Rapport – øving 4

Oppgave 2 og 3

Oppgave 2 (og 5-2 i boka)

Jeg har benyttet meg av en implementasjon av en stakkstruktur for å håndtere nøstingen av parenteser. Programmet leser den gitte filen bokstav for bokstav, og lagrer alle åpningsparenteser i stakken. Dersom den møter en lukkeparentes, sammenligner den denne med den øverste åpningsparentesen i stakken. Dersom programmet som leses er skrevet korrekt, vil alltid den siste åpningsparentesen og den påfølgende første lukkeparentesen være av samme type.

Programmet sier ifra dersom programmet som leses ikke har nøstet parentesene korrekt, og forsøker å lokalisere hvor det for eksempel mangler en lukkeparentes.

I tillegg har jeg implementert en slags «tilstand» som sjekker om bokstavene som leses er del av en streng eller en type kommentar, slik at den kan se bort ifra disse da de ikke kan påvirke nøstingen av parenteser i programmet. Dette gjorde jeg ved hjelp av «bool»-variabler.

Jeg fikk testet at programmet kjørte korrekt på både C- og Java-filer. I tillegg forsøkte jeg å ødelegge for programmet ved å ødelegge nøstingen i filene, og dette fikk jeg tilbakemelding fra programmet om. Jeg forsøkte også å legge til parenteser i kommentarer og strenger, og programmet ignorerte de som forventet.

Oppgave 3 (og 6.2-5 i boka)

A graph paper with green circles and dots

Description automatically generated

1. Jeg ville ha benyttet meg av inordentraversering (hvis vi skal regne det ut for hånd eller på en kalkulator), fordi vi da regner ut subtrærne til venstre før vi utfører operasjonen (foreldernoden) på det høyre subtreet.

Rekkefølgen vil da se slik ut:

A graph paper with numbers and circles

Description automatically generated

1. Denne måten å traversere treet på kalles for inordentraversering.
2. Vi kan se at den følgende metoden evner å kalkulere den endelige verdