

# HÁZI FELADAT

Programozás alapjai 2.

Pontosított feladatspecifikáció

Németh Dániel

FTYYJR

2019. március 31.

---

## TARTALOM

1. Feladat.....	2
2. Pontosított feladatspecifikáció.....	2

# 1. Feladat

Szoftver laboratórium II. házi feladat  
Teszt Elek (ELEK07) részére:

Kifejezés fa

Készítsen kifejezés fát reprezentáló osztályt! Definiáljon egyszerű általánosított műveleteket, amelyeket kiértékel a kifejezésfa segítségével! Valósítsa meg az összes értelmes műveletet operátor átdefiniálással (overload), de nem kell ragaszkodni az összes operátor átdefiniálásához! Demonstrálja a működést külön modulként fordított tesztprogrammal! A megoldáshoz ne használjon STL tárolót!

## 2. Pontosított feladatspecifikáció

A feladat egy kifejezésfát kezelő osztály elkészítése. Az osztály segítségével kifejezéseket hozhatunk létre, tárolhatunk, módosíthatunk, kombinálhatunk és kiértékelhetünk. Az osztállyal bármekkora, a memóriában elférő kifejezést előállíthatunk (dinamikus memóriakezelés).

Az Expression osztály képes többféle típusú kifejezést is kezelni (int, double) bemenetként és kimenetként, ezt template segítségével fogom megoldani.

Az alábbi funkciókat tervezem megvalósítani:

Expression(): számértéknél nullára inicializál, esetlegesen egyéb használt osztálynál az osztály default konstruktorát hívja.

Expression(érték): a felhasználó által megadott értékre inicializál, ennek az értéknek a típusával fog dolgozni a fa.

Támogatott műveletek: +, -, \*, /, ^ (hatványozás): két kifejezést ezekkel tudunk összekapcsolni akár helyben (működik pl a +=, -=, stb. operátorok), akár új változóba (copy constructor)

Kiírás: az osztály képes olvasható formában megjeleníteni a gráfot (postfix), jelöli a node-ok indexeit is.

Fájlba mentés: képes a kifejezést fájlba menteni, onnan betölteni

Indexelés: A kifejezésfa csomópontjait egyenként elérjük (a kiírásban megjelenített index szerint), és ezeken is el tudjuk végezni az osztályon végzett műveleteket (kiértékelés, kiírás, módosítás, stb). Hibás indexeléskor out of range hibát dob.

Evaluate(): az adott csomóponton (alapértelmezett esetben a gyökéren) értelmezett kifejezés értékét adja vissza.

Tesztelés: különböző kifejezéseket állítok elő, és a kiértékeléskor kapott értékeket összevetem a c++ beépített kiértékelője által visszaadott értékekkel.