Gauss Kalkulator

Jeg har valgt å lage en app som løser ligningssystemer med 3 ligninger. Jeg valge dette prosjektet fordi jeg mente det kunne være nyttig både i objekt, men også i matematikk 3. Derfor får jeg et læringspotensial i begge fagene. I appen har jeg en klasse som gjennomfører kalkulasjonene som heter «GaussCalculator2». Appen tar inn en matrise og en vektor med double verdier. Denne vektoren og matrisen blir i controller klassen gjort om til en matrise som videre blir sent til «GaussCalculator2» Når man bruker metoden «GaussElimination()» blir matrisen og vektoren endret til den ferdige matrisen og vektoren. Så sendes matrisen og vektoren til klassen «GaussShowCalculation» som har metoden «printFinal()» som oversetter matrisen og vektoren tilbake til et ligningssystem iform av en String. Dette sendes tilbake til kontrolleren som endrer et TextField til dette ligninsystemet. Matrisen og vektoren kan lagres i en fil, som siden kan importeres så man får matrisen og vektoren tilbake. I dette prosjektet har jeg fått bruk for store deler av pensum, og bruk en god del matte også.

Sekvensdiagram

Et bilde som inneholder tekst, diagram, nummer, line

Automatisk generert beskrivelse

Spørsmål

1. Hvilke deler av pensum i emnet dekkes i prosjektet, og på hvilken måte? (For eksempel bruk av arv, interface, delegering osv.)

Prosjektet bruker store deler av pensum. Prosjektet bruker mye innkapsling for å holde programmet trygt. Programmet bruker også validering, et eksempel på dette er at input kan bare være nummere og ikke bokstaver. Appen bruker lambda funksjoner i form av streams. Og appen bruker fillagring til å lagre matriser og for å importere matriser. Prosjektet bruker også flere klasser som har forskjellige oppgaver. Prosjektet bruker delegering iform av at en klasse blir brukt til kalkulasjoner og en klasse blir brukt til å lage en String.

1. Dersom deler av pensum ikke er dekket i prosjektet deres, hvordan kunne dere brukt disse delene av pensum i appen?

Appen bruker ikke observatør-observert fordi jeg mente det ikke var nødvendig for denne oppgaven. Appen bruker heller ikke arv fordi det ikke nødvendig siden appen ikke trengte mange klasser. Har egentlig en klasse som gjør mesteparten av jobben i form av kalkulasjoner.

1. Hvordan forholder koden deres seg til Model-View-Controller-prinsippet? (Merk: det er ikke nødvendig at koden er helt perfekt i forhold til Model-View-Controller standarder. Det er mulig (og bra) å reflektere rundt svakheter i egen kode)

Prosjektet forholder seg nogen lunde til Model View Controller prinsippet i form av at det er en Controller klasse som gjør minimalt av arbeid men delegerer kalkulasjonene til «GaussCalculator2» klassen som tar seg av kalkulasjonene. Den har også en View klasse som heter «GaussShowCalculation» som har methoden «printFinale()» som gjør om matrisen og vektoren om til en String som kan printes på appen.

1. Hvordan har dere gått frem når dere skulle teste appen deres, og hvorfor har dere valgt de testene dere har? Har dere testet alle deler av koden? Hvis ikke, hvordan har dere prioritert hvilke deler som testes og ikke? (Her er tanken at dere skal reflektere rundt egen bruk av tester)

Den viktigste delen av appen er klassen «GaussCalculator2» som tar seg av selve gauss eliminasjonen. Prioriterte derfor å teste denne klassen grundigst. Klassen er bygget opp av flere metoder som tilhører de forskjellige stegene i Gauss elemineringen. Lagde en test for hver av disse metodene som testet om metoden manipulerer matrisen og vektoren slik den skal. Metoden «GaussEliminationStep1» skal legge den raden som har et element først i kolonnen øverst i matrisen. Metoden «GaussEliminationStep2» skal trekke fra den øverste raden fra de resterende radene. Metoden «GaussEliminationStep3» skal fjerne den øverste raden, og den første kolonnen. Disse tre stegene skal gjøres helt til den aktive matrisen bare har et element. Da er matrisen rad redusert så langt det går.Så skal metoden «GaussEliminationStep4» trekke fra hver rand fra de andre radene så matrisen blir på redusert trappeform. For hver av disse metodene er det laget tester som tester at de fungerer slik som de skal. Har også tatt med en test som tester lagring av matriser, og at matrisene er på riktig format i txt filen som blir lagret.