**SAMM-SAMMULT**

***Iga lahendus/konstruktsioon on universaalsem ja töökindlam, kui koosneb väiksematest korralikult testitud plokkidest.***

**Sama kehtib ka ülesannete lahendamise kohta, luues selleks programme.**

**Oma kursuse jooksul liigume selles suunas samm-sammult ja järgmiselt:**

**1. Iga ülesanne peab olema üheselt arusaadav – selleks kõigile loetav algoritmi kirjeldus.**

**2. Iga ülesanne tuleb jagada alamülesanneteks ja kindlasti nii, et andmete sisestamine ja väljund oleks töötlemisest eraldatud.**

**3. Iga ülesannet tuleb testida ekstreemsete olukordade jaoks.**

**Kasutades C-keelt, toimime järgmiselt:**

**1. Koostame ülesande lahendusprotsessi näiteks UMli abil järgmiselt, et lahendusprotsess ei ole seotud C-keelega. Samal ajal toome iga UML-i osa jaoks lahendusetapil C-keele laused.**

**2. Koostame programmi C-keeles tervikuna, vastavalt algoritmile.**

**3. Leiame ja eraldame lahendusest võimalikud alamülesanded.**

**4.Vastavalt C keele reeglitele, loome alamülesannete jaoks funktsioone.**

**5. Testime kõiki funktsioone eraldi ja kogu programmi koos.**

**6. Koostame oma funktsioonide kataloogi.**

**Näide:esimene samm**

**kus on vaja leida mõõtmistulemuste keskmine väärtus.**

**1. Tuleb selgeks teha, mis tüüpi numbreid on ja kas on teada, kui palju neid on.**

**2. UML, kasutades „tegevust“:**

**3. Siin näeme kolme eraldatavat tegevust:**

**sisend, töötlemine, väljund.**

**4.Algoritm UMLis**

**4. Programm C-keeles**

Konsultatsioonid teisipäeviti eelregistreerimisega.

consultations on Tuesday **U05 104A at 16.00-18.00** with advance registration **vladimir.viies@gmail.com**