

# Rühmatöö NR. 1

## Autorid:

Marek Lepik, Kevin Kits

## Projekt Sudoku:

Projekt Sudoku programm, mille eesmärk on sudoku lahendamine. Programm kasutab lahenduse leidmiseks *brute force*-i. See tähendab, et programm paneb kohtadele arve 1-9 ning seejärel jooksub läbi mitme kontrolli, mis teavitab programmi, kas ta on õigel teel või mitte. Programm küsib alguses, millisele sudokule soovid lahendust sudokude tekst failist. Programm oli tehtud selliselt, et hiljem saaks Rühmatöö NR. 2 raames lisada GUI-d ja rohkem valikuvõimalusi. **Õpetus:** Sudoku lisamise soovil oleme jätnud faili „Sudokud.txt“ rea vabaks selleks, et lisamine oleks eriti kerge. Lisades tuleb meeles pidada, et tühjad kohad on märgistatud elemendiga 0 ning lisamine käib 3x3 ruutude kaupa (mitte ridade või veergude).

## Klasside eesmärgid:

**Peameetod** – Vastutab kasutaja sisestuse, failist sisselugemise ja sudoku algse ja lõpu versiooni printimise ning lahedaja välja kutsumise eest.

**Sudoku** – Vastutab sudoku andmete hoiustamise ja neile kergesti ligipääsetavuse eest. Selles klassis on ka kõik isoleerivad meetodid.

**SudokuLahendaja** – Vastutab sudoku lahendamise ja lahenduse õigsuse kontrollimise eest.

## Projekti tegemise protsess:

- 1) Projekti teema väljamõtlemine
- 2) Vajaminevate etappide üle loetlemine ning nende eest vastutuse jagamine
- 3) Sisestuse viisi kokkuleppimine
- 4) Enda osa tegemine
- 5) Kogu programmi esialgne testimine
- 6) Vigadeparandus
- 7) Uus test mitme sudokuga
- 8) Vigadeparandus
- 9) Dokumentatsioon

Kõigis etappides oleme panustanud ühtlaselt ning teinud koos.

## Rühmaliikmete vastutus:

Meil ei olnud otsest jaotust klasside suhtes, kuid üldiselt juhtus nii, et Marek tegeles rohkem Sudoku klassiga ning Kevin SudokuLahendaja klassiga. Mõlema klassi puhul oli siiski teise rühmakaaslase panus võrdne ning soovitusi ja lahenduskäike mõeldi ja arutati koos. Kahe rühmaliikme peale oli orienteeruv ajakulu 14 tundi.

## Mured:

Mureks ei saaks tuua mõnda oskuse puudmist (isegi kui see oli siis ei oskaks seda välja tuua). Tugevateks pähkliteks oli rohkem süsteemide väljamõtlemine, et iga programmi etapp mööduks probleemideta. Programi kirjutamise käigus oli väga oluline, et eelnev etapp töötaks laitmatult, selleks et järgnev saaks toimida. Kogesime väga tugevat tiimitööd ning

keskendusime enda etappi täiustamisel selleks, et rühmakaaslane saaks järgmisel etapil ülevõtta. Konstantsel etappide üleandmisel kogesime aktiivset rühmatööd, mis eeldas arusaadavat programmeerimist, et teine osapool ka sellest arusaaks.

## Hinnang:

Oleme lõpptulemusega enamjaolt rahul. On osasi, mis õnnestusid ootustest paremini ning ka osasi, mis võiksid olla ilusamad. Kõik enda selle rühmatöö nõrkused muudame tugevusteks nii, et „ju siis see oligi niimodi plaanitud, et Rühmatöö NR.2 oleks piisavas koguses täiustust“. Kuid tegelikult jäime mõlemad väga rahule.

## Testimine:

Esimeseks suureks osaks sai sudoku sisestus ning erinevate osade isoleerimine, selleks sai loodud paint-iga pilt, mida vaadates valmiski viis kuidas isolatsioon peaks toimima. Välja kirjutades oli palju arusaadavam, mis rea või veeru vaatamisel indeksid oleksid. (Pilt 1)

```
[2, 6, 9, 4, 4, 0, 7, 0, 3]
[2, 6, 9, 5, 4, 0, 7, 0, 3]
[2, 6, 9, 6, 4, 0, 7, 0, 3]
[2, 6, 9, 7, 4, 0, 7, 0, 3]
[2, 6, 9, 8, 4, 0, 7, 0, 3]
[2, 6, 9, 9, 4, 0, 7, 0, 3]
[2, 7, 9, 0, 4, 0, 7, 0, 3]
[2, 8, 9, 0, 4, 0, 7, 0, 3]
[2, 9, 9, 0, 4, 0, 7, 0, 3]
[3, 0, 9, 0, 4, 0, 7, 0, 3]
[4, 0, 9, 0, 4, 0, 7, 0, 3]
[5, 0, 9, 0, 4, 0, 7, 0, 3]
[5, 1, 9, 0, 4, 0, 7, 0, 3]
[5, 1, 9, 1, 4, 0, 7, 0, 3]
[5, 1, 9, 2, 4, 0, 7, 0, 3]
[5, 1, 9, 2, 4, 1, 7, 0, 3]
[5, 1, 9, 2, 4, 2, 7, 0, 3]
[5, 1, 9, 2, 4, 3, 7, 0, 3]
[5, 1, 9, 2, 4, 4, 7, 0, 3]
[5, 1, 9, 2, 4, 5, 7, 0, 3]
[5, 1, 9, 2, 4, 6, 7, 0, 3]
[5, 1, 9, 2, 4, 7, 7, 0, 3]
[5, 1, 9, 2, 4, 8, 7, 0, 3]
[5, 1, 9, 2, 4, 8, 7, 1, 3]
[5, 1, 9, 2, 4, 8, 7, 2, 3]
[5, 1, 9, 2, 4, 8, 7, 3, 3]
[5, 1, 9, 2, 4, 8, 7, 4, 3]
[5, 1, 9, 2, 4, 8, 7, 5, 3]
[5, 1, 9, 2, 4, 8, 7, 6, 3]
[3, 1, 0, 0, 5, 0, 0, 1, 8]
[3, 2, 0, 0, 5, 0, 0, 1, 8]
[3, 2, 1, 0, 5, 0, 0, 1, 8]
[3, 2, 2, 0, 5, 0, 0, 1, 8]
[3, 2, 3, 0, 5, 0, 0, 1, 8]
[3, 2, 4, 0, 5, 0, 0, 1, 8]
[3, 2, 5, 0, 5, 0, 0, 1, 8]
[3, 2, 6, 0, 5, 0, 0, 1, 8]
```

Pilt 2. konsool

Teiseks osaks sai esialgse kontrolli loomine, mis vaatas, et järjendis olevad arvud on kõik ühekordselt.

Kolmandaks ja vast kõige olulisemaks osaks sai muidugi lahendaja loomine ja testimine. Selle teostamine käis nii, et tehti midagi valmis ning siis kontrolliti ühe 3x3 sudoku osa pealt kas teoreetiliselt see lahendus tuleb õige. Kui see õnnestus siis liiguti edasi ja vaadati kogu sudokut. See nõudis palju näpuga järke ajamist konsooli järjendites, kus on palju arve ja kohati arusaamatuid lahendusi. (Pilt 2)

Neljandaks osaks oli lõpliku kontrolli kirjutamine, mis juba kontrollis ka seda, et ühtegi tühja kohta pole jäetud.

Viiendaks osaks kujunes erinevate sudokude tekst faili sisestamine ning valiku võimaluse loomine. Kontrollisime mitme erineva raskustaseme sudokude lahendust ning avastasime, et meie lahendaja ei väljasta kõige raskemate sudokude vastust, mis võis tähendada kahte asja, kas ta jääb kuhugile kinni ja ei suudagi lõpule viia seda või lahendusi on liiga palju ja tal läheb *Brute force*-miseks liiga kaua aega. Proovisime mõneks ajaks programmi käima jätta, kuid tulemust ei saanud. Otsustasime, et proovime Rühmatöö NR.2- eks täiustada enda programmi.

	0	1	2		3	4	5		6	7	8
0	1	5	2		4	8	9		3	7	6
1	7	3	9		2	5	6		8	4	1
2	4	6	8		3	7	1		2	9	5
3	3	8	7		1	2	4		6	5	9
4	5	9	1		7	6	3		4	2	8
5	2	4	6		8	9	5		7	1	3
6	9	1	4		6	3	7		5	8	2
7	6	2	5		9	4	8		1	3	7
8	8	7	3		5	1	2		9	6	4

Pilt 1. Sudoku lahamine.

## Lõppsõna

Oleme väga rahul enda programmiga ning tunneme, et oleme midagi lahendat valmis saanud. Mõte, et kirjutada programm, mis lahendab sudokut tundus palju raskem kui see tegelikult välja kujunes. Sellistel hetkedel on tunne, et oleme suuteliselt programmeerima juba raskematele probleemidele ka lahendust. Loodame, et programm valmistab kontrollijale samapalju rõõmu kui meile.