

Szoftvertervezés és -fejlesztés II. Házi Feladat

Gráf I.

Készítse el az alábbi feladat megoldásához szolgáló, saját implementálású gráf adatszerkezetet, elsősorban szomszédsági lista, másodsorban (további gyakorlásként) pedig csúcsmátrix reprezentációval.

```
class Graph
```

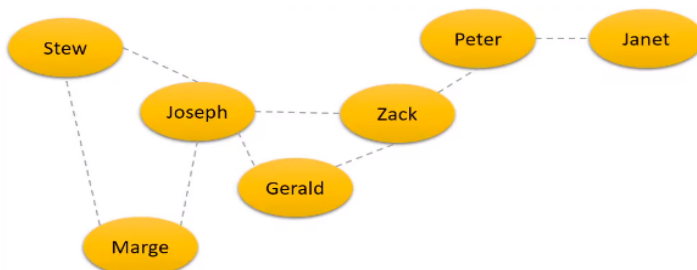
Az adatszerkezethez hozza létre a szükséges osztályt a tanult módon és biztosítsa a következőket:

- legyen generikus a T típusra nézve
- legyen külső metódus megadására lehetőség, amely a feldolgozás módját fogja jelenteni
 - ebből adódóan az adatszerkezetből Console metódusokat használni **tilos**
 - ezt a funkciót a tanultak alapján valósítsa meg **metódusreferencia segítségével**
 - `void ExternalProcessor(string item)`
- tesztelje a működést
 - egyrészt a kívülről átadott Console.WriteLine metódussal,
 - másrészt pedig hozzon létre egy saját metódust amely eleget tesz a metódus szignatúrának (pl. `void Proc(string item)`) aminek segítségével az adott feldolgozandó elemet ne a konzolra írja ki, hanem egy fájlhoz fűzze hozzá mindig
- írja meg a szélességi- valamint mélységi bejárásokat
 - `void BFS(---)`
 - `void DFS(---)`
 - a paraméterek megválasztása a feladat része

```
class Person
```

A személyeket reprezentáló objektumokhoz hozza létre az entitás osztályt *Person* néven, amelyben 1 db *Name* írható/olvasható tulajdonságot helyezzen el. Az osztályban írja felül a *ToString* metódust, hogy a metódusreferenciát megfelelő módon tudja használni.

Miután az adatszerkezet elkészült, a mellékelt ábrán látható személyek közötti kapcsolati hálót kell elkészíteni, a megfelelő csúcsok és élek beállításával.



Dolgozza fel az elemeket DFS és BFS alapján, melyeket szükség szerint írjon meg az osztályban, felhasználva a metódus referenciát (melyen keresztül tudja kívülről biztosított metódust meghívni)!

```
class Graph
```

Egészítse ki az adatszerkezet osztályt egy eseménnyel (mely alapját szolgáltassa a `void GraphEventHandler<T>(object source, GraphEventArgs<T> geargs)` delegált), amely legyen elsütve egy él felvitelét követően. Az eseményben legyen benne, hogy melyik A és B csúcs között sikerült az él felvenni. Ehhez használjon *GraphEventArgs* osztályt. Mind az esemény argumentum osztály, mind a delegált és a hozzá tartozó esemény **generikus legyen** szintén a T típusra nézve! A Program részben hozzon létre egy metódust melyet iratkoztasson fel az adatszerkezet eseményére.

```
class Graph
```

Egészítse ki a BFS bejárást, és készítsen egy olyan algoritmust amely segítségével meg tudjuk mondani, hogy adott X személy hányad fokú ismerettségben áll adott Y személlyel. Az ismerettség fokát a két személy közötti ismerősök száma jelenti, egészen pontosan a rajtuk keresztül megtett legrövidebb út hossza (élek száma). Például az ábrából kiindulva: Janet harmadfokú ismerettségben áll Gerald-dal, mert Peter-en majd Zack-en keresztül van kapcsolat közöttük (1. Janet > Peter; 2. Peter > Zack; 3. Zack > Gerald). Az is megfelelő ha az X és Y közötti út kerül kiírásra.

A feladat megoldása során elfogadott és használható a beépített List valamint Queue osztály!