

Reptéri csomagok

specifikáció

A program egy reptér csomagszállító rendszerét szimulálja. A modellben vannak csomagok, amelyek bizonyos csomópontokra érkeznek be. Innen futószalagok továbbítják azokat az úticélukhoz, amely bármely más csomópont lehet. Egy központi vezérlőrendszer adja ki az utasításokat, hogy a csomópontok hova, mely futószalagon küldjék tovább a csomagokat.

A program képes fájlból beolvasni a reptér rendszerét, valamint a csomagokat. Itt bemenetként a következőket várja soronkénti, vesszővel tagolt felsorolásban az alábbi sorrendben:

- Csomópontok neve (angol abc 1db nagy betűje) (egy sor egy db)
- Futószalag azonosítója (pozitív egész szám), kezdő csomópont neve, vége csomópont neve (egy sor egy db)
- Csomag azonosítója (pozitív egész szám), kiinduló csomópont neve, cél csomópont neve (egy sor egy db)

Ezeket az adatokat be lehet vinni manuálisan is a konzolról a megfelelő menüpont segítségével. A programmal tetszőleges mennyiségű csomópont, futószalag és csomag létrehozható.

A program lefutásához az adatoknak helyesnek kell lenni. Ennek feltételei:

- Minden csomóponthoz kapcsolódjon legalább egy ott végződő és egy ott kezdődő futószalag, annak érdekében, hogy oda csomagok érkezhessenek és onnan indulhassanak.
- A csomópontok rendszerének összefüggőnek kell lennie (összefüggő gráf). Tehát bármelyik csomópontból bármelyik másik elérhető legyen.
- Egy futószalag nem végződhet ott, ahol indul. Ezeket a program automatikusan törli.
- Két csomópont között közvetlen csak egy oda/vissza futószalag lehet. Ha már létezik A-->B futószalag, akkor a program nem fogja létrehozni azt újra. (Egyszerű gráf)

Ezen kívül alapvető feltétel, hogy a platform ahol a program fut rendelkezzen C++ fordítóval.

Ha megfelelő bemenet érkezik, a program elkezd szimulálni a modellt. Ezt a konzolon végig lehet követni, ahol a program írja, hogy éppen az adott pillanatban melyik csomóponton mely csomagok vannak (azonosító szerint). Így jól látható ahogy egy csomag a lehető legrövidebb úton eljut az úticéljához. Végezetül a program kiírja (konzolra és fájlba is) a végleges állapotot a csomópontokkal és a csomagokkal.

terv

A programot a fő függvény irányítja. Ez felel az adatok beolvasásáért, kiírásáért (mindezek fájlban tárolt és konzolos megjelenítésű megoldásaiért), valamint ez indítja el a vezérlőközpont működését.

Vezérlőközpont

A vezérlőközpont class felel azért, hogy a csomagok a csomópontokról eljussanak a célállomásaikra.

Privát adattagjai, függvényei:

- csomópontok: egy csomagokat tároló adatszerkezet.
- útvonalak: egy útvonalakat tároló adatszerkezet.
- bfsArray: a bfs osztály egyedeit tárolja.
- makeBfs (függvény): bemenete egy char típusú adat a nevével annak a csomópontnak, amelyből futtatja majd a BFS algoritmust. Létrehoz egy bfs típusú adatot, amelyhez létrehozza a bfsAdat típusú adattagokat. Ezt az adatot eltárolja a bfsArray-ben.
- sendParancs (függvény): bemenete egy int típusú csomópontID, csomagID és futószalagID. Meghívja az adott csomópont send parancsát, hogy az elküldje a csomagot a megfelelő futószalaggal.
- vanÚtvonala (függvény): bemenete egy int típusú csomagID. A függvény megállapítja, hogy az adott csomagnak van-e már útvonala.

Publikus függvénye:

- run: ez a függvény gyakorlatilag a fő program, ami kezeli a csomagokat, útvonalakat és kiküldi a csomópontoknak az utasításokat

Útvonal

Az útvonal class egy adott csomag útvonalát tárolja.

Privát adattagjai:

- csomagID: egy adott csomag int típusú azonosítóját tárolja
- állomások: egy char típusú adatokat tároló adatszerkezet, amely a csomag állomásainak neveit tárolja, beleértve a kezdő és végpontokat. Ezeken a csomópontokon kell majd, hogy a csomag keresztül menjen, hogy eljusson a céljához.

Publikus függvényei:

- addÁllomás: egy char típusú adatot kér be, egy csomópont nevével, amit eltárol az állomások adatszerkezetében.
- nextÁllomás: bekéri az aktuális állomás nevét egy char típusú adatban, majd visszaadja, hogy az útvonalon mely a következő állomás neve (szintén char).
- getID: visszaadja int típusként a csomag azonosítóját

Csomópont

Privát adattagok:

- név: char típusú, tárolja a csomópont nevét
- csomagok: egy csomagokat tároló adatszerkezet
- futószalagok: a csomópontból induló futószalagokat tároló adatszerkezet

Publikus függvényei:

- getNév: visszaadja a csomópont nevét (char).
- getNextCsomag: visszaadja a listában következő csomagot (csomag típus).
- send: bekéri egy csomag, illetve egy futószalag azonosítóit, majd az adott azonosítójú csomagot átteszi a megadott futószalagra

Csomag

Publikus adattagjai:

- ID: int típusú, a csomag azonosítója
- Start: char típusú, a csomópont neve, ahonnan indul a csomag
- Cél: char típusú, a csomópont neve, ahova el akar jutni a csomag

Publikus függvényei:

- Kész: bool típusú tér vissza, megadja, hogy a csomag megérkezett-e a céljára
- getID: int típusú tér vissza, megadja a csomag azonosítóját
- getStart: visszaadja a csomag char típusú kezdőpontját
- getCél: visszaadja a csomag char típusú célállomását

Futószalag

Publikus adattagjai:

- ID: int típusú, a futószalag azonosítója
- Start: char típusú, a csomópont neve, ahonnan indul a futószalag

- Vég: char típusú, a csomópont neve, ahova vezet a futószalag
- Csomagok: csomag típusú adatszerkezet, amely a futószalagon lévő csomagokat tárolja

Publikus függvényei:

- getID: int típusú tér vissza, a futószalag azonosítóját adja meg
- getStart: char típusú visszaadja a futószalag kezdőpontját
- getVég: char típusú visszaadja a futószalag végpontját
- addCsomag: bemenetként megkap egy csomagot, amit eltárol
- removeCsomag: leveszi a futószalagról a legelső csomagot és ezt adja vissza (csomag típus)

bfs

Privát adattagok:

- csomópont: char típusú tárolja a csomópont nevét, ahonnan futott a BFS algoritmus
- tábla: bfsAdat típusú tároló, amely az algoritmus által elkészített adatokat tárolja

Publikus függvényei:

- getCsomópont: visszaadja char típusú a csomópont nevét
- makeÚtvonal: bekéri char típusú a csomópont nevét, ahova el akarunk jutni abból a csomópontból, ahonnan futott a BFS, majd az adatai alapján visszaadja az utat.
Visszatérési értéke egy char tároló, ami tartalmazza a csomópontokat az út során.

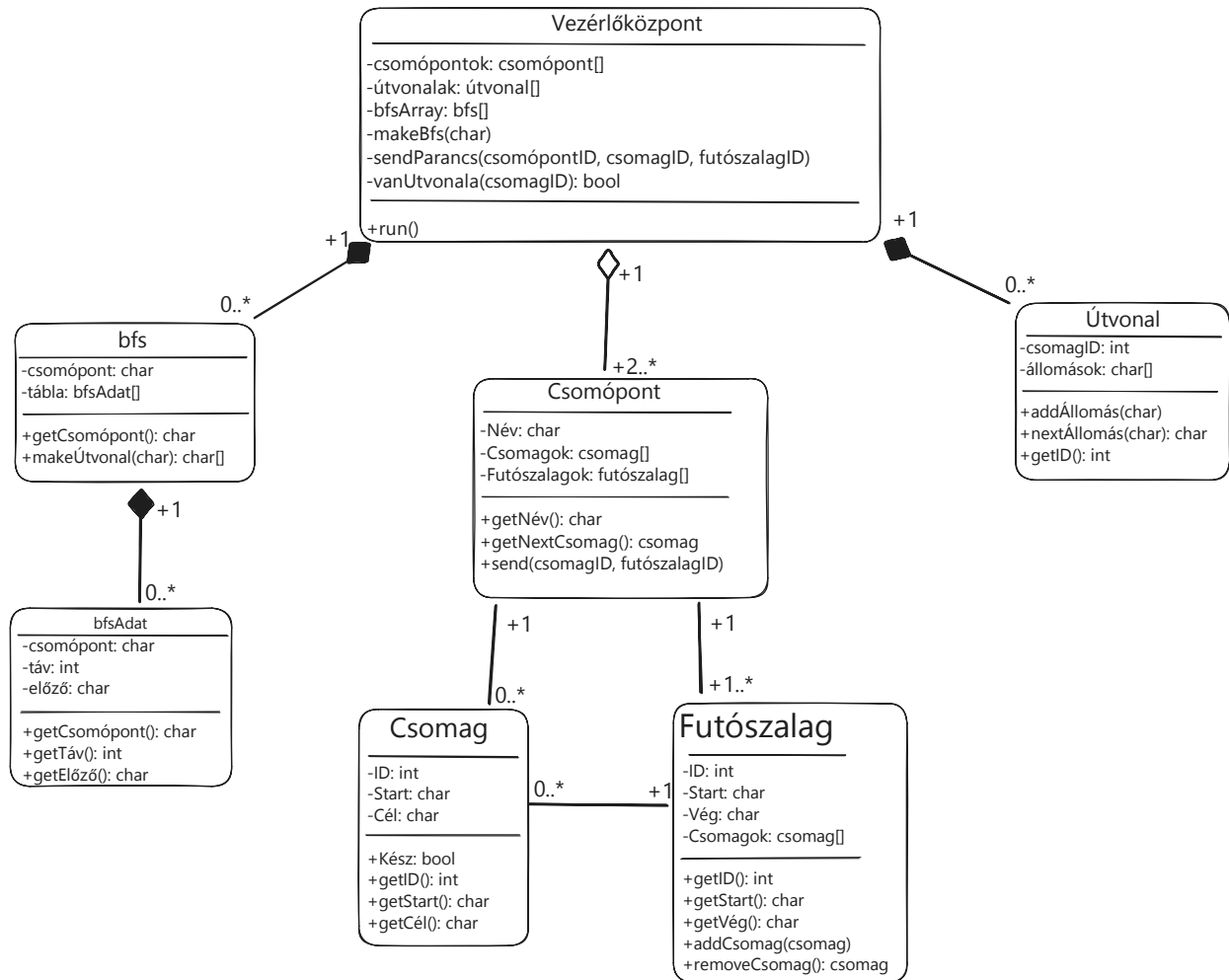
bfsAdat

Privát adattagok:

- csomópont: char típus, ami tárolja azt a csomópont nevét, ahova eljutottunk
- táv: int típusú tárolja az addig megtett távot a kezdeti csomóponttól (egy futószalag = 1 táv)
- előző: char típusú tárolja annak a csomópontnak a nevét, ahonnan ebbe a csomópontba jött az algoritmus

Publikus függvényei:

- getCsomópont: visszaadja char típusú a csomópont nevét
- getTáv: visszaadja int típusú a távot
- getElőző: visszaadja char típusú az előző csomópont nevét



Botrágyi Gergő