítója |
| *vonat* | - a vonat azonosítója |
| *indulo* | - az indulási állomás |
| *cel* | - a célállomás |
| *kedvezmeny* | - a kedvezmény mértékét kifejezőérték |
| *tipus\_nev* | - a kedvezményes típus neve |

* + - 1. *∼***KedvezmenyesJegy()**

KedvezmenyesJegy::*∼*KedvezmenyesJegy ( )

Az osztály destruktora.

* + 1. **Tagfüggvények dokumentációja**
       1. **clone()**

[Jegy](#_bookmark133) *∗* KedvezmenyesJegy::clone ( ) const [virtual]

Létrehoz egy új [Jegy](#_bookmark133) objektum másolatot az aktuális objektumról.

**Visszatérési érték**

[Jegy](#_bookmark133) objektumra mutató pointer, amely az új másolatot tartalmazza.

Újraimplementált ősök: [Jegy](#_bookmark148).

* + - 1. **kiir()**

void KedvezmenyesJegy::kiir (

std::ostream & *os* ) const

Objektum kiírása adatfolyamba esztétikus formában.

**Paraméterek**

- Az írásra használt kimeneti adatfolyam.

*os*

* + - 1. **operator=()**

[KedvezmenyesJegy](#_bookmark186) & KedvezmenyesJegy::operator= ( const [KedvezmenyesJegy](#_bookmark186) & *other* )

Értékadó operátor.

**Paraméterek**

- másolni kívánt [KedvezmenyesJegy](#_bookmark186) objektum

*other*

**Visszatérési érték**

A másolt [KedvezmenyesJegy](#_bookmark186) objektum referenciája.

* + - 1. **read()**

void KedvezmenyesJegy::read (

std::istream & *is* ) [virtual]

[Serializable](#_bookmark289) interfész implementációja: objektum olvasása adatfolyamból.

**Paraméterek**

- A beolvasásra használt bemeneti adatfolyam.

*is*

Újraimplementált ősök: [Jegy](#_bookmark162).

* + - 1. **write()**

void KedvezmenyesJegy::write (

std::ostream & *os* ) const [virtual]

[Serializable](#_bookmark289) interfész implementációja: objektum írása adatfolyamba.

**Paraméterek**

- Az írásra használt kimeneti adatfolyam.

*os*

Újraimplementált ősök: [Jegy](#_bookmark163).

* + 1. **Adattagok dokumentációja**
       1. **kedvezmeny**

double KedvezmenyesJegy::kedvezmeny [private]

A kedvezményes típus neve.

* + - 1. **tipus**

std::string KedvezmenyesJegy::tipus [private]

Ez a dokumentáció az osztályról a következőfájlok alapján készült:

* + - * + [jegy.h](#_bookmark412)
        + [jegy.cpp](#_bookmark410)
  1. **Kocsi osztályreferencia**

Szabványos bemenet és kimenet kezeléséhez.

#include *<*kocsi.h*>*

A Kocsi osztály származási diagramja:

|  |  |
| --- | --- |
| Serializable | |
|  |  |
| Kocsi | |

**Publikus tagfüggvények**

* [Kocsi](#_bookmark219) ()

*A kocsihoz tartozo foglalt szekek szama.*

* [Kocsi](#_bookmark221) (const [Kocsi](#_bookmark215) &other)
* [Kocsi](#_bookmark223) (int azonosito, int szekek)
* bool [tele](#_bookmark236) () const
* int [reserveHely](#_bookmark232) ()
* int [getSzekek\_szama](#_bookmark229) () const
* int [getAzonosito](#_bookmark227) () const
* void [write](#_bookmark238) (std::ostream &os) const
* void [read](#_bookmark231) (std::istream &is)
* [*∼*Kocsi](#_bookmark224) ()

*Az osztály destruktora.*

**Privát attribútumok**

* int [kocsi\_azonosito](#_bookmark241)
* int [szekek\_szama](#_bookmark243)

*A kocsi azonositoja.*

* int [foglalt\_szekek\_szama](#_bookmark240)

*A kocsihoz tartozo szekek szama.*

* + 1. **Részletes leírás**

Szabványos bemenet és kimenet kezeléséhez.

Interfész az objektumok sorosításához és deszerializálásához. Memória helyes használatának ellőrzéséhez. A [Kocsi](#_bookmark215) osztaly reprezentalja a vonatkocsikat. Tartalmazza a kocsi azonositojat, a szekek szamat, valamint a foglalt szekek szamat.

* + 1. **Konstruktorok és destruktorok dokumentációja**
       1. **Kocsi() [1/3]**

Kocsi::Kocsi ( )

A kocsihoz tartozo foglalt szekek szama. Az osztály alapértelmezett konstruktora.

* + - 1. **Kocsi() [2/3]**

Kocsi::Kocsi (

const [Kocsi](#_bookmark215) & *other* )

Az osztály másoló konstruktora.

**Paraméterek**

- Másolni kívánt [Kocsi](#_bookmark215) objektum.

*other*

* + - 1. **Kocsi() [3/3]**

Kocsi::Kocsi (

int *azonosito,*

int *szekek* )

Egyedi konstruktor.

**Paraméterek**

|  |  |  |
| --- | --- | --- |
| *azonosito* | - A kocsi azonosítója. |  |
| *szekek* | - A kocsiban található üres székek száma. |
|  |  |  |

* + - 1. *∼***Kocsi()**

Kocsi::*∼*Kocsi ( )

Az osztály destruktora.

* + 1. **Tagfüggvények dokumentációja**
       1. **getAzonosito()**

int Kocsi::getAzonosito ( ) const

Az osztály azonosítójának lekérdezése.

**Visszatérési érték**

A kocsi azonosítója.

* + - 1. **getSzekek\_szama()**

int Kocsi::getSzekek\_szama ( ) const

Szákek számának lekérdezése.

**Visszatérési érték**

A kocsihoz tartozó székek száma.

* + - 1. **read()**

void Kocsi::read (

std::istream & *is* ) [virtual]

[Serializable](#_bookmark289) interfész implementációja: objektum olvasása adatfolyamból.

**Paraméterek**

- A beolvasásra használt bemeneti adatfolyam.

*is*

Megvalósítja a következőket: [Serializable](#_bookmark292).

* + - 1. **reserveHely()**

int Kocsi::reserveHely ( )

Foglal egy helyet a kocsiban, és visszaadja a hely azonosítóját.

**Visszatérési érték**

A foglalt hely azonosítója.

* + - 1. **tele()**

bool Kocsi::tele ( ) const

Megvizsgálja, hogy a kocsiban van-e üres szék.

**Visszatérési érték**

igaz, ha a kocsi tele van, különben hamis.

* + - 1. **write()**

void Kocsi::write (

std::ostream & *os* ) const [virtual]

[Serializable](#_bookmark289) interfész implementációja: objektum írása adatfolyamba.

**Paraméterek**

- Az írásra használt kimeneti adatfolyam.

*os*

Megvalósítja a következőket: [Serializable](#_bookmark294).

* + 1. **Adattagok dokumentációja**
       1. **foglalt\_szekek\_szama**

int Kocsi::foglalt\_szekek\_szama [private]

A kocsihoz tartozo szekek szama.

* + - 1. **kocsi\_azonosito**

int Kocsi::kocsi\_azonosito [private]

* + - 1. **szekek\_szama**

int Kocsi::szekek\_szama [private]

A kocsi azonositoja.

Ez a dokumentáció az osztályról a következőfájlok alapján készült:

* + - * + [kocsi.h](#_bookmark418)
        + [kocsi.cpp](#_bookmark416)
  1. **Menetrend osztályreferencia**

Szabványos bemenet és kimenet kezeléséhez.

#include *<*menetrend.h*>*

A Menetrend osztály származási diagramja:

|  |  |
| --- | --- |
| Serializable | |
|  |  |
| Menetrend | |

**Publikus tagfüggvények**

* [Menetrend](#_bookmark251) ()

*A menetrendben szereplővonatok tömbje.*

* [Menetrend](#_bookmark257) ([Vonat](#_bookmark336) v)
* [Menetrend](#_bookmark254) ([Vonat](#_bookmark336) *∗*v, int meret)
* [Menetrend](#_bookmark256) (const [Menetrend](#_bookmark247) &m)
* void [addVonat](#_bookmark262) ([Vonat](#_bookmark336) v)
* void [changeVonat](#_bookmark264) ([Vonat](#_bookmark336) v, int index)
* [Vonat](#_bookmark336) & [getVonat](#_bookmark271) (int index) const
* int [getVonatokSzama](#_bookmark273) () const
* void [removeVonat](#_bookmark279) (int index)
* void [createJegy](#_bookmark267) (std::string indulo, std::string erkezo, int indulo\_ora, int indulo\_perc, double discountOrFee=0, const std::string &tipus="")
* void [clear](#_bookmark266) ()

*Menetrendben tárolt adatok törlése.*

* void [kiir](#_bookmark275) (std::ostream &os, std::string indulo, std::string erkezo) const
* void [write](#_bookmark282) (std::ostream &os) const
* void [read](#_bookmark278) (std::istream &is)
* [*∼*Menetrend](#_bookmark258) ()

*Az osztály destruktora.*

**Privát attribútumok**

* int [vonatok\_szama](#_bookmark286)
* [Vonat](#_bookmark336) *∗* [vonatok](#_bookmark285)

*A menetrendben szereplővonatok száma.*

* + 1. **Részletes leírás**

Szabványos bemenet és kimenet kezeléséhez.

Interfész az objektumok sorosításához és deszerializálásához. Memória helyes használatának ellőrzéséhez. A [Menetrend](#_bookmark247) osztály reprezentálja a vonatok menetrendjét. Tartalmazza a vonatok számát és a vonatokat.

* + 1. **Konstruktorok és destruktorok dokumentációja**
       1. **Menetrend() [1/4]**

Menetrend::Menetrend ( ) [inline] A menetrendben szereplővonatok tömbje. Az osztály alapértelmezett konstruktora.

* + - 1. **Menetrend() [2/4]**

Menetrend::Menetrend (

[Vonat](#_bookmark336) *v* )

Paraméteres konstruktor.

