Rendszerterv

Tartalom

[Feladatkörök 2](#_Toc469584658)

[Specifikáció 2](#_Toc469584659)

[Tároló mátrix: 3](#_Toc469584660)

[Fejlesztés összetétele: 4](#_Toc469584661)

[Grafikus felület 4](#_Toc469584662)

[Adat tárolás 4](#_Toc469584663)

[Lehetséges bevitt adatok: 5](#_Toc469584664)

[Adat tárolás megvalósítása 5](#_Toc469584665)

[Kalkuláció elvégzése 7](#_Toc469584666)

[Program logikai részének felépítése: 8](#_Toc469584667)

[A függvények a következő sorrendben hívják meg egymást: 8](#_Toc469584668)

[Végeredmény megjelenítése 9](#_Toc469584669)

[Java programozási nyelv: 10](#_Toc469584670)



## Feladatkörök

**Vető Roland**: projectmenedzser

**Kondor Gergely**: grafikai kiemelt szakértő

**Végh László**: programozó

**Balogh Tamás:** adatkezelő

## Részletes feladatkörök:

**Vető Roland** (projectmenedzser): Feladatok kiosztása a project résztvevői között. Állandó kapcsolattartás a megrendelővel. Alkalmazottak munkavégzésének felügyelete. Kritikus döntések meghozatala.

**Kondor Gergely** (grafikai kiemelt szakértő, programozó): Fő feladatköre a grafikai feladatok ellátása. Szükség esetén programozói feladatok ellátása.

**Végh László** (programozó): Programkód megírása, ellenőrzése, javítása. Logikai feladatok megoldása.

**Balogh Tamás** (adatkezelő): A project során keletkező adatok kezelése eltárolása. Szükséges dokumentumok elkészítése, megosztása.

## Specifikáció

A megbízás egy szálitmányozással foglalkozó céghez köthető, amely egy telephellyel és csomagátvételi pontokkal rendelkezik.

A vállalkozás szeretné a telephelyről eljuttatni a csomagokat az átvételi pontokra a lehető legrövidebb útvonalon.

A program az átvételi pontokat egy gráfként kezeli.

A gráf csúcsai megfelelnek egy-egy átvételi pontnak.

A gráfra igaz, hogy ha egy átvételi pontból el tudunk jutni egy másik átvételi pontra akkor a gráfban a pontoknak megfelelő csúcsok között húzódik egy él.

Az élek súlya megfelel a cég által választott út hosszának.

Két csúcs között maximum két él futhat.

Amennyiben két él fut két különböző csúcs között abban az esetben a A-ból B-be vezető út nem egyezik meg B-ből A-ba vezető úttal.

A csúcsból önmagába nem vezethet út.

A csúcsok súlya nem lehet negatív érték.

Az élek súlyának tárolására egy mátrix szolgál amely az esetünkben egy két dimenziós tömb.

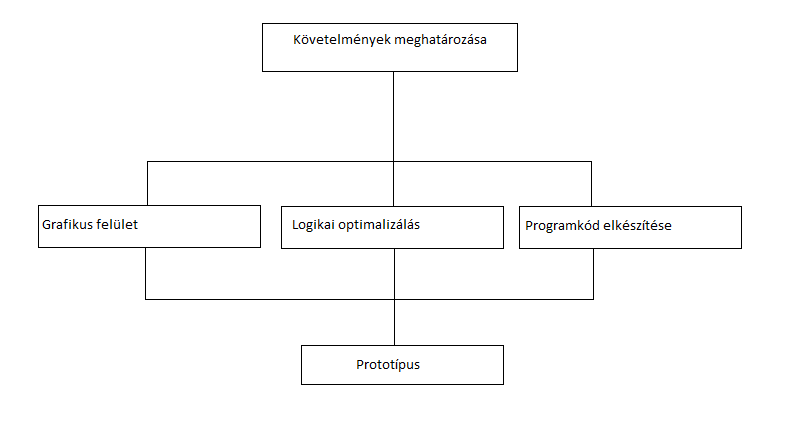
A feladat teljesítése során meg kell valósítani egy olyan Java kódot, amely megfelel a követelményeknek.

A fejlesztés során folyamatosan prototípusokat készítünk míg végül megkapjuk a megfelelő változatot.

A program tartalmaz egy osztályt, amely példányosításával lehet majd meghatározni a legrövidebb utat a csúcsok között.

## Tároló mátrix:

# Fejlesztés összetétele:



## Grafikus felület

Az adatok bevitelére szánt felület a Netbeans program nyújtott lehetőségek alapján fog elkészülni. A felületen bevitt adatokat két nagyobb részre oszthatjuk a gráf csúcsai számának meghatározása, amelyet egy beviteli mező segítségével határozhat meg. A másik nagyobb rész a csúcsok között húzódó élek súlyának meghatározása , amely szintén egy beviteli mező segítségével paraméterezheti a felhasználó.

## Adat tárolás

A bevitel után egy mátrixban tárolódik el a bevitt információ. A mátrix dimenzió számát a felhasználó által megadott csúcsok száma határozza meg. A mátrix egyes értékei az élek súlya határozza meg(pl: A mátrix[1][2] eleme az 1 csúcsból a 2 csúcsba húzódó él súlya)

## Lehetséges bevitt adatok:

## Adat tárolás megvalósítása

A mátrix adatstruktúra megvalósításra egy két dimenziós tömb szolgál.  A két dimenziós tömb feltöltése a grafikus felületen történik, a tömb hossza dinamikusan változik A megadott csúcsok számának függvényében. A megrendelő nem szeretné eltárolni a számított adatokat, mivel nem tudja biztosítani ezen adatok védelmét külső személyekkel szemben , ezért az adatok csak a futás közben történek eltárolásra program futása után minden megadott adat törlésre kerül.

## Kalkuláció elvégzése

Ahhoz, hogy megkapjuk a legrövidebb utat a program végigmegy ezen a tömbön. A tömb vizsgálata összetett for ciklusok segítségével történik, valamint a ciklusok futtatása során az értékek kiszámítása függvények segítségével valósul meg. A programban használt algoritmus egy általunk készített rekurzív megoldás, amely, a nyers erőn alapszik rekurzívan felépíti a lehetséges útvonalat futás közben folyamatosan csak az aktuális legrövidebb történik eltárolásra. Majd az algoritmus eredményeként megkapjuk a legrövidebb utat valamint ennek az útnak súlyát. A program legrövidebb út meghatározása alatt számolja ki a legrövidebb út súlyát, hogy a futási idő a lehető legrövidebb legyen.

## Program logikai részének felépítése:

1. Költség függvény: A gráfban húzódó legrövidebb út éleinek súlyával tér vissza.
2. Min függvény: Az aktuálisan legrövidebb csúccsal tér vissza.
3. ÚtManager függvény: A min függvény meghívásával meghatározza a legrövidebb útra eső csúcsok halmazát, sorrendjét.
4. Végeredmény függvény: Összefogja az egész programot valamint kiszámítja a végső költséget.

## A függvények a következő sorrendben hívják meg egymást:

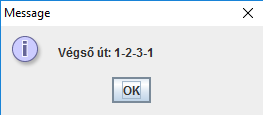
## Végeredmény megjelenítése

A végeredmény megjelenésé a grafikus felületen történik. A számítások végén megjelenik a felhasználó számára a legrövidebb út összetétele valamint ennek a költsége. A vevő igényeinek megfelelve a program sallangmentesen csak arra a feladatra nyújtson megoldást amelyet a vevő közvetített csapatunk felé. A vevő kikötötte, hogy miután megadták ”a gráf „ csúcsainak számát illetve a csúcsok közötti él súlyát kizárólag a gráfban húzódó legrövidebb útra illeszkedő csúcsok halmazát valamint ennek a csúcsnak a súlyát jelenítsük csak meg. Az erőforrásbeli korlátok miatt a megrendelő kihangsúlyozta, hogy a térképes megvalósítást majd csak a jövőben szeretnék megrendelni ,ha már bevált az általunk készített verzió.

Kiír függvény felépítése:



A kiír, függvény az adatok megjelenítésére szolgál.



## Java programozási nyelv:

A feladat megvalósítása során Java programozási nyelvet használunk. A vevő külön kérése volt, hogy egy strukturált objektum orientált programozási nyelvet alkalmazzuk, amely már kellően elterjedt a köztudatban alkalmas a feladat megvalósítására valamint kompatibilis az eddig általunk használt programokkal. A csapatunk megvizsgálta a cég által használt programokat valamint, hogy a feladat megvalósítás szempontjából melyik lenne a legmegfelelőbb programozási nyelv. Az információ begyűjtése után meghoztuk a végső döntést, hogy melyik programozási nyelven fogjuk megvalósítani a felkérést. A csapat egyhangúan egyetértett abban, hogy a Java programozási nyelv az amely minden feltételnek megfelel a feladat megvalósításához.