KALKULATOR GUI

Osnovni princip delovanja je na prioritetni lestvici, kjer nas vsak oklepaj pomakne en nivo višje. Imamo še branch-e, ki je v bistvu indeks elementa na istem prioritetnem nivoju. Lahko imamo več elementov z isto prioriteto, torej na primer več enojnih zaključenih oklepajev, ki imajo enako prioriteto vendar se ne računajo skupaj, zato drugi branch. Zaklepaj nam inkrementira branch in nas pomakne en nivo nižje.

CLASS ELEMENT

* Float number
* Int priority
* Int branch

Vector of elements

Vector of operators (1 manj kot elementov)

Primer: 5+4\*(3+2\*5)+1 =

Vector elementov: 5[0], 4[1], 3[2], 2[3], 5[4], 1[5]

Vector operatorjev: +[0], \*[1], +[2], \*[3], +[4]

Najprej se lotimo oklepaja saj ima višjo prioriteto. Preverimo operaterje in prvo zračunamo zmnožke in deljence. \* je na indeksu 3 zato pomnožimo število na tem indeksu in prvo naslednjo, torej indeksa 3 in 4 (vektor elementov). Sedaj namesto teh dveh števil vstavimo v vektor elementov njun zmnožek in odstranimo operator iz vektorja operatorjev. Nato dobimo:

5+4\*(3+10)+1 =

Vector elementov: 5[0], 4[1], 3[2], 10[3], 1[4] (5 element se pomakne na 4 tako dobimo en element manj

Vector operatorjev: +[0], \*[1], +[2], +[3]

Sedaj bi sešteli 3 in 10 in ko dobimo znotraj oklepaja oziroma na isti prioriteti zgolj en element (eno število) lahko prioriteto znižamo. Ob tem pogledamo enega/oba sosednja elementa in prevzamemo njuno/njegovo vrednost brancha (eden od sosedov je lahko v drugem oklepaju (druga prioriteta) zato zmeraj pogledamo oba in enačimo branch s tistim sosedom, kateri ima enako prioriteto). V kolikor imata oba soseda nižjo prioriteto pač za dan element še enkrat znižamo prioriteto in preverimo sosede še enkrat.

//////////////////// Testni primer ///////////////

3\*((2+5)\*2)+(3\*2) = 48

3\*(7\*2)+(3\*2)

…. (3+(2\*5)+(3\*2))

…((2+5)\*(3+2))