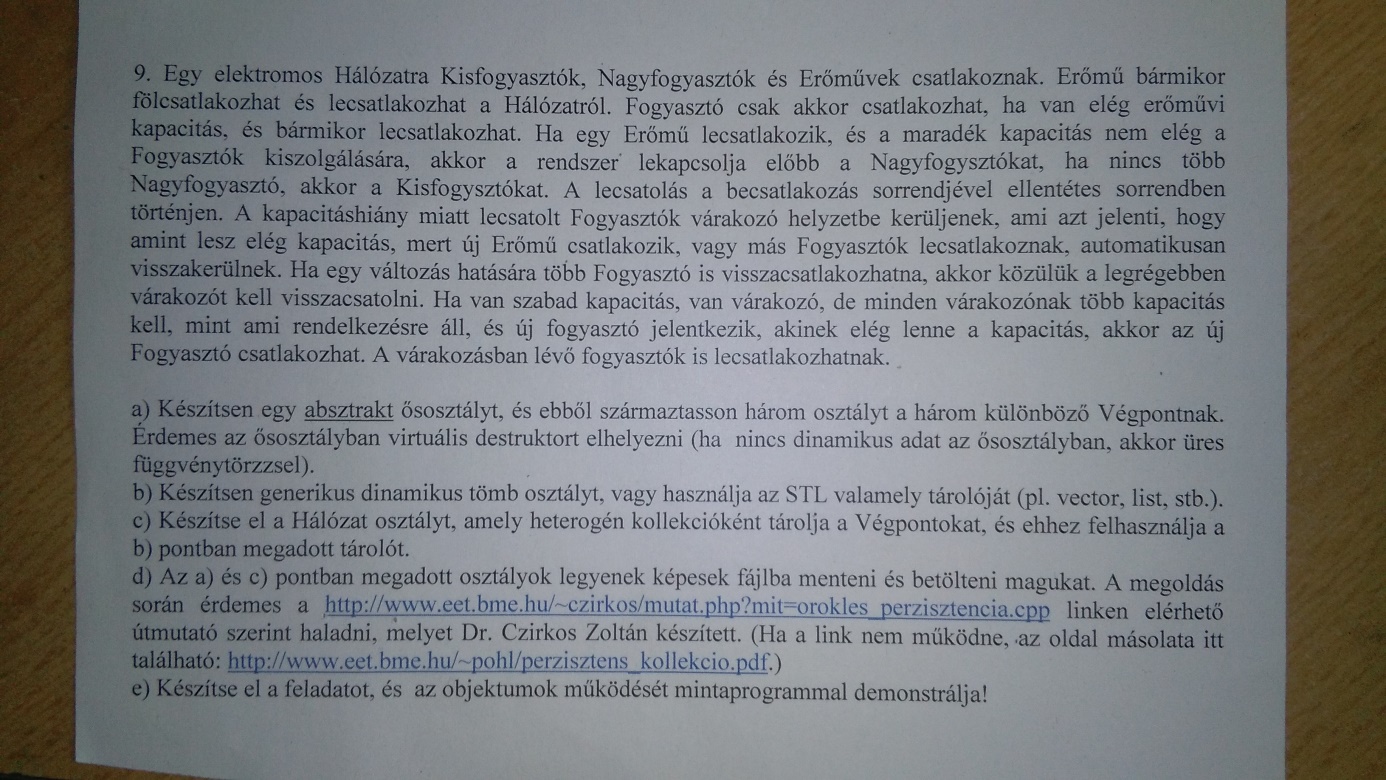
Programozás alapjai 2. nagy HF dokumentációja

Az általam készített program egy elektromos hálózat működését modellezi. Vannak fogyasztók, és erőművek, és ezek fel-le csatlakozhatnak a hálózatra. Persze mindig csak úgy, hogy az erőművek kapacitásának ki kell szolgálnia a fogyasztók által elhasznált teljesítményt.

A feladat leírása:



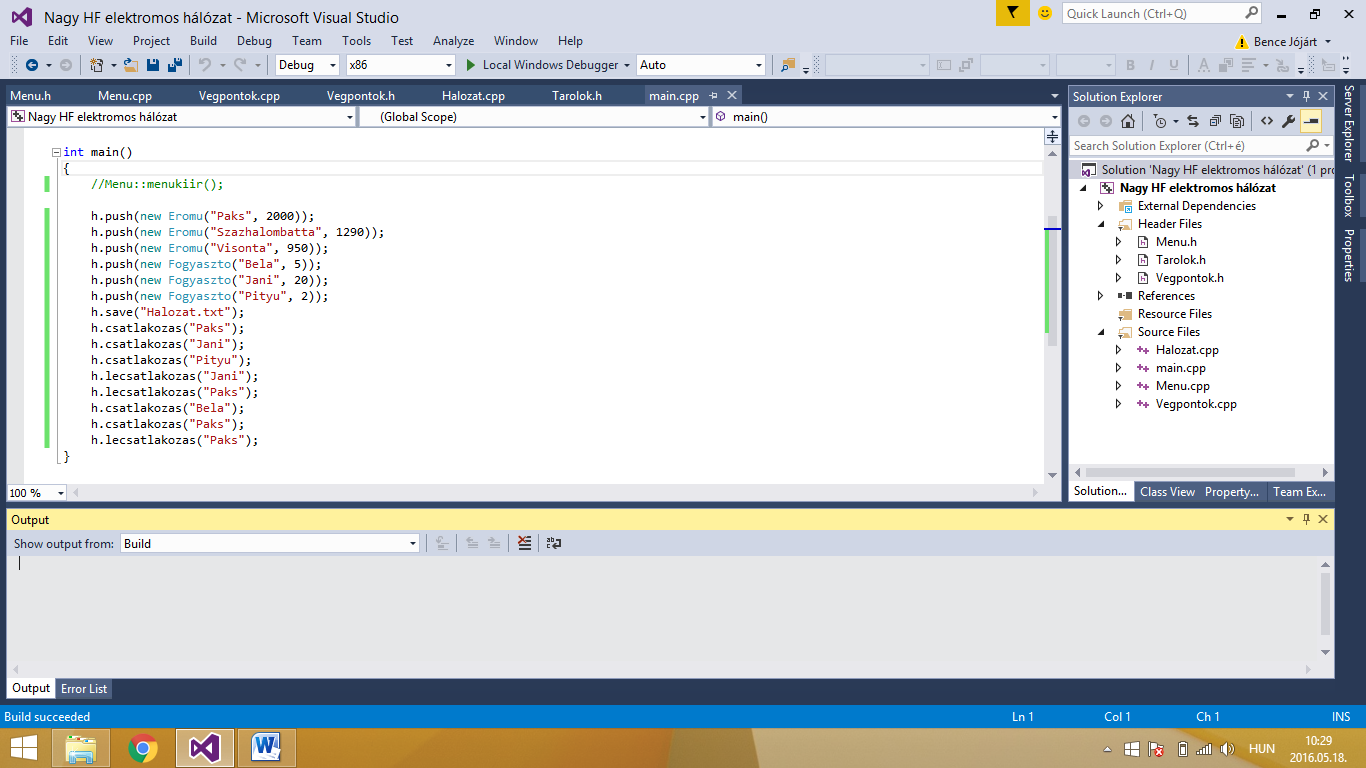
A feladatot annyiban módosítottam, hogy a kis- és nagyfogyasztókat nem különböztettem meg, mert csak elenyésző mértékben térnek el egymástól.

Először a végpontok osztályait készítettem el. A végpont ősosztályból származtatom a fogyasztót és az erőművet, hogy együtt lehessen őket tárolni egy heterogén kollekcióban. Minden végpontnak van neve, teljesítménye, illetve státusza, hogy épp be van csatlakozva a hálózatba, le van csatlakozva a hálózatról, vagy várólistán van (várólista: ha valaki csatlakozni szeretne, de nincs elég kapacitás, akkor az ide kerül). A végpont osztály egy absztrakt ősosztály, tisztán virtuális függvényei vannak. Van egy kiíró függvénye, ez az egyetlen, ami nem virtuális, mert ugyanaz a függvény az erőműnél és a fogyasztónál is. A fajlban\_nev() függvény a fájlba írást és a beolvasást segíti: az erőműveknél ”eromu” stringet ad vissza, a fogyasztóknál ”fogyaszto”-t. Az osztály két legfontosabb tagfüggvénye a csatlakozás, ill. a lecsatlakozás, ezek természetesen tisztán virtuálisak. Az alosztályok ezen függvényei a Vegpontok.cpp-ben vannak megírva, minden más a Vegpontok.h-ban. Minden csatlakozás és lecsatlakozás függvény 3 if ágra oszlik, attól függően, hogy a végpont státusza eddig mi volt. Ezen függvényekben történik meg a feladatban kiírt feltételek vizsgálata: ha például csatlakozott egy erőmű, akkor megnézi a függvény, hogy tudunk e valakit csatlakoztatni a várólistáról. Ha olyat akarunk csinálni, ami hibás, pl. csatlakoztatni akarunk egy erőművet, akkor a program hibát dob. Ez az egész hálózat pedig egy globális, Halozat típusú h nevű változóban tárolódik el.

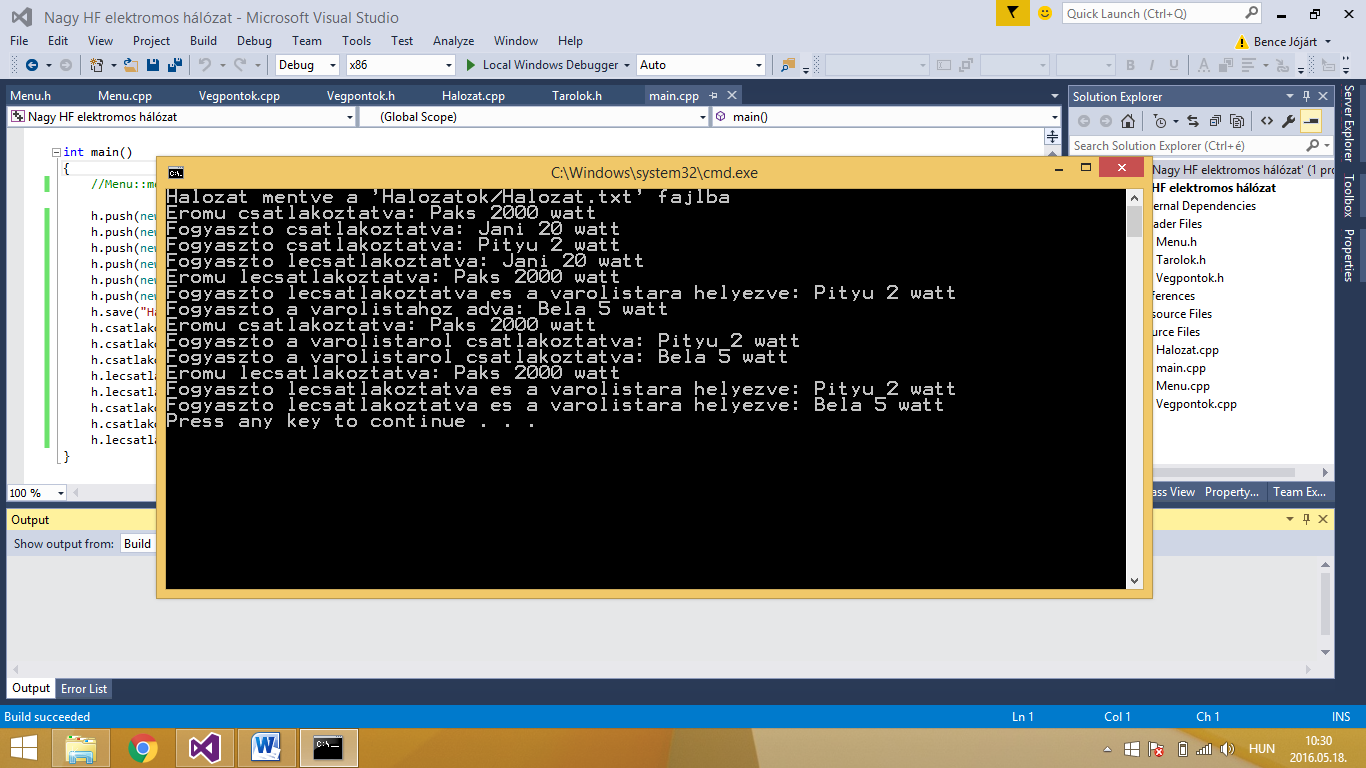
A tárolás alapvetően egy template<class T> típusú ősosztályból származtatott generikus dinamikus tömbökben zajlik. Ennek az osztálynak a neve Tarolo. Két tagváltozója van: egy T-re mutató pointer, ami a tömb elejére mutat, illetve egy int n, ami a darabszámot mondja meg. Ezek privát tagváltozók. Tagfüggvényként van neki alapértelmezett konstruktora, másolókonstruktora, egyenlőség operátora, ill. virtuális destruktora. Ezeken kívül van push függvénye (elem betétele), pop függvénye (adott sorszámú elem kivétele), operator[] függvénye, az egyszerűbb és kényelmesebb elérhetőség miatt. Van még listabankeres függvénye, ami egy adott végpont sorszámát adja vissza. Van még virtuális kiíró függvénye, illetve clear függvénye, amit a destruktor hív meg, ez szabadítja fel a tömböt.

Ebből a sablon tároló ősosztályból származtatok 3 osztályt. A Halozat osztály tárolja el az összes erőművet és fogyasztót. Ezt egy globális tagváltozón keresztül éri el a program. Származik még az ősosztályból a várólista, ill. a becsatlakozott hálózatokat nyilvántartó lista. Ezek közül a legfontosabb a Halozat osztály: ez tartalmaz tagváltozóként egy Varolista osztályt, illetve egy Becsatlakozott osztályt. A Halozat tegváltozója még ezen kívül a kapacitas, ami az erőművek teljesítménye mínusz a fogyasztók teljesítménye. Ez soha sem lehet negatív. Ezenken kívül még 4 tagfüggvénye van az osztálynak, ezek a Halozat.cpp-ben vannak megírva: a save elmenti a hálózatot fájlba, a load betölti. A csatlakozás ill. a lecsatlakozás függvény azt csinálja, hogy megkeresi a paraméterként kapott nevű végpontot a hálózatban, majd meghívja ennek csatlakozás ill. lecsatlakozás függvényét.

A feladathoz készült main mintaprogram a következő:



Lefuttatva:



A programhoz készül egy menü is, ami jelenleg átalakítás alatt van, félkész állapotban. Ennek segítségével a felhasználó több fájlba tud menteni különböző hálózatokat, módosítani tudja a meglévőket, újakat tud létrehozni, illetve szimulációkat tud végezni a betöltött hálózattal, a console-ról. A terv a menüre a következő:

