Nõo Reaalgümnaasium

Margus Kevin Sünter

12.b klass

**KALKULAATORI PROGRAMMEERIMINE**

Praktiline töö

Juhendaja Andres Mihkelson

Nõo 2014

**SISUKORD**

[SISSEJUHATUS 3](#_Toc387932975)

[1 PROGRAMMIGA ALUSTAMINE 4](#_Toc387932976)

[1.1 Python 4](#_Toc387932977)

[1.2 DataTable klass 4](#_Toc387932978)

[2. PROGRAMM 5](#_Toc387932979)

[2.1 Funktsioonid 5](#_Toc387932980)

[2.2 Veakontrollid 6](#_Toc387932981)

[2.3 Disain 6](#_Toc387932982)

[3. SILUMINE 7](#_Toc387932983)

[KOKKUVÕTE 8](#_Toc387932984)

[RESÜMEE 9](#_Toc387932985)

[LISAD 10](#_Toc387932986)

[Lisa 1. Regioonid 10](#_Toc387932987)

[Lisa 2. Overflow Error 10](#_Toc387932988)

[Lisa 3. Nupp 5 disain 11](#_Toc387932989)

[Lisa 4. Põhiliste nuppude funktsioonid 11](#_Toc387932990)

[Lisa 5. Kalkulaatori programm 11](#_Toc387932991)

[KASUTATUD MATERJALID 12](#_Toc387932992)

# **SISSEJUHATUS**

Kui ma alustasin oma programmi kirjutamise projektiga, mis suudaks avaldist lahendada, siis ma polnud üldse kindel, kas sellist programmi on üldse võimalik C# programmeerimiskeeles kirjutada. Ma otsisin päris pikalt internetist, kuni loobusin ja mõtlesin, et kirjutan programmi, kasutades programmerimiskeelt Python, sest ma teadsin, et seal saab sõne, teksti kujul märgijada, lahendada ainult ühe käsuga – *eval(string).* Mõtlesin kalkulaatorit tehes kasutada mõnd Pythoni GUI tegemiseks mõeldud moodulit. Hiljem kaalusin ka PyGame’i kasutamist, kuid lõpuks jäin ikkagi C# ja Microsoft Visual Studio juurde, sest mulle väga meeldib, kui mugavaks on tehtud graafiliste kasutajaliidese tegemine Visual Studios.

C# juurde tagasi tõi mind ka see, et kui ma uurisin Pythoni mooduleid ja teksti kujul avaldise lahendamist, siis ma sain teada, et sellised kalkulaatorid on inglise keeles *math parser* nime all. Uurides natuke rohkem nende kohta, leidsin, et keegi oli midagi laadset kirjutanud ka C# keeles. Tema kalkulaator on küll teistmoodi üles ehitatud ja töötab minu omaga võrreldes erinevalt, kuid idee ja motivatsioon tuli nähes, et sellist programmi on võimalik teha ja keegi on juba hakkama saanud.

Kalkulaator saab hakkama lihtsamate tegevustega: liitmine, lahutamine, korrutamine, jagamine ja sulgude kasutamine. Veel lisasin juurde põhiliste arvusüsteemide vahel arvude konvertimise (2, 8, 10 süsteemid). Ainukeseks probleemiks on see, et arvude maksimumpikkus on 2147483647 tähemärki.

# **PROGRAMMIGA ALUSTAMINE**

Alustada oli suhteliselt keeruline, sest ma ei olnud kindel, mis keelt kasutada: kas Python või C#. Pythonis ma oleks teadnud, kuidas sõne lahendada, kuid seal ma ei osanud graafilist poolt teha ja vastupidi.

## **Python**

Alustasin kalkulaatori kirjutamist Pythonis mõttega, et see ei saa eriti keeruline olla, sest ma tean, kuidas panna kalkulaator sõnest arve saama. Graafilise poole plaanisin algselt teha Pythoni mooduliga TkInter. Kuna ma polnud kunagi varem Pythoni programme GUIga teinud ning TkInter tundus suhteliselt keeruline, mõtlesin kasutada PyGame’i. Sellega olime natuke tutvunud ka arvutiõpetuse tunnis. Peale paari tundi programmi kirjutamist otsustasin loobuda ka PyGame’i kasutamisest, sest mulle hakkas tunduma, et PyGame’i kasutades läheb kokkuvõttes palju rohkem aega, kui seda läheks C#d ja Visual Studiot kasutades.

## **DataTable klass**

Lõpuks leidsin, et C#-s saab kasutada sõne arvutamiseks andmetabeleid. Sellega saab ideaalselt hakkama DataTable klass, kuid sellel on ka omad miinused. Näiteks ei saa arvutada suuri arve. Tehted, mille pikkus või vastus on üle 2147483648 tähemärgi [[1]](#footnote-1), ei ole lahendatavad ja kuvavad veateate, sest arvutuste käigus muudetakse vahepeal arve 32-biti andmetüüpi. Teine probleem on see, et kõik Windows Vistast vanemad Windowsi operatsioonisüsteemid oskavad kasutada ainult ühte protsessorituuma, et teha selle programmi poolt ette antud arvutusi. Samas ei ole seda muret teistel operatsioonisüsteemidel. Linuxi ja Max operatsioonisüsteemide peal programmi jooksutamiseks on soovitatav kasutada Mono või Wine programmi. Mono on vajalik .NETi jaoks ja Wine selleks, et programmi käima panna. Kõige suurem probleem on see, et Itaniumi seeria protsessorid ei oska sellist käsku kuidagi käsitleda[[2]](#footnote-2). Siiski on peale miinuste DataTable’i suureks eeliseks see, et sellist kalkulaatorit saab väga palju edasi arendada: lisada trigonomeetiat, graafikute joonistamist ja erinevate Microsoft Office programmide vahel andmete kopeerimist ning töötlemist, kuigi nende juurde lisamine on suhteliselt keeruline.

# **PROGRAMM**

Programm töötab suhteliselt lihtsalt ning võimalikult suur osa programmist on kirjutatud funktsioonidena, et funktsioone saaks ükshaaval ja eraldi välja kutsuda. Veel on funktsioonide puhul hea see, et programmeerija ei pea koodijuppe iga kord uuesti kirjutama, vaid saab alati pöörduda selle osa juurde, mida vaja läheb.

## **Funktsioonid**

Väga hea funktsioonide puhul on see, et on võimalik pöörduda konkreetselt selle programmiosa juurde, mida parasjagu läheb vaja. Selline tükkideks jaotamine teeb programmi tunduvalt lühemaks, kiiremaks ning programmi koodist on tunduvalt lihtsam aru saada. Kui tahta programmis midagi muuta, siis ei ole vajadust otsida õigeid kohti programmist, kus on viga, vaid on ainult üks funktsioon, mida lihtsalt kutsutakse korduvalt välja. Näiteks on funktsioon:

public int FtPikkus()

{

int pikkus = textBox\_Tehe.Text.Length;

return pikkus;

}

See funktsioon leiab, kui pikk on kasutaja sisestatud tehe, salvestab selle täisarvuna muutuja „pikkus“ nimega ning tagastab selle. Funktsioonina on seda väga hea kasutada, sest hiljem on vaja seda korduvalt kasutada. Näiteks kasutab programm seda funktsiooni hiljem kontrollis, kus kontrollitakse, et tehte alguses ei oleks jagamise ega korrutamise märki. Koodi kergemini loetavaks tegemise huvides leidsin, et mul on võimalik programmiosi jagada erinevatesse regioonidesse, mida on võimalik eraldi sulgeda, et regioonis olev osa ei hakkaks segama koodi lugemist, kui mul seda parasjagu vaja ei lähe.

Kalkulaatorile sai juurde kirjutatud väikesed lisad, mida tavalistel kalkulaatoritel tavaliselt olemas ei ole. Esiteks, võttes linnukese ära kastist „Halliks“, saab tehte ja vastuse kasti otse kirjutada. See võimaldab tehteid sisse kirjutada, kasutades klaviatuuri või neid kopeerida mõnest teisest programmist – ka saab vastust kopeerida. Programmi enda nupustikku kasutades lisavad need tähemärke siiski tehtekasti lõppu. Teiseks saab muuta arvusüsteeme, milles kalkulaator vastuse annab. Vajutades nuppu „Teisenda!“, proovib programm lahendada sisestatud tehet ning kirjutada vastusena kasti juba teisendatud arvu. Teisendada saab kümnendsüsteemist kahend-, kaheksand- ja tagasi kümnendsüsteemi.

## **Veakontrollid**

Alguses tegin ise ka nulliga jagamise kontrolli, kuid Visual Studio uuendusega parandati nulliga jagamise vead ära Microsofti poolt, nii et minu kontroll ei olnud enam vajalik. See oli väga hea, et Microsoft ise tegi sellise uuenduse, sest perfektse nulliga jagamise kontrolli tegemine oleks olnud päris keeruline. Seega kustutasin oma funktsiooni ära, sest see töötas alles poolikult ning oleks jäänud mõttetult arvutit kurnama.

Enamik veakontrolle tehakse ära nupuvajutuse hetkel, et hiljem oleks võimalikult vähe vigu, mida välja filtreerida. Tehtemärkide nuppude vajutamisel kontrollitakse eelnevat tähemärki tehtekastis op\_check funktsiooniga – see funktsioon tagab, et ei pandaks kahte tehtemärki järjest, näiteks kahte korrutamis- või jagamismärki. Kõikvõimalikud vead, mis võivad tekkida, ei tohiks tervet programmi kokku jooksutada. Nii palju, kui on võimalik, kuvatakse vigade tagasiside vastusekasti, kuhu tulevad vead, mida kasutaja peaks jälgima. Näiteks: „\* tehte alguses“ teade kuvatakse, kui kasutaja proovib tehet alustada korrutamise märgiga.

Arvude teisendamisel erinevate süsteemide9 vahel on veakontrollid, mis käivitatakse alles siis, kui kasutaja vajutab „Teisenda!“ nuppu. Nuppu vajutades kutsutakse välja vastav funktsioon, mis teisendab arvu ümber teisse arvusüsteemi. Kui arvu teisendamine ei tule välja, siis kuvatakse viga vastusekasti. Samal põhimõttel tehakse ka viimane veakontroll. Kui programm ei suuda mingil põhjusel vastust välja arvutada, siis kuvab kalkulaator veateate „Viga arvutamisel!“.

## **Disain**

Koodi disainiosa lasin Visual Studiol genereerida. Käsitsi kirjutades oleks pidanud ise paika panema, millistel koordinaatidel mis nupp asub ning kuidas ta teiste nuppudega seotud on. Visual Studio genereeritud kood on küll natuke pikem, kuid see teeb programmeerimise palju kiiremaks, kui ei pea käsitsi kõike välja kirjutama. Visual Studios saab lihtsalt lohistada nupu sinna, kuhu on vaja, kasutades *drag-and-drop-*tehnikat. Kuna disaini käsitsi kirjutamine pole üldse nii huvitav kui programmi enda, siis panin rohkem rõhku programmi kirjutamisele ja lasin programmi disainiosa Visual Studiol ise koodiks muuta.

# **SILUMINE**

Nagu ikka, võttis ka minul programmist vigade eemaldamine väga pikalt aega. Tuli esile palju vigu, mida alguses ei pannud tähelegi ja mis nõudsid parandamist. Mõne vea lahendamine võttis päris mitu tundi ja nõudis poole programmi loogika ümberkirjutamist, samal ajal kui mõne teise vea parandamine võttis ainult hetke.

Esimest korda, kui mõtlesin, et programm võiks valmis olla, ja andsin tuttavatele katsetada, et vigu otsida, olin unustanud „0“ nupu lisada. Ühe nupu juurdelisamine käib üpriski kiirelt, seepärast mõtlesin, et võiks tekitada ka juba koma ja sulgude nupud. Sellest ajast, kui ma tehtemärgid lisasin, oli juba mitu päeva möödas, seepärast olin ära unustanud, kus on kontroll, et ei saaks mitut märki järjest panna. Mõnda aega saigi kalkulaatorisse sisestada sulgusid ja komasid ükskõik kuidas: läbisegi ja lugematul hulgal.

Teist korda, kui arvasin, et on valmis, oli sisse jäänud siiski väga palju vigu. Näiteks ei saanud teha tehteid, kus jagamine oli ühest väiksema arvuga, sest programm arvas, et kasutaja jagab nulliga ja kahendsüsteemi teisendades läks arv millegipärast lõpmatuseni pikaks, kuni programm kokku jooksis.

Kolmanda korraga peaks olema programmist vead eemaldatud. Muidugi võib olla, et on vigasid, mis on märkamata jäänud, aga selliseid probleeme, et kalkulaator nii kokku jookseb, et lakkab töötamast ja läheb kinni, ei tohiks enam olla.

# **KOKKUVÕTE**

Kalkulaatori peamine eesmärk on täidetud: see oskab sisestatud sõne lahendada ning anda sellele arvuline vastus. Väikseid tegevusi sai õpieesmärgil veel juurde lisatud.

Kõige tähtsam on minu arvates motivatsioon programmeerimiseks, mida saab selliste programmide kirjutamisest. Kui tekib mõni probleem, mida kohe lahendada ei oska, siis on suur teotahtelangus ja ei taha enam üldse asjaga tegeleda, kuni lahendus pähe torkab. Peale seda, kui probleem on lahendatud, on jälle hea tunne ning tekib tahtmine edasi teha ja ikka midagi uut lisada.

Edasi plaanin teha enda universaalse programmi, mis teeb kõike, mida ma soovin. Kord juba alustasin selle projektiga, kuid see kukkus läbi. Antud kalkulaator on jälle samm edasi ning aitas koguda teadmisi. Natuke sai ka muud teada, näiteks, et kuidas töötab Microsoft Excel ning millised vead sellel küljes on ja missugused vead olid aastaid tagasi. Kalkulaator on tihedalt seotud Exceliga, sest mõlemad kasutavad arvutamiseks DataTable’it.

# **RESÜMEE**

I made my first program when I was around age of 7. It was quite simple, program asked for my name and age. When user typed them in it responded „Tere #name sa oled #age aastat vana“. After that program I did not do any programming until I went to 7th grade. We made some really basic programs with friend of mine, who is professional Java developer now. Programs were made quite like copy – paste from TheNewBoston video tutorials. We used those tutorials to get to know some basic knowledge and syntax of Java, PHP and C programming languages. He moved away when I went to 8th grade and so I stopped programming until I came to NRG.

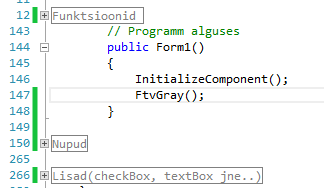
When I came to Nõo reaalgümnasium I started to learn more about programming in our computer lessons. In our lessons we used programming language called C# and Microsoft Visual Studio. Visual Studio is an integrated development environment made by Microsoft. We used it to develop programs that have graphical user interface (GUI) – buttons and textboxes. Before that I had made only command promt based programs, where I enter text and get some sort of respond as in text format.

Programming is fun and hard at the same time – I can virtually make anything I want, but it takes knowledge and logical thinking. I feel like I lack both of them to be professional programmer. I have planned to make calculator for quite long, just to test my skills.

My calculator is quite unique, becouse it does not take numerial input, it takes string. Tries to understand it, calculate it and give right answer for user entered operation. Using string to calculate has its own pros and cons. Such program has almost infinite upgrading potential and can communicate perfectly with Microsoft Office programs such as Excel and Word for example. Huge shorcoming is that such program is too slow to calculate ultra long calculations necessary on supercomputers.

# **LISAD**

## **Lisa 1. Regioonid**

****

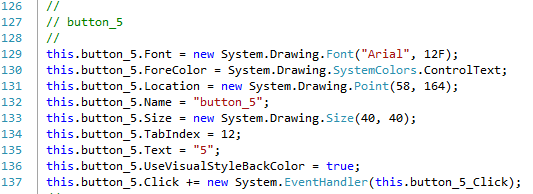
Regioonide kommentaarid on kastis. Neid saab avada plussile vajutades. Vasakul olevad numbrid on koodiread, kust on näha, kuidas regioonid neid kokku pakivad.

## **Lisa 2. Overflow Error**



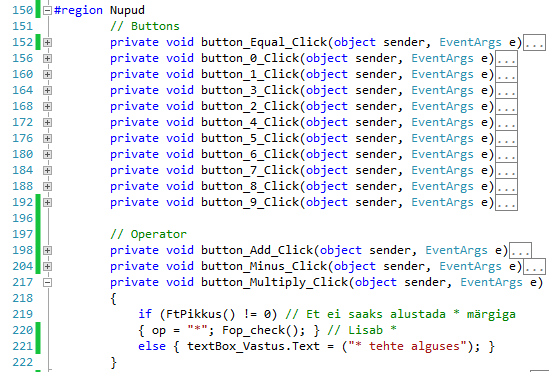
Alguses tekkis viga, kui arvud läksid arvutuse käigus liiga suureks.

## **Lisa 3. Nupp 5 disain**



Nupu 5 disainikoodi osa, mille Visual Studio teeb automaatselt.

## **Lisa 4. Põhiliste nuppude funktsioonid**



Nuppude funktsioonid on automaatselt genereeritud, nende sisu ise kirjutatud

## **Lisa 5. Kalkulaatori programm**

Programm on lisatud mälupulgal.

## **Lisa 6. Kasutatud materjalide loetelu**

Juhendid:

* <http://msdn.microsoft.com/en-us/library/8z6watww.aspx>
* <http://msdn.microsoft.com/en-us/library/ff700515(v=office.14).aspx>
* <http://msdn.microsoft.com/en-us/library/0yd65esw.aspx>
* <http://msdn.microsoft.com/en-us/library/system.windows.forms.textboxbase.clear(v=vs.110).aspx>
* <http://msdn.microsoft.com/en-us/library/14kwkz77(v=vs.110).aspx>
* <http://msdn.microsoft.com/en-us/library/system.data.datatable.compute.aspx>
* <http://msdn.microsoft.com/en-us/library/9ha04z01.aspx>
* <http://msdn.microsoft.com/en-us/library/system.data.datatable.aspx>
* <http://technet.microsoft.com/en-us/library/aa933198(v=sql.80).aspx>
* <http://www.dotnetperls.com/datatable>
* <http://csharp.net-tutorials.com/basics/functions/>
* <http://www.dotnetperls.com/string>
* <http://www.codeproject.com/Tips/381509/Math-Parser-NET-Csharp>
* <http://www.rapidtables.com/math/number/Numeral_system.htm>

Stackoverflow kommuun:

* <http://stackoverflow.com/questions/14668750/calculating-strings-as-values>
* <http://stackoverflow.com/questions/136035/catch-multiple-exceptions-at-once>
* <http://stackoverflow.com/questions/94591/what-is-the-maximum-value-for-a-int32>
* <http://stackoverflow.com/questions/1802597/identifying-exception-type-in-a-handler>
* <http://stackoverflow.com/questions/923771/quickest-way-to-convert-a-base-10-number-to-any-base-in-net>
* <http://stackoverflow.com/questions/14668750/calculating-strings-as-values>
* <http://stackoverflow.com/questions/10175357/c-sharp-creating-and-using-functions>
* <http://stackoverflow.com/questions/13103360/how-to-write-programs-in-c-sharp-net-to-run-them-on-linux-wine-mono>

Teave:

* <http://en.wikipedia.org/wiki/Itanium>
* <http://www.mono-project.com/Main_Page>
* <http://www.pygame.org/news.html>
* <https://wiki.python.org/moin/TkInter>
* <https://wiki.python.org/moin/GuiProgramming>

1. Stackoverlow küsimused. (8.9.2008). Küsimus: “What is the maximum value of int32?” Loetud aadressil <http://stackoverflow.com/questions/94591/what-is-the-maximum-value-for-a-int32> [↑](#footnote-ref-1)
2. Microsoft bõrsiettevõte. (kuupäev puudub). Loetud aadressil [http://msdn.microsoft.com/en-us/library/system.data.datatable.aspx](%20http://msdn.microsoft.com/en-us/library/system.data.datatable.aspx) [↑](#footnote-ref-2)