**Paraméterek**

- hozzáadni kívánt vonat

*v*

* + - 1. **Menetrend() [3/4]**

Menetrend::Menetrend (

[Vonat](#_bookmark336) *∗ v,*

int *meret* ) [inline]

Paraméteres konstruktor.

**Paraméterek**

|  |  |
| --- | --- |
| *v* | - a vonatok tömbje |
| *meret* | - a vonatok tömbjének mérete |

* + - 1. **Menetrend() [4/4]**

Menetrend::Menetrend (

const [Menetrend](#_bookmark247) & *m* )

Masoló konstruktor.

**Paraméterek**

- másolni kívánt [Menetrend](#_bookmark247) objektum

*m*

* + - 1. *∼***Menetrend()**

Menetrend::*∼*Menetrend ( )

Az osztály destruktora.

* + 1. **Tagfüggvények dokumentációja**
       1. **addVonat()**

void Menetrend::addVonat (

[Vonat](#_bookmark336) *v* )

[Vonat](#_bookmark336) hozzáadása a menetrendhez.

**Paraméterek**

- hozzáadni kívánt vonat

*v*

* + - 1. **changeVonat()**

void Menetrend::changeVonat (

[Vonat](#_bookmark336) *v,*

int *index* )

[Vonat](#_bookmark336) módosítása a menetrendben.

**Paraméterek**

|  |  |
| --- | --- |
| *v* | - a módosítandó vonat |
| *index* | - a vonat indexe a menetrendben |

* + - 1. **clear()**

void Menetrend::clear ( )

Menetrendben tárolt adatok törlése.

* + - 1. **createJegy()**

void Menetrend::createJegy (

std::string *indulo,* std::string *erkezo,* int *indulo\_ora,*

int *indulo\_perc,*

double *discountOrFee = 0,*

const std::string & *tipus = ""* )

[Jegy](#_bookmark133) létrehozása a megadott állomások között.

**Paraméterek**

|  |  |
| --- | --- |
| *indulo* | - Az induló állomás neve. |
| *erkezo* | - Az érkezőállomás neve. |
| *indulo\_ora* | - Az indulási időórája. |
| *indulo\_perc* | - Az indulási időperce. |
| *discountOrFee* | - Kedvezmény vagy felár értéke. |
| *tipus* | - A jegy típusa. |

* + - 1. **getVonat()**

[Vonat](#_bookmark336) & Menetrend::getVonat (

int *index* ) const

[Vonat](#_bookmark336) lekérdezése a megadott indexen.

**Paraméterek**

- a kívánt vonat indexe

*index*

**Visszatérési érték**

Az adott indexen található vonat

* + - 1. **getVonatokSzama()**

int Menetrend::getVonatokSzama ( ) const

Vonatok számának lekérdezése.

**Visszatérési érték**

A vonatok száma a menetrendben

* + - 1. **kiir()**

void Menetrend::kiir (

std::ostream & *os,*

std::string *indulo = "",*

std::string *erkezo = ""* ) const

Kiírja a menetrendben található vonatok útvonalát a megadott állomások között. Ha indulo és erkezo üres, akkor mindet. Ha csak indulo üres, akkor kiír minden útvonalat, amely az adott állomásba tart. Ha csak erkezo üres, akkor kiír minden útvonalat, amely az adott állomásból indul. Ha erkezo és indulo üres, akkor kiír minden útvonalat, amely a két állomás között van.

**Paraméterek**

|  |  |
| --- | --- |
| *os* | - Az írásra használt kimeneti adatfolyam. |
| *indulo* | - Az induló állomás neve. |
| *erkezo* | - Az érkezőállomás neve. |

* + - 1. **read()**

void Menetrend::read (

std::istream & *is* ) [virtual]

[Serializable](#_bookmark289) interfész implementációja: objektum olvasása adatfolyamból.

**Paraméterek**

- A beolvasásra használt bemeneti adatfolyam.

*is*

Megvalósítja a következőket: [Serializable](#_bookmark292).

* + - 1. **removeVonat()**

void Menetrend::removeVonat (

int *index* )

[Vonat](#_bookmark336) törlése a megadott indexen.

**Paraméterek**

- a törlendővonat indexe

*index*

* + - 1. **write()**

void Menetrend::write (

std::ostream & *os* ) const [virtual]

[Serializable](#_bookmark289) interfész implementációja: objektum írása adatfolyamba.

**Paraméterek**

- Az írásra használt kimeneti adatfolyam.

*os*

Megvalósítja a következőket: [Serializable](#_bookmark294).

* + 1. **Adattagok dokumentációja**
       1. **vonatok**

[Vonat](#_bookmark336)*∗* Menetrend::vonatok [private]

A menetrendben szereplővonatok száma.

* + - 1. **vonatok\_szama**

int Menetrend::vonatok\_szama [private]

Ez a dokumentáció az osztályról a következőfájlok alapján készült:

* + - * + [menetrend.h](#_bookmark429)
        + [menetrend.cpp](#_bookmark427)
  1. **Serializable osztályreferencia**

Ostream és istream használatához.

#include *<*serializable.h*>*

A Serializable osztály származási diagramja:

Jegy

FelarasJegy KedvezmenyesJegy

Vonat

Utvonal

Menetrend

Kocsi

Ido

Allomas

Serializable

**Publikus tagfüggvények**

* virtual void [write](#_bookmark294) (std::ostream &os) const =0
* virtual void [read](#_bookmark292) (std::istream &is)=0
  + 1. **Részletes leírás**

Ostream és istream használatához.

Memória helyes használatának ellőrzéséhez. Absztrakt osztály a perzisztencia megvalósításához. Tartalmazza a [write()](#_bookmark294) és [read()](#_bookmark292) virtuális függvényeket, amelyeket a leszármazott osztályoknak meg kell valósítaniuk.

* + 1. **Tagfüggvények dokumentációja**
       1. **read()**

virtual void Serializable::read (

std::istream & *is* ) [pure virtual]

Adatfolyamról olvasásért felelős függvény.

**Paraméterek**

- Az objektum olvasása során használt bemeneti adatfolyam, például egy fájl vagy a konzol bemenete.

*is*

Megvalósítják a következők: [Allomas](#_bookmark45), [Ido](#_bookmark117), [Jegy](#_bookmark162), [KedvezmenyesJegy](#_bookmark206), [FelarasJegy](#_bookmark80), [Kocsi](#_bookmark231), [Menetrend](#_bookmark278), [Utvonal](#_bookmark322) és [Vonat](#_bookmark372).

* + - 1. **write()**

virtual void Serializable::write (

std::ostream & *os* ) const [pure virtual]

Adatfolyamra írásért felelős függvény.

**Paraméterek**

- Az objektum írása során használt kimeneti adatfolyam, például egy fájl vagy a konzol kimenete.

*os*

Megvalósítják a következők: [Allomas](#_bookmark47), [Ido](#_bookmark126), [Jegy](#_bookmark163), [KedvezmenyesJegy](#_bookmark207), [FelarasJegy](#_bookmark81), [Kocsi](#_bookmark238), [Menetrend](#_bookmark282), [Utvonal](#_bookmark327) és [Vonat](#_bookmark381).

Ez a dokumentáció az osztályról a következőfájl alapján készült:

* + - * + [serializable.h](#_bookmark433)
  1. **Utvonal osztályreferencia**

Interfész az objektumok sorosításához és deszerializálásához.

#include *<*utvonal.h*>*

Az Utvonal osztály származási diagramja:

|  |  |
| --- | --- |
| Serializable | |
|  |  |
| Utvonal | |

**Publikus tagfüggvények**

* [Utvonal](#_bookmark301) ()

*Az állomások tömbje az útvonalon.*

* [Utvonal](#_bookmark303) (const [Utvonal](#_bookmark297) &other)
* void [addAllomas](#_bookmark308) ([Allomas](#_bookmark9) allomas)
* void [createAllomas](#_bookmark312) (std::string nev, int erkezes\_ora, int erkezes\_perc, int indulas\_ora, int indulas\_perc)
* [Utvonal](#_bookmark297) & [operator=](#_bookmark320) (const [Utvonal](#_bookmark297) &other)
* void [changeAllomas](#_bookmark310) (int index, [Allomas](#_bookmark9) allomas)
* void [removeAllomas](#_bookmark324) (int index)
* int [getAllomasokSzama](#_bookmark316) () const

*Az útvonalon található állomások számának lekérdezése.*

* [Allomas](#_bookmark9) & [getAllomas](#_bookmark313) (int i) const
* void [kiir](#_bookmark317) (std::ostream &os) const
* void [write](#_bookmark327) (std::ostream &os) const
* void [read](#_bookmark322) (std::istream &is)
* [*∼*Utvonal](#_bookmark304) ()

*Az osztály destruktora.*

**Privát attribútumok**

* int [utvonal\_azonosito](#_bookmark332)
* int [allomasok\_szama](#_bookmark330)

*Az útvonal azonosítója.*

* [Allomas](#_bookmark9) *∗* [allomasok](#_bookmark329)

*Az állomások száma az útvonalon.*

* + 1. **Részletes leírás**

Interfész az objektumok sorosításához és deszerializálásához.

Memória helyes használatának ellőrzéséhez. Az Útvonal osztály az egyes vonatútvonalakat reprezentálja. Tartal- mazza az útvonal azonosítóját, az állomások számát, valamint az állomásokat.

* + 1. **Konstruktorok és destruktorok dokumentációja**
       1. **Utvonal() [1/2]**

Utvonal::Utvonal ( )

Az állomások tömbje az útvonalon.

Az osztály alapértelmezett konstruktora.

* + - 1. **Utvonal() [2/2]**

Utvonal::Utvonal (

const [Utvonal](#_bookmark297) & *other* )

Az osztály másoló konstruktora.

**Paraméterek**

- másolni kívánt [Utvonal](#_bookmark297) objektum

*other*

* + - 1. *∼***Utvonal()**

Utvonal::*∼*Utvonal ( )

Az osztály destruktora.

### Tagfüggvények dokumentációja

#### addAllomas()

void Utvonal::addAllomas (

[Allomas](#_bookmark9) *allomas* )

Állomás hozzáadása az útvonalhoz.

**Paraméterek**

- hozzáadni kívánt állomás

*allomas*

#### changeAllomas()

void Utvonal::changeAllomas (

int *index,*

[Allomas](#_bookmark9) *allomas* )

Állomás cseréje az útvonalon.

**Paraméterek**

|  |  |
| --- | --- |
| *index* | - az állomás indexe, amelyet cserélni kívánunk |
| *allomas* | - beszúrni kívánt állomás |

#### createAllomas()

void Utvonal::createAllomas (

std::string *nev,* int *erkezes\_ora,* int *erkezes\_perc,* int *indulas\_ora,* int *indulas\_perc* )

Új állomás létrehozása az útvonalon.

**Paraméterek**

|  |  |  |
| --- | --- | --- |
| *nev* | - az állomás neve |  |
| *erkezes\_ora* | - az állomásra érkezővonat óra |
| *erkezes\_perc* | - az állomásra érkezővonat perc |
| *indulas\_óra* | - az állomásról induló vonat óra |
| **Ké***i***s***n***z***d***ít***u***et***l***t***a***e** *s***D***\_***ox***p***y***e***g***r***e***c***n** | - az állomásról induló vonat perc |  |

#### getAllomas()

[Allomas](#_bookmark9) & Utvonal::getAllomas (

int *i* ) const

Visszaad egy kívánt számú állomást a listából az index alapján.

**Paraméterek**

- az állomás indexe

*i*

**Visszatérési érték**

Az állomás referenciája az adott indexen

#### getAllomasokSzama()

int Utvonal::getAllomasokSzama ( ) const

Az útvonalon található állomások számának lekérdezése.

#### kiir()

void Utvonal::kiir (

std::ostream & *os* ) const

Objektum kiírása adatfolyamba esztétikus formában.

**Paraméterek**

- Az írásra használt kimeneti adatfolyam.

*os*

#### operator=()

[Utvonal](#_bookmark297) & Utvonal::operator= (

const [Utvonal](#_bookmark297) & *other* )

Az útvonal értékadás operátora.

**Paraméterek**

- másolni kívánt [Utvonal](#_bookmark297) objektum

*other*

#### read()

void Utvonal::read (

std::istream & *is* ) [virtual]

[Serializable](#_bookmark289) interfész implementációja: objektum olvasása adatfolyamból.

**Paraméterek**

- A beolvasásra használt bemeneti adatfolyam.

*is*

Megvalósítja a következőket: [Serializable](#_bookmark292).

#### removeAllomas()

void Utvonal::removeAllomas (

int *index* )

Állomás eltávolítása az útvonalról.

**Paraméterek**

- az eltávolítani kívánt állomás indexe

*index*

#### write()

void Utvonal::write (

std::ostream & *os* ) const [virtual]

[Serializable](#_bookmark289) interfész implementációja: objektum írása adatfolyamba.

**Paraméterek**

- Az írásra használt kimeneti adatfolyam.

*os*

Megvalósítja a következőket: [Serializable](#_bookmark294).

### Adattagok dokumentációja

#### allomasok

[Allomas](#_bookmark9)*∗* Utvonal::allomasok [private]

Az állomások száma az útvonalon.

#### allomasok\_szama

int Utvonal::allomasok\_szama [private]

Az útvonal azonosítója.

#### utvonal\_azonosito

int Utvonal::utvonal\_azonosito [private]

Ez a dokumentáció az osztályról a következőfájlok alapján készült:

* + - * + [utvonal.h](#_bookmark449)
        + [utvonal.cpp](#_bookmark447)

## Vonat osztályreferencia

#include *<*vonat.h*>*

A Vonat osztály származási diagramja:

|  |  |
| --- | --- |
| Serializable | |
|  |  |
| Vonat | |

**Publikus tagfüggvények**

* [Vonat](#_bookmark340) ()

*A vonathoz tartozó útvonal.*

* [Vonat](#_bookmark342) (const [Vonat](#_bookmark336) &other)
* [Vonat](#_bookmark336) & [operator=](#_bookmark370) (const [Vonat](#_bookmark336) &other)
* [Vonat](#_bookmark344) (int azonosito, int kocsik\_sz, [Kocsi](#_bookmark215) kocsik\_tomb[ ], [Utvonal](#_bookmark297) utv, int jegyek\_sz, [Jegy](#_bookmark133) *∗∗*jegyek\_ptr)
* int [getAzonosito](#_bookmark361) () const
* [Utvonal](#_bookmark297) [getUtvonal](#_bookmark365) () const
* void [setAzonosito](#_bookmark377) (int azonosito)
* void [setUtvonal](#_bookmark379) (const [Utvonal](#_bookmark297) &[utvonal](#_bookmark391))
* void [addKocsi](#_bookmark351) ([Kocsi](#_bookmark215) &kocsi)
* void [createKocsi](#_bookmark356) (int szekek)
* void [addJegy](#_bookmark349) ([Jegy](#_bookmark133) &jegy)
* [Jegy](#_bookmark133) *∗* [getJegy](#_bookmark363) (int index) const
* bool [routeExists](#_bookmark373) (std::string indulo, std::string erkezo) const
* int [findAllomas](#_bookmark358) (std::string nev) const
* int [indulasiIdoKulonbseg](#_bookmark367) (std::string nev, int ora, int perc)
* int [createJegy](#_bookmark352) (std::string indulo, std::string erkezo, int indulo\_ora=0, int indulo\_perc=0, double discountOr*←›* Fee=0, const std::string &tipus="")
* void [write](#_bookmark381) (std::ostream &os) const
* void [read](#_bookmark372) (std::istream &is)
* [*∼*Vonat](#_bookmark347) ()

*Az osztály destruktora.*

**Privát attribútumok**

* int [vonat\_azonosito](#_bookmark393)
* int [kocsik\_szama](#_bookmark389)

*A vonat azonosítója.*

* int [jegyek\_szama](#_bookmark385)

*A vonathoz tartozó kocsik száma.*

* [Kocsi](#_bookmark215) *∗* [kocsik](#_bookmark387)

*A vonathoz tartozó jegyek száma.*

* [Jegy](#_bookmark133) *∗∗* [jegyek](#_bookmark384)

*A vonathoz tartozó kocsik tömbje.*

* [Utvonal](#_bookmark297) [utvonal](#_bookmark391)

*A vonathoz tartozó jegyek tömbje.*

### Részletes leírás

A [Vonat](#_bookmark336) osztály a reprezentálja a vonatokat. Tartalmazza a vonat azonosítóját, a vonathoz tartozó kocsik számát, a kocsik tömbjét, a vonathoz tartozó jegyeket, azok számát, valamint az útvonalát.

### Konstruktorok és destruktorok dokumentációja

* + - 1. **Vonat() [1/3]**

Vonat::Vonat ( )

A vonathoz tartozó útvonal.

Az osztály alapértelmezett konstruktora.

* + - 1. **Vonat() [2/3]**

Vonat::Vonat (

const [Vonat](#_bookmark336) & *other* )

Az osztály másoló konstruktora.

**Paraméterek**

- Másolni kívánt [Vonat](#_bookmark336) objektum.

*other*

* + - 1. **Vonat() [3/3]**

Vonat::Vonat (

int *azonosito,*

int *kocsik\_sz,*

[Kocsi](#_bookmark215) *kocsik\_tomb[ ],*

[Utvonal](#_bookmark297) *utv,*

int *jegyek\_sz,*

[Jegy](#_bookmark133) *∗∗ jegyek\_ptr* )

Paraméteres konstruktor.

**Paraméterek**

|  |  |
| --- | --- |
| *azonosito* | - A vonat azonosítója. |
| *kocsik\_sz* | - A vonathoz tartozó kocsik száma. |
| *kocsik\_tomb* | - A kocsik tömbje. |
| *utv* | - Az útvonal, amelyhez a vonat tartozik. |
| *jegyek\_sz* | - A vonathoz tartozó jegyek száma. |
| *jegyek\_ptr* | - A jegyek tömbjére mutató pointer. |

* + - 1. *∼***Vonat()**

Vonat::*∼*Vonat ( )

Az osztály destruktora.

* + 1. **Tagfüggvények dokumentációja**

#### addJegy()

void Vonat::addJegy (

[Jegy](#_bookmark133) & *jegy* )

[Jegy](#_bookmark133) hozzáadása a vonathoz.

**Paraméterek**

- hozzáadni kívánt jegy

*jegy*

#### addKocsi()

void Vonat::addKocsi (

[Kocsi](#_bookmark215) & *kocsi* )

[Kocsi](#_bookmark215) hozzáadása a vonathoz.

**Paraméterek**

- hozzáadni kívánt kocsi

*kocsi*

#### createJegy()

int Vonat::createJegy (

std::string *indulo,* std::string *erkezo,* int *indulo\_ora = 0,* int *indulo\_perc = 0,*

double *discountOrFee = 0,*

const std::string & *tipus = ""* )

Új jegy létrehozása az adott indulási és érkezési állomások között.

**Paraméterek**

|  |  |
| --- | --- |
| *indulo* | - Az indulási állomás neve. |
| *erkezo* | - Az érkezési állomás neve. |
| *indulo\_ora* | - Az indulási időórája (alapértelmezett érték: 0). |
| *indulo\_perc* | - Az indulási időperce (alapértelmezett érték: 0). |
| *discountOrFee* | - Kedvezmény vagy felár mértéke (alapértelmezett érték: 0). |
| *tipus* | - [Jegy](#_bookmark133) típusa (alapértelmezett érték: üres string). |

**Visszatérési érték**

Az újonnan létrehozott jegy azonosítója.

#### createKocsi()

void Vonat::createKocsi (

int *szekek* )

Új kocsi létrehozása a vonathoz.

**Paraméterek**

- Az új kocsin elérhetőszékek száma.

*szekek*

#### findAllomas()

int Vonat::findAllomas (

std::string *nev* ) const

Megkeresi az állomás indexét az állomás neve alapján.

**Paraméterek**

- Az állomás neve, aminek az indexét keresi.

*nev*

**Visszatérési érték**

Az állomás indexe, vagy -1, ha nem található az állomás.

#### getAzonosito()

int Vonat::getAzonosito ( ) const

Azonosító lekérdezése.

**Visszatérési érték**

A vonat azonosítója.

#### getJegy()

[Jegy](#_bookmark133) *∗* Vonat::getJegy (

int *index* ) const

[Jegy](#_bookmark133) lekérdezése adott indexen.

**Paraméterek**

- A jegy indexe a tömbben.

*index*

**Visszatérési érték**

Az adott indexen található jegy pointer.

#### getUtvonal()

[Utvonal](#_bookmark297) Vonat::getUtvonal ( ) const

Útvonal lekérdezése.

**Visszatérési érték**

Az útvonal, amelyhez a vonat tartozik.

#### indulasiIdoKulonbseg()

int Vonat::indulasiIdoKulonbseg (

std::string *nev,*

int *ora,*

int *perc* )

Meghatározza, hogy hány perc különbség van a megadott időpont és a vonat indulásának időpontja között.

**Paraméterek**

|  |  |
| --- | --- |
| *nev* | - Az állomás neve, ahonnan az indulási időt számítja. |
| *ora* | - Az indulási időórája. |
| *perc* | - Az indulási időperce. |

**Visszatérési érték**

Perc különbség van a megadott időpont és a vonat indulásának időpontja között vagy -1, ha az állomás nem található vagy a vonat hamarabb indul mint a megadott időpont.

#### operator=()

[Vonat](#_bookmark336) & Vonat::operator= (

const [Vonat](#_bookmark336) & *other* )

Értékadó operátor.

**Paraméterek**

- Másolni kívánt [Vonat](#_bookmark336) objektum.

*other*

**Visszatérési érték**

Az értékadás eredménye, az új [Vonat](#_bookmark336) objektum referenciája.

#### read()

void Vonat::read (

std::istream & *is* ) [virtual]

[Serializable](#_bookmark289) interfész implementációja: objektum olvasása adatfolyamból.

**Paraméterek**

- A beolvasásra használt bemeneti adatfolyam.

*is*

Megvalósítja a következőket: [Serializable](#_bookmark292).

#### routeExists()

bool Vonat::routeExists (

std::string *indulo,*

std::string *erkezo* ) const

Ellenőrzi, hogy létezik-e útvonal az adott indulási és érkezési állomások között.

**Paraméterek**

|  |  |
| --- | --- |
| *indulo* | - Az indulási állomás neve. |
| *erkezo* | - Az érkezési állomás neve. |

**Visszatérési érték**

igaz, ha létezik útvonal, különben hamis.

#### setAzonosito()

void Vonat::setAzonosito (

int *azonosito* )

Azonosító beállítása.

**Paraméterek**

- Az új vonat azonosítója.

*azonosito*

#### setUtvonal()

void Vonat::setUtvonal (

const [Utvonal](#_bookmark297) & *utvonal* )

Útvonal beállítása.

**Paraméterek**

- az útvonal, amelyhez a vonat tartozik

*utvonal*

#### write()

void Vonat::write (

std::ostream & *os* ) const [virtual]

[Serializable](#_bookmark289) interfész implementációja: objektum írása adatfolyamba.

**Paraméterek**

- Az írásra használt kimeneti adatfolyam.

*os*

Megvalósítja a következőket: [Serializable](#_bookmark294).

### Adattagok dokumentációja

#### jegyek

[Jegy](#_bookmark133)*∗∗* Vonat::jegyek [private]

A vonathoz tartozó kocsik tömbje.

#### jegyek\_szama

int Vonat::jegyek\_szama [private]

A vonathoz tartozó kocsik száma.

#### kocsik

[Kocsi](#_bookmark215)*∗* Vonat::kocsik [private]

A vonathoz tartozó jegyek száma.

#### kocsik\_szama

int Vonat::kocsik\_szama [private]

A vonat azonosítója.

#### utvonal

[Utvonal](#_bookmark297) Vonat::utvonal [private]

A vonathoz tartozó jegyek tömbje.

#### vonat\_azonosito

int Vonat::vonat\_azonosito [private]

Ez a dokumentáció az osztályról a következőfájlok alapján készült:

* + - * + [vonat.h](#_bookmark455)
        + [vonat.cpp](#_bookmark453)

1. **fejezet**

**Fájlok dokumentációja**

## allomas.cpp fájlreferencia

#include "allomas.h" #include *<*cstring*>* #include *<*cstddef*>* #include *<*iostream*>*

## allomas.h fájlreferencia

#include "ido.h" #include *<*iostream*>* #include "memtrace.h"

#include "serializable.h"

**Osztályok**

* class [Allomas](#_bookmark9)

*Időtárolásához.*

## allomas.h

[Ugrás a fájl dokumentációjához.](#_bookmark400)

00001 #ifndef ALLOMAS\_H

00002 #define ALLOMAS\_H

00003

00005 #include "[ido.h](#_bookmark406)"

00006

00008 #include <iostream>

00009

00011 #include "memtrace.h"

00012

00014 #include "[serializable.h](#_bookmark433)"

00015

00017 class [Allomas](#_bookmark9) : public [Serializable](#_bookmark289) 00018 {

00019 int [allomas\_azonosito](#_bookmark51);

00020 std::string [nev](#_bookmark56);

00021 [Ido](#_bookmark89) [indulas](#_bookmark54);

00022 [Ido](#_bookmark89) [erkezes](#_bookmark52);

00023 public:

00025 [Allomas](#_bookmark13)();

00026

00029 [Allomas](#_bookmark13)(const [Allomas](#_bookmark9) &other); 00030

00032 [Allomas](#_bookmark9) &[operator=](#_bookmark44)(const [Allomas](#_bookmark9) &other); 00033

00041 [Allomas](#_bookmark13)(int azonosito, const std::string &[nev](#_bookmark56), int erkezes\_ora, int erkezes\_perc, int indulas\_ora, int indulas\_perc);

00042

00046 void [changeErkezes](#_bookmark22)(int ora, int perc); 00047

00051 void [changeIndulas](#_bookmark24)(int ora, int perc); 00052

00055 void [changeNev](#_bookmark25)(const std::string &other); 00056

00059 int [getIndulasOra](#_bookmark36)();

00060

00063 int [getIndulasPerc](#_bookmark40)();

00064

00067 int [getErkezesOra](#_bookmark31)();

00068

00071 int [getErkezesPerc](#_bookmark33)();

00072

00075 [Ido](#_bookmark89) &[getIndulas](#_bookmark35)();

00076

00079 [Ido](#_bookmark89) &[getErkezes](#_bookmark29)();

00080

00083 std::string &[getNev](#_bookmark42)();

00084

00087 void [write](#_bookmark47)(std::ostream &os) const; 00088

00091 void [read](#_bookmark45)(std::istream &is); 00092

00094 [~Allomas](#_bookmark20)();

00095 };

00096

00097 #endif

## ido.cpp fájlreferencia

#include "ido.h"

## ido.h fájlreferencia

#include *<*iostream*>* #include "serializable.h" #include "memtrace.h"

**Osztályok**

* class [Ido](#_bookmark89)

*Szabványos bemenet és kimenet kezeléséhez.*

**5.6 ido.h** **61**

* 1. **ido.h**

[Ugrás a fájl dokumentációjához.](#_bookmark406)

00001 #ifndef IDO\_H

00002 #define IDO\_H

00003

00005 #include <iostream>

00006

00008 #include "[serializable.h](#_bookmark433)"

00009

00011 #include "memtrace.h"

00012

00015 class [Ido](#_bookmark89) : public [Serializable](#_bookmark289) 00016 {

00017 private:

00018 int [ora](#_bookmark129);

00019 int [perc](#_bookmark130);

00020 public:

00022 [Ido](#_bookmark93)();

00023

00027 [Ido](#_bookmark93)(int o, int p); 00028

00031 [Ido](#_bookmark93)(const [Ido](#_bookmark89) &other); 00032

00035 int [getOra](#_bookmark106)() const; 00036

00039 int [getPerc](#_bookmark108)() const; 00040

00043 void [setOra](#_bookmark122)(int o); 00044

00047 void [setPerc](#_bookmark124)(int p); 00048

00052 void [setIdo](#_bookmark119)(int o, int p); 00053

00056 void [addPerc](#_bookmark104)(int p); 00057

00060 void [addOra](#_bookmark102)(int o); 00061

00065 bool [operator==](#_bookmark115)(const [Ido](#_bookmark89) &other) const; 00066

00070 bool [operator!=](#_bookmark113)(const [Ido](#_bookmark89) &other) const; 00071

00074 void [kiir](#_bookmark109)(std::ostream &os) const; 00075

00078 void [write](#_bookmark126)(std::ostream &os) const; 00079

00082 void [read](#_bookmark117)(std::istream &is); 00083

00085 [~Ido](#_bookmark97)();

00086 };

00087

00088 #endif

## jegy.cpp fájlreferencia

#include "jegy.h" #include *<*iostream*>*

## jegy.h fájlreferencia

#include *<*iostream*>* #include "serializable.h" #include "memtrace.h" #include "allomas.h" #include "ido.h"

**Osztályok**

* class [Jegy](#_bookmark133)

*Szabványos bemenet és kimenet kezeléséhez.*

* class [KedvezmenyesJegy](#_bookmark186)
* class [FelarasJegy](#_bookmark60)

## jegy.h

[Ugrás a fájl dokumentációjához.](#_bookmark412)

00001 #ifndef JEGY\_H

00002 #define JEGY\_H

00003

00005 #include <iostream>

00006

00008 #include "[serializable.h](#_bookmark433)"

00009

00011 #include "memtrace.h"

00012

00013 #include "[allomas.h](#_bookmark400)"

00014

00016 #include "[ido.h](#_bookmark406)"

00017

00021 class [Jegy](#_bookmark133) : public [Serializable](#_bookmark289) 00022 {

00023 protected:

00024 int [jegy\_azonosito](#_bookmark178);

00025 int [helyszam](#_bookmark172);

00026 int [kocsi\_szam](#_bookmark180);

00027 int [vonat\_szam](#_bookmark182);

00028 std::string [indulo\_nev](#_bookmark176);

00029 [Ido](#_bookmark89) [indulo\_ido](#_bookmark174);

00030 std::string [cel\_nev](#_bookmark169);

00031 [Ido](#_bookmark89) [cel\_ido](#_bookmark167);

00032 int [ar](#_bookmark166);

00033 virtual int [getAr](#_bookmark149)(double pred); 00034 public:

00036 [Jegy](#_bookmark137)();

00037

00045 [Jegy](#_bookmark137)(int azonosito, int hely, int kocsi, int vonat, const std::string &indulo, [Ido](#_bookmark89) indulo\_idopont, 00046 const std::string &cel, [Ido](#_bookmark89) cel\_idopont);

00047

00056 [Jegy](#_bookmark137)(int azonosito, int hely, int kocsi, int vonat, const std::string &indulo, [Ido](#_bookmark89) indulo\_idopont, 00057 const std::string &cel, [Ido](#_bookmark89) cel\_idopont, double pred);

00058

00061 [Jegy](#_bookmark137)(const [Jegy](#_bookmark133) &other); 00062

00066 [Jegy](#_bookmark133) &[operator=](#_bookmark157)(const [Jegy](#_bookmark133) &other); 00067

00071 bool [operator==](#_bookmark159)(const [Jegy](#_bookmark133) &other) const; 00072

00076 bool [operator!=](#_bookmark155)(const [Jegy](#_bookmark133) &other) const; 00077

00080 void [kiir](#_bookmark151)(std::ostream &os) const; 00081

00084 void [write](#_bookmark163)(std::ostream &os) const; 00085

00088 void [read](#_bookmark162)(std::istream &is); 00089

00092 virtual [Jegy](#_bookmark133) \*[clone](#_bookmark148)() const; 00093

00095 virtual [~Jegy](#_bookmark146)() {} 00096 };

00097

00100 class [KedvezmenyesJegy](#_bookmark186) : public [Jegy](#_bookmark133) 00101 {

00102 std::string [tipus](#_bookmark212);

00103 double [kedvezmeny](#_bookmark211);

00104 public:

00106 [KedvezmenyesJegy](#_bookmark190)();

00107

00110 [KedvezmenyesJegy](#_bookmark190)(const [KedvezmenyesJegy](#_bookmark186) &other); 00111

00121 [KedvezmenyesJegy](#_bookmark190)(int azonosito, int hely, int kocsi, int vonat, const std::string &indulo, [Ido](#_bookmark89) indulo\_idopont,

00122 const std::string &cel, [Ido](#_bookmark89) cel\_idopont, double [kedvezmeny](#_bookmark211), std::string [tipus](#_bookmark212)); 00123

00127 [KedvezmenyesJegy](#_bookmark186) &[operator=](#_bookmark204)(const [KedvezmenyesJegy](#_bookmark186) &other);

**5.10 kocsi.cpp fájlreferencia** **63**

00128

00131 void [kiir](#_bookmark200)(std::ostream &os) const; 00132

00135 void [write](#_bookmark207)(std::ostream &os) const; 00136

00139 void [read](#_bookmark206)(std::istream &is); 00140

00143 [Jegy](#_bookmark133) \*[clone](#_bookmark199)() const; 00144

00146 [~KedvezmenyesJegy](#_bookmark197)();

00147 };

00148

00151 class [FelarasJegy](#_bookmark60) : public [Jegy](#_bookmark133) 00152 {

00153 std::string [tipus](#_bookmark86);

00154 double [felar](#_bookmark85);

00155 public:

00157 [FelarasJegy](#_bookmark64)();

00158

00161 [FelarasJegy](#_bookmark64)(const [FelarasJegy](#_bookmark60) &other); 00162

00172 [FelarasJegy](#_bookmark64)(int azonosito, int hely, int kocsi, int vonat, const std::string &indulo, [Ido](#_bookmark89) indulo\_idopont,

00173 const std::string &cel, [Ido](#_bookmark89) cel\_idopont, double [felar](#_bookmark85), std::string [tipus](#_bookmark86)); 00174

00178 [FelarasJegy](#_bookmark60) &[operator=](#_bookmark78)(const [FelarasJegy](#_bookmark60) &other); 00179

00182 void [kiir](#_bookmark74)(std::ostream &os) const; 00183

00186 void [write](#_bookmark81)(std::ostream &os) const; 00187

00190 void [read](#_bookmark80)(std::istream &is); 00191

00194 [Jegy](#_bookmark133) \*[clone](#_bookmark73)() const; 00195

00197 [~FelarasJegy](#_bookmark71)();

00198 };

00199

00200 #endif

* 1. **kocsi.cpp fájlreferencia**

#include "kocsi.h"

## kocsi.h fájlreferencia

#include *<*iostream*>* #include "serializable.h" #include "memtrace.h"

**Osztályok**

* class [Kocsi](#_bookmark215)

*Szabványos bemenet és kimenet kezeléséhez.*

## kocsi.h

[Ugrás a fájl dokumentációjához.](#_bookmark418)

00001 #ifndef KOCSI\_H

00002 #define KOCSI\_H

00003

00005 #include <iostream>

00006

00008 #include "[serializable.h](#_bookmark433)"

00009

00011 #include "memtrace.h"

00012

00015 class [Kocsi](#_bookmark215) : public [Serializable](#_bookmark289) 00016 {

00017 int [kocsi\_azonosito](#_bookmark241);

00018 int [szekek\_szama](#_bookmark243);

00019 int [foglalt\_szekek\_szama](#_bookmark240);

00020 public:

00022 [Kocsi](#_bookmark219)();

00023

00026 [Kocsi](#_bookmark219)(const [Kocsi](#_bookmark215) &other); 00027

00031 [Kocsi](#_bookmark219)(int azonosito, int szekek); 00032

00035 bool [tele](#_bookmark236)() const; 00036

00039 int [reserveHely](#_bookmark232)();

00040

00043 int [getSzekek\_szama](#_bookmark229)() const; 00044

00047 int [getAzonosito](#_bookmark227)() const; 00048

00051 void [write](#_bookmark238)(std::ostream &os) const; 00052

00055 void [read](#_bookmark231)(std::istream &is); 00056

00058 [~Kocsi](#_bookmark224)();

00059 };

00060

00061 #endif

## main.cpp fájlreferencia

#include "menetrend.h" #include "vonat.h" #include "jegy.h" #include "utvonal.h" #include "ido.h" #include "allomas.h" #include "kocsi.h" #include "test.h" #include *<*iostream*>* #include *<*sstream*>* #include *<*fstream*>* #include "memtrace.h"

**Függvények**

* int [main](#_bookmark424) ()

### Függvények dokumentációja

#### main()

int main ( )

**5.14 menetrend.cpp fájlreferencia** **65**

* 1. **menetrend.cpp fájlreferencia**

#include "menetrend.h"

## menetrend.h fájlreferencia

#include "vonat.h" #include *<*iostream*>* #include "serializable.h" #include "memtrace.h"

**Osztályok**

* class [Menetrend](#_bookmark247)

*Szabványos bemenet és kimenet kezeléséhez.*

## menetrend.h

[Ugrás a fájl dokumentációjához.](#_bookmark429)

00001 #ifndef MENETREND\_H

00002 #define MENETREND\_H

00003

00004 #include "[vonat.h](#_bookmark455)"

00005

00007 #include <iostream>

00008

00010 #include "[serializable.h](#_bookmark433)"

00011

00013 #include "memtrace.h"

00014

00017 class [Menetrend](#_bookmark247) : public [Serializable](#_bookmark289) 00018 {

00019 int [vonatok\_szama](#_bookmark286);

00020 [Vonat](#_bookmark336) \*[vonatok](#_bookmark285);

00021 public:

00023 [Menetrend](#_bookmark251)() : [vonatok\_szama](#_bookmark286)(0), [vonatok](#_bookmark285)(NULL) {} 00024

00027 [Menetrend](#_bookmark251)([Vonat](#_bookmark336) v);

00028

00032 [Menetrend](#_bookmark254)([Vonat](#_bookmark336) \*v, int meret) : [vonatok\_szama](#_bookmark286)(meret), [vonatok](#_bookmark285)(v) {} 00033

00036 [Menetrend](#_bookmark256)(const [Menetrend](#_bookmark247) &m); 00037

00040 void [addVonat](#_bookmark262)([Vonat](#_bookmark336) v); 00041

00045 void [changeVonat](#_bookmark264)([Vonat](#_bookmark336) v, int index); 00046

00050 [Vonat](#_bookmark336) &[getVonat](#_bookmark271)(int index) const; 00051

00054 int [getVonatokSzama](#_bookmark273)() const; 00055

00058 void [removeVonat](#_bookmark279)(int index); 00059

00067 void [createJegy](#_bookmark267)(std::string indulo, std::string erkezo, int indulo\_ora, int indulo\_perc, double discountOrFee = 0, const std::string &tipus = "");

00068

00070 void [clear](#_bookmark266)();

00071

00080 void [kiir](#_bookmark275)(std::ostream &os, std::string indulo, std::string erkezo) const; 00081

00084 void [write](#_bookmark282)(std::ostream &os) const; 00085

00088 void [read](#_bookmark278)(std::istream &is); 00089

00091 [~Menetrend](#_bookmark258)();

00092 };

00093

00094 #endif

## serializable.h fájlreferencia

#include *<*iostream*>*

#include "memtrace.h"

**Osztályok**

* class [Serializable](#_bookmark289)

*Ostream és istream használatához.*

## serializable.h

[Ugrás a fájl dokumentációjához.](#_bookmark433)

00001 #ifndef SERIALIZABLE\_H

00002 #define SERIALIZABLE\_H

00003

00005 #include <iostream>

00006

00008 #include "memtrace.h"

00009

00012 class [Serializable](#_bookmark289)

00013 {

00014 public:

00017 virtual void [write](#_bookmark294)(std::ostream &os) const = 0; 00018

00021 virtual void [read](#_bookmark292)(std::istream &is) = 0; 00022 };

00023

00024 #endif

## test.cpp fájlreferencia

#include "test.h"

**Függvények**

* void [test](#_bookmark438) ()

*Teszteléskor vizsgált osztályok.*

### Függvények dokumentációja

#### test()

void test ( )

Teszteléskor vizsgált osztályok.

Programhoz tartozó teszteket futtató függvény

**Paraméterek**

*void*

**Visszatérési érték**

void

Időosztály kiiírás funkciójánák vizsgálata StringStream létrehozása a kiíráshoz

időobjektumok létrehozása paraméteres konstruktorokkal 10 óra, 12 perc

2 óra, 1 perc

időobjektumok kiírása vart: "10:12"

vart: "02:01"

kiírt értékek vizsgálata

/// Időosztály perzisztenciájának vizsgálata StringStream létrehozása

időobjektum létrehozása paraméteres konstruktorokkal 8 óra, 30 perc

időobjektum létrehozása beolvasáshoz ido1 objektum kiírása StringStream-be

ido2 objektum beolvasása a StringStream-ből időobjektumok egyezésének vizsgálata

[Ido](#_bookmark89) osztály alapvetőmu˝ veleteinek vizsgálata

időobjektum létrehozása paraméteres konstruktorokkal 2 óra, 45 perc

konstruktor értékbeállításának vizsgálata új értékek beállítása setter függvényekel setterek mu˝ ködésének ellenőrzése

órák és percek hozzáadása

aritmetikai mu˝ velet ellenőrzése 3 óra 15 perc + 2 óra 30 perc = 5 óra 45perc

időobjektum létrehozása egyezés vizsgálathoz egyenlőség operátor ellenőrzése

különbözőség operátor ellenőrzése Állomás konstruktorának vizsgálata állomás létrehozása konstruktorral

paraméterek helyes beállítottságának vizsgálata állomás létrehozása paraméteres konstruktorral paraméterek helyes beállítottságának vizsgálata Állomás osztály attribútum változtatásának vizsgálata állomás létrehozása paraméteres konstruktorral indulás és érkezés időpontjainak változtatása

indulás és érkezés sikeres változtatásának ellenőrzése állomás létrehozása paraméteres konstruktorral

nev paraméter módosítása

nev paraméter állításának ellenőrzése Állomás osztály perzisztenciájának vizsgálata StringStream létrehozása

állomás létrehozása paraméteres konstruktorral állomás létrehozása beolvasáshoz

állomás kiírása StringStream-be állomás beolvasása StringStream-ből

eredeti es beolvasott objektum egyezésének vizsgálata Útvonal osztály állomás hozzáadás függvényének vizsgálata Útvonal objektum létrehozása

állomás objektum létrehozása paraméteres konstruktorral állomások hozzáadása az útvonalhoz

útvonalban tárolt állomások ellenőrzése Útvonal állomás cseréjének vizsgálata Útvonal létrehozása

állomások létrehozása paraméteres konstruktorral

állomásokok hozzáadása az útvonalhoz

útvonalban tárolt 2. állomás cseréje másik állomásra útvonalban tárolt állomások vizsgálata a csere után Útvonal osztály perzisztenciájának vizsgálata Útvonal létrehozása

állomások létrehozása paraméteres konstruktorral állomásokok hozzáadása az útvonalhoz StringStream létrehozása

útvonal objektum kiírása StringStream-be útvonal objektum létrehozása beolvasáshoz útvonal objektum beolvasása StringStream-ből útvonal objektumok egyezésének vizsgálata

[Kocsi](#_bookmark215) osztály szék foglalás funkciójának vizsgálata [Kocsi](#_bookmark215) objektum létrehozása 5 u˝ lőhely kapacitással minden u˝ lőhely lefoglalasa

nincs több szabad hely, foglalásnál hibát kell dobnia

[Kocsi](#_bookmark215) osztály foglaltság ellenőrzőfüggvényének vizsgálata [Kocsi](#_bookmark215) objektum létrehozása 2 u˝ lőhely kapacitással

van szabad hely, a tele függvény hamis értéket kell adjon minden u˝ lőhely lefoglalasa

nincs több szabad hely, a tele függvény igaz értéket kell adjon [Kocsi](#_bookmark215) osztály perzisztenciájának vizsgálata

[Kocsi](#_bookmark215) objektum létrehozása 3 u˝ lőhely kapacitással 1 hely lefoglalása

StringStream létrehozása

kocsi objektum kiírása StringStream-be kocsi objektum létrehozása beolvasáshoz

kcosi objektum beolvasása a StringStream-ből kocsi objektumok egyezésének vizsgálata

[Vonat](#_bookmark336) osztály útvonal beállíto és lekérdezőfüggvényeinek vizsgálata

Útvonal objektum létrehozása állomások létrehozása az útvonalon [Vonat](#_bookmark336) objektum létrehozása

útvonal beállítása

útvonal lekérdezése és eltárolása

eredeti es beolvasott objektum egyezésének vizsgálata [Vonat](#_bookmark336) osztály jegy hozzáadó függvényének vizsgálata [Vonat](#_bookmark336) objektumok létrehozása

időobjektum létrehozása paraméteres konstruktorokkal 8 óra, 0 perc

10 óra, 30 perc [Jegy](#_bookmark133) létrehozása

jegy hozzáadása a vonathoz

eredeti és vonat-ból kapott jegyek egyezésének vizsgálata

[Vonat](#_bookmark336) osztaly indulási időkülönbség számító függvényének vizsgálata Útvonal objektum létrehozása

állomások létrehozása az útvonalon [Vonat](#_bookmark336) objektum létrehozása

útvonal beállítása

időkülönbség kiszámítása a vonat indulása Budapest állomásról és a megadott időpont között

különbség ellenőzrése az állomásról való indulás ideje 10:30, ami 21 percel később van mint a megadott 10:09 [Vonat](#_bookmark336) osztály perzisztenciájának vizsgálata

Kocsik létrehozása

Időobjektumok létrehozása Jegyek létrehozása

Útvonal létrehozása ás állomások hozzáadása állomások létrehozása az útvonalon StringStream létrehozása

vonat objektum kiírása StringStream-be vonat objektum létrehozása beolvasáshoz

kcosi objektum beolvasása a StringStream-ből vonatok egyezésének vizsgálata

[Jegy](#_bookmark133) osztály == operátorának vizsgálata

időobjektumok létrehozása paraméteres konstruktorokkal jegyek létrehozása

jegyek egyezésének vizsgálata

[Jegy](#_bookmark133) osztály másoló konstruktorának vizsgálata

időobjektumok létrehozása paraméteres konstruktorokkal jegy létrehozása

jegy létrehozása másoló konstruktorral jegyek egyezésének vizsgálata

[Jegy](#_bookmark133) osztály értékadó operátorának vizsgálata

időobjektumok létrehozása paraméteres konstruktorokkal jegy létrehozása

jegy létrehozása értékadáshoz jegy értéküladása

jegyek egyezésének vizsgálata

[Jegy](#_bookmark133) osztály perzisztenciájának vizsgálata

időobjektumok létrehozása paraméteres konstruktorokkal jegy létrehozása

StringStream létrehozása jegy kiírása StringStream-be

jegy létrehozása beolvasáshoz jegy beolvasása StringStream-ből jegyek egyezésének vizsgálata

Felarasjegy osztály perzisztenciájának vizsgálata

időobjektumok létrehozása paraméteres konstruktorokkal felarasjegy létrehozása

StringStream létrehozása jegy kiírása StringStream-be

jegy létrehozása beolvasáshoz jegy beolvasása StringStream-ből jegyek egyezésének vizsgálata

Kedvezmenyesjegy osztály perzisztenciájának vizsgálata időobjektumok létrehozása paraméteres konstruktorokkal kedvezmenyesjegy létrehozása

StringStream létrehozása jegy kiírása StringStream-be

jegy létrehozása beolvasáshoz jegy beolvasása StringStream-ből jegyek egyezésének vizsgálata Állomás konstruktorának vizsgálata menetrend objektum létrehozása

paraméterek helyes beállítottságának vizsgálata [Menetrend](#_bookmark247) vonat hozzáadó függvényének vizsgálata menetrend objektum létrehozása

útvonal objektum létrehozása állomások létrehozása az útvonalon vonat lérehozása az útvonallal vonat hozzáadása a menetrendhez vonat hozzáadásának ellenőrzése

[Menetrend](#_bookmark247) vonat módosításának vizsgálata menetrend objektum létrehozása

útvonal objektumok létrehozása állomások létrehozása az útvonalon vonatok lérehozása az útvonallal vonat hozzáadása a menetrendhez hozzáadott vonat cseréje

vonat cseréjének ellenőrzése

[Menetrend](#_bookmark247) osztály perzisztenciájának vizsgálata menetrend objektum létrehozása

útvonal objektum létrehozása állomások létrehozása az útvonalon vonat lérehozása az útvonallal vonat hozzáadása a menetrendhez [Menetrend](#_bookmark247) kiírása StringStream-be menetrend kiírása StringStream-be

[Menetrend](#_bookmark247) objektum létrehozása beolvasáshoz menetrend beolvasása StringStream-ből menetrendek egyezésének vizsgálata

## test.h fájlreferencia

#include "menetrend.h" #include "vonat.h" #include "jegy.h" #include "utvonal.h" #include "ido.h" #include "allomas.h" #include "kocsi.h" #include *<*sstream*>* #include *<*iostream*>* #include "memtrace.h" #include "gtest\_lite.h"

**Függvények**

* void [test](#_bookmark443) ()

*Teszteléskor vizsgált osztályok.*

### Függvények dokumentációja

#### test()

void test ( )

Teszteléskor vizsgált osztályok.

Perzisztencia vizsgálatához. Memória helyes használatának ellőrzéséhez. Teszteléshez használt makrók. Prog- ramhoz tartozó teszteket futtató függvény.

Programhoz tartozó teszteket futtató függvény

**Paraméterek**

*void*

**Visszatérési érték**

void

## test.h

[Ugrás a fájl dokumentációjához.](#_bookmark441)

00001 #ifndef TEST\_H\_

00002 #define TEST\_H\_

00003

00005 #include "[menetrend.h](#_bookmark429)"

00006 #include "[vonat.h](#_bookmark455)"

00007 #include "[jegy.h](#_bookmark412)"

00008 #include "[utvonal.h](#_bookmark449)"

00009 #include "[ido.h](#_bookmark406)"

00010 #include "[allomas.h](#_bookmark400)"

00011 #include "[kocsi.h](#_bookmark418)"

00012

00014 #include <sstream>

00015 #include <iostream>

00016

00018 #include "memtrace.h"

00020 #include "gtest\_lite.h"

00021

00023 void [test](#_bookmark443)();

00024

00025 #endif /\* TEST\_H\_ \*/

## utvonal.cpp fájlreferencia

#include "utvonal.h"

## utvonal.h fájlreferencia

#include "allomas.h" #include "serializable.h" #include "memtrace.h"

**Osztályok**

* class [Utvonal](#_bookmark297)

*Interfész az objektumok sorosításához és deszerializálásához.*

## utvonal.h

[Ugrás a fájl dokumentációjához.](#_bookmark449)

00001 #ifndef UTVONAL\_H

00002 #define UTVONAL\_H

00003

00004 #include "[allomas.h](#_bookmark400)"

00005

00007 #include "[serializable.h](#_bookmark433)"

00008

00010 #include "memtrace.h"

00011

00014 class [Utvonal](#_bookmark297) : public [Serializable](#_bookmark289) 00015 {

00016 int [utvonal\_azonosito](#_bookmark332);

00017 int [allomasok\_szama](#_bookmark330);

00018 [Allomas](#_bookmark9) \*[allomasok](#_bookmark329);

00019 public:

00021 [Utvonal](#_bookmark301)();

00022

00025 [Utvonal](#_bookmark301)(const [Utvonal](#_bookmark297) &other); 00026

00029 void [addAllomas](#_bookmark308)([Allomas](#_bookmark9) allomas); 00030

00037 void [createAllomas](#_bookmark312)(std::string nev, int erkezes\_ora, int erkezes\_perc, int indulas\_ora, int indulas\_perc);

00038

00041 [Utvonal](#_bookmark297) &[operator=](#_bookmark320)(const [Utvonal](#_bookmark297) &other); 00042

00046 void [changeAllomas](#_bookmark310)(int index, [Allomas](#_bookmark9) allomas); 00047

00050 void [removeAllomas](#_bookmark324)(int index); 00051

00053 int [getAllomasokSzama](#_bookmark316)() const; 00054

00058 [Allomas](#_bookmark9) &[getAllomas](#_bookmark313)(int i) const; 00059

00062 void [kiir](#_bookmark317)(std::ostream &os) const; 00063

00066 void [write](#_bookmark327)(std::ostream &os) const; 00067

00070 void [read](#_bookmark322)(std::istream &is); 00071

00073 [~Utvonal](#_bookmark304)();

00074 };

00075

00076 #endif

## vonat.cpp fájlreferencia

#include "vonat.h"

## vonat.h fájlreferencia

#include *<*iostream*>* #include "memtrace.h" #include "serializable.h" #include "kocsi.h" #include "jegy.h" #include "utvonal.h"

**Osztályok**

* class [Vonat](#_bookmark336)

## vonat.h

[Ugrás a fájl dokumentációjához.](#_bookmark455)

00001 #ifndef VONAT\_H

00002 #define VONAT\_H

00003

00004 #include <iostream>

00005

00006 #include "memtrace.h"

00007

00008 #include "[serializable.h](#_bookmark433)"

00009 #include "[kocsi.h](#_bookmark418)"

00010 #include "[jegy.h](#_bookmark412)"

00011 #include "[utvonal.h](#_bookmark449)"

00012

00016 class [Vonat](#_bookmark336) : public [Serializable](#_bookmark289) 00017 {

00018 private:

00019 int [vonat\_azonosito](#_bookmark393);

00020 int [kocsik\_szama](#_bookmark389);

00021 int [jegyek\_szama](#_bookmark385);

00022 [Kocsi](#_bookmark215) \*[kocsik](#_bookmark387);

00023 [Jegy](#_bookmark133) \*\*[jegyek](#_bookmark384);

00024 [Utvonal](#_bookmark297) [utvonal](#_bookmark391);

00025 public:

00027 [Vonat](#_bookmark340)();

00028

00031 [Vonat](#_bookmark340)(const [Vonat](#_bookmark336) &other); 00032

00036 [Vonat](#_bookmark336) &[operator=](#_bookmark370)(const [Vonat](#_bookmark336) &other); 00037

00045 [Vonat](#_bookmark340)(int azonosito, int kocsik\_sz, [Kocsi](#_bookmark215) kocsik\_tomb[], [Utvonal](#_bookmark297) utv, int jegyek\_sz, [Jegy](#_bookmark133)

\*\*jegyek\_ptr);

00046

00049 int [getAzonosito](#_bookmark361)() const; 00050

00053 [Utvonal](#_bookmark297) [getUtvonal](#_bookmark365)() const; 00054

00057 void [setAzonosito](#_bookmark377)(int azonosito); 00058

00061 void [setUtvonal](#_bookmark379)(const [Utvonal](#_bookmark297) &[utvonal](#_bookmark391)); 00062

00065 void [addKocsi](#_bookmark351)([Kocsi](#_bookmark215) &kocsi); 00066

00069 void [createKocsi](#_bookmark356)(int szekek); 00070

00073 void [addJegy](#_bookmark349)([Jegy](#_bookmark133) &jegy); 00074

00078 [Jegy](#_bookmark133) \*[getJegy](#_bookmark363)(int index) const; 00079

00084 bool [routeExists](#_bookmark373)(std::string indulo, std::string erkezo) const; 00085

00089 int [findAllomas](#_bookmark358)(std::string nev) const; 00090

00097 int [indulasiIdoKulonbseg](#_bookmark367)(std::string nev, int ora, int perc); 00098

00107 int [createJegy](#_bookmark352)(std::string indulo, std::string erkezo, int indulo\_ora = 0, int indulo\_perc = 0, double discountOrFee = 0, const std::string &tipus = "");

00108

00111 void [write](#_bookmark381)(std::ostream &os) const; 00112

00115 void [read](#_bookmark372)(std::istream &is); 00116

00118 [~Vonat](#_bookmark347)();

00119 };

00120

00121 #endif

# Tárgymutató

*∼*Allomas

Allomas, [10](#_bookmark18)

*∼*FelarasJegy

FelarasJegy, [16](#_bookmark67)

*∼*Ido

Ido, [19](#_bookmark90)

*∼*Jegy

Jegy, [27](#_bookmark142)

*∼*KedvezmenyesJegy KedvezmenyesJegy, [33](#_bookmark193)

*∼*Kocsi

Kocsi, [37](#_bookmark225)

*∼*Menetrend

Menetrend, [40](#_bookmark250)

*∼*Utvonal

Utvonal, [46](#_bookmark298)

*∼*Vonat

Vonat, [52](#_bookmark345)

addAllomas

Utvonal, [47](#_bookmark306) addJegy

Vonat, [52](#_bookmark345) addKocsi

Vonat, [52](#_bookmark345) addOra

Ido, [20](#_bookmark100) addPerc

Ido, [20](#_bookmark100) addVonat

Menetrend, [41](#_bookmark260)

Allomas, [7](#_bookmark7)

*∼*Allomas, [10](#_bookmark18)

Allomas, [8](#_bookmark10)

allomas\_azonosito, [13](#_bookmark48)

changeErkezes, [10](#_bookmark18)

changeIndulas, [10](#_bookmark18)

changeNev, [10](#_bookmark18)

erkezes, [13](#_bookmark48)

getErkezes, [11](#_bookmark27)

getErkezesOra, [11](#_bookmark27)

getErkezesPerc, [11](#_bookmark27)

getIndulas, [11](#_bookmark27)

getIndulasOra, [11](#_bookmark27)

getIndulasPerc, [12](#_bookmark38)

getNev, [12](#_bookmark38)

indulas, [13](#_bookmark48)

nev, [13](#_bookmark48)

operator=, [12](#_bookmark38)

read, [12](#_bookmark38)

write, [13](#_bookmark48)

allomas.cpp, [59](#_bookmark396)

allomas.h, [59](#_bookmark396) allomas\_azonosito

Allomas, [13](#_bookmark48) allomasok

Utvonal, [49](#_bookmark323) allomasok\_szama

Utvonal, [49](#_bookmark323)

ar

Jegy, [29](#_bookmark160)

cel\_ido

Jegy, [29](#_bookmark160) cel\_nev

Jegy, [29](#_bookmark160) changeAllomas

Utvonal, [47](#_bookmark306) changeErkezes

Allomas, [10](#_bookmark18) changeIndulas

Allomas, [10](#_bookmark18) changeNev

Allomas, [10](#_bookmark18) changeVonat

Menetrend, [41](#_bookmark260)

clear

Menetrend, [41](#_bookmark260) clone

FelarasJegy, [16](#_bookmark67)

Jegy, [27](#_bookmark142)

KedvezmenyesJegy, [33](#_bookmark193) createAllomas

Utvonal, [47](#_bookmark306) createJegy

Menetrend, [41](#_bookmark260)

Vonat, [52](#_bookmark345) createKocsi

Vonat, [53](#_bookmark354)

erkezes

Allomas, [13](#_bookmark48)

felar

FelarasJegy, [18](#_bookmark83)

FelarasJegy, [14](#_bookmark59)

*∼*FelarasJegy, [16](#_bookmark67)

clone, [16](#_bookmark67)

felar, [18](#_bookmark83)

FelarasJegy, [15](#_bookmark61), [16](#_bookmark67)

kiir, [16](#_bookmark67)

operator=, [17](#_bookmark76)

read, [17](#_bookmark76)

tipus, [18](#_bookmark83)

write, [17](#_bookmark76) findAllomas

Vonat, [53](#_bookmark354) foglalt\_szekek\_szama

Kocsi, [38](#_bookmark234)

getAllomas

Utvonal, [48](#_bookmark314) getAllomasokSzama Utvonal, [48](#_bookmark314)

getAr

Jegy, [27](#_bookmark142) getAzonosito

Kocsi, [37](#_bookmark225)

Vonat, [54](#_bookmark359) getErkezes

Allomas, [11](#_bookmark27) getErkezesOra

Allomas, [11](#_bookmark27) getErkezesPerc

Allomas, [11](#_bookmark27) getIndulas

Allomas, [11](#_bookmark27) getIndulasOra

Allomas, [11](#_bookmark27) getIndulasPerc

Allomas, [12](#_bookmark38) getJegy

Vonat, [54](#_bookmark359) getNev

Allomas, [12](#_bookmark38) getOra

Ido, [20](#_bookmark100) getPerc

Ido, [20](#_bookmark100) getSzekek\_szama

Kocsi, [37](#_bookmark225) getUtvonal

Vonat, [54](#_bookmark359) getVonat

Menetrend, [42](#_bookmark269) getVonatokSzama

Menetrend, [42](#_bookmark269)

helyszam

Jegy, [30](#_bookmark171)

Ido, [18](#_bookmark83)

*∼*Ido, [19](#_bookmark90)

addOra, [20](#_bookmark100)

addPerc, [20](#_bookmark100)

getOra, [20](#_bookmark100)

getPerc, [20](#_bookmark100)

Ido, [19](#_bookmark90)

kiir, [20](#_bookmark100)

operator!=, [21](#_bookmark111)

operator==, [21](#_bookmark111)

ora, [24](#_bookmark127)

perc, [24](#_bookmark127)

read, [21](#_bookmark111)

setIdo, [22](#_bookmark118)

setOra, [22](#_bookmark118)

setPerc, [22](#_bookmark118)

write, [22](#_bookmark118)

ido.cpp, [60](#_bookmark402)

ido.h, [60](#_bookmark402) indulas

Allomas, [13](#_bookmark48) indulasiIdoKulonbseg

Vonat, [54](#_bookmark359) indulo\_ido

Jegy, [30](#_bookmark171) indulo\_nev

Jegy, [30](#_bookmark171)

Jegy, [24](#_bookmark127)

*∼*Jegy, [27](#_bookmark142)

ar, [29](#_bookmark160)

cel\_ido, [29](#_bookmark160)

cel\_nev, [29](#_bookmark160)

clone, [27](#_bookmark142)

getAr, [27](#_bookmark142)

helyszam, [30](#_bookmark171)

indulo\_ido, [30](#_bookmark171)

indulo\_nev, [30](#_bookmark171)

Jegy, [26](#_bookmark136), [27](#_bookmark142)

jegy\_azonosito, [30](#_bookmark171)

kiir, [27](#_bookmark142)

kocsi\_szam, [30](#_bookmark171)

operator!=, [28](#_bookmark153)

operator=, [28](#_bookmark153)

operator==, [28](#_bookmark153)

read, [29](#_bookmark160)

vonat\_szam, [30](#_bookmark171)

write, [29](#_bookmark160)

jegy.cpp, [61](#_bookmark407)

jegy.h, [61](#_bookmark407) jegy\_azonosito

Jegy, [30](#_bookmark171) jegyek

Vonat, [57](#_bookmark383) jegyek\_szama

Vonat, [57](#_bookmark383)

kedvezmeny

KedvezmenyesJegy, [35](#_bookmark209)

KedvezmenyesJegy, [31](#_bookmark185)

*∼*KedvezmenyesJegy, [33](#_bookmark193)

clone, [33](#_bookmark193)

kedvezmeny, [35](#_bookmark209)

KedvezmenyesJegy, [32](#_bookmark187), [33](#_bookmark193)

kiir, [33](#_bookmark193)

operator=, [34](#_bookmark202)

#### TÁRGYMUTATÓ 85

kiir

read, [34](#_bookmark202)

tipus, [35](#_bookmark209)

write, [34](#_bookmark202)

FelarasJegy, [16](#_bookmark67)

Ido, [20](#_bookmark100)

Jegy, [27](#_bookmark142)

KedvezmenyesJegy, [33](#_bookmark193)

Menetrend, [42](#_bookmark269)

Utvonal, [48](#_bookmark314)

Ido, [21](#_bookmark111)

Jegy, [28](#_bookmark153) operator=

Allomas, [12](#_bookmark38)

FelarasJegy, [17](#_bookmark76)

Jegy, [28](#_bookmark153)

KedvezmenyesJegy, [34](#_bookmark202)

Utvonal, [48](#_bookmark314)

Vonat, [55](#_bookmark368) operator==

Kocsi, [35](#_bookmark209)

*∼*Kocsi, [37](#_bookmark225)

foglalt\_szekek\_szama, [38](#_bookmark234)

getAzonosito, [37](#_bookmark225)

getSzekek\_szama, [37](#_bookmark225)

Kocsi, [36](#_bookmark216)

kocsi\_azonosito, [38](#_bookmark234)

read, [37](#_bookmark225)

reserveHely, [37](#_bookmark225)

szekek\_szama, [38](#_bookmark234)

tele, [38](#_bookmark234)

write, [38](#_bookmark234)

kocsi.cpp, [63](#_bookmark414)

kocsi.h, [63](#_bookmark414) kocsi\_azonosito

Kocsi, [38](#_bookmark234) kocsi\_szam

Jegy, [30](#_bookmark171)

kocsik

ora

perc

read

Ido, [21](#_bookmark111)

Jegy, [28](#_bookmark153)

Ido, [24](#_bookmark127)

Ido, [24](#_bookmark127)

Allomas, [12](#_bookmark38)

FelarasJegy, [17](#_bookmark76)

Ido, [21](#_bookmark111)

Jegy, [29](#_bookmark160)

KedvezmenyesJegy, [34](#_bookmark202)

Kocsi, [37](#_bookmark225)

Menetrend, [43](#_bookmark276)

Serializable, [44](#_bookmark284)

Utvonal, [48](#_bookmark314)

Vonat, [55](#_bookmark368)

Vonat, [57](#_bookmark383) kocsik\_szama

Vonat, [57](#_bookmark383)

main

main.cpp, [64](#_bookmark420)

main.cpp, [64](#_bookmark420)

main, [64](#_bookmark420)

Menetrend, [39](#_bookmark246)

*∼*Menetrend, [40](#_bookmark250)

addVonat, [41](#_bookmark260)

changeVonat, [41](#_bookmark260)

clear, [41](#_bookmark260)

createJegy, [41](#_bookmark260)

getVonat, [42](#_bookmark269)

getVonatokSzama, [42](#_bookmark269)

kiir, [42](#_bookmark269)

Menetrend, [40](#_bookmark250)

read, [43](#_bookmark276)

removeVonat, [43](#_bookmark276)

vonatok, [44](#_bookmark284)

vonatok\_szama, [44](#_bookmark284)

write, [43](#_bookmark276)

menetrend.cpp, [65](#_bookmark425)

menetrend.h, [65](#_bookmark425)

nev

Allomas, [13](#_bookmark48) operator!=

removeAllomas

Utvonal, [49](#_bookmark323) removeVonat

Menetrend, [43](#_bookmark276) reserveHely

Kocsi, [37](#_bookmark225) routeExists

Vonat, [55](#_bookmark368)

Serializable, [44](#_bookmark284)

read, [44](#_bookmark284)

write, [45](#_bookmark293)

serializable.h, [66](#_bookmark432) setAzonosito

Vonat, [56](#_bookmark375) setIdo

Ido, [22](#_bookmark118) setOra

Ido, [22](#_bookmark118) setPerc

Ido, [22](#_bookmark118) setUtvonal

Vonat, [56](#_bookmark375) szekek\_szama

Kocsi, [38](#_bookmark234)

tele

Kocsi, [38](#_bookmark234)

test

test.cpp, [66](#_bookmark432)

test.h, [73](#_bookmark440)

test.cpp, [66](#_bookmark432)

test, [66](#_bookmark432)

test.h, [73](#_bookmark440)

test, [73](#_bookmark440)

tipus

FelarasJegy, [18](#_bookmark83)

KedvezmenyesJegy, [35](#_bookmark209)

Utvonal, [45](#_bookmark293)

*∼*Utvonal, [46](#_bookmark298)

addAllomas, [47](#_bookmark306)

allomasok, [49](#_bookmark323)

allomasok\_szama, [49](#_bookmark323)

changeAllomas, [47](#_bookmark306)

createAllomas, [47](#_bookmark306)

getAllomas, [48](#_bookmark314)

getAllomasokSzama, [48](#_bookmark314)

kiir, [48](#_bookmark314)

operator=, [48](#_bookmark314)

read, [48](#_bookmark314)

removeAllomas, [49](#_bookmark323)

Utvonal, [46](#_bookmark298)

utvonal\_azonosito, [49](#_bookmark323)

write, [49](#_bookmark323) utvonal

Vonat, [57](#_bookmark383)

utvonal.cpp, [80](#_bookmark445)

utvonal.h, [80](#_bookmark445) utvonal\_azonosito

Utvonal, [49](#_bookmark323)

Vonat, [50](#_bookmark334)

*∼*Vonat, [52](#_bookmark345)

addJegy, [52](#_bookmark345)

addKocsi, [52](#_bookmark345)

createJegy, [52](#_bookmark345)

createKocsi, [53](#_bookmark354)

findAllomas, [53](#_bookmark354)

getAzonosito, [54](#_bookmark359)

getJegy, [54](#_bookmark359)

getUtvonal, [54](#_bookmark359)

indulasiIdoKulonbseg, [54](#_bookmark359)

jegyek, [57](#_bookmark383)

jegyek\_szama, [57](#_bookmark383)

kocsik, [57](#_bookmark383)

kocsik\_szama, [57](#_bookmark383)

operator=, [55](#_bookmark368)

read, [55](#_bookmark368)

routeExists, [55](#_bookmark368)

setAzonosito, [56](#_bookmark375)

setUtvonal, [56](#_bookmark375)

utvonal, [57](#_bookmark383)

Vonat, [51](#_bookmark337)

vonat\_azonosito, [57](#_bookmark383)

write, [56](#_bookmark375)

vonat.cpp, [81](#_bookmark451)

vonat.h, [81](#_bookmark451) vonat\_azonosito

Vonat, [57](#_bookmark383) vonat\_szam

Jegy, [30](#_bookmark171) vonatok

Menetrend, [44](#_bookmark284) vonatok\_szama

Menetrend, [44](#_bookmark284)

write

Allomas, [13](#_bookmark48)

FelarasJegy, [17](#_bookmark76)

Ido, [22](#_bookmark118)

Jegy, [29](#_bookmark160)

KedvezmenyesJegy, [34](#_bookmark202)

Kocsi, [38](#_bookmark234)

Menetrend, [43](#_bookmark276)

Serializable, [45](#_bookmark293)

Utvonal, [49](#_bookmark323)

Vonat, [56](#_bookmark375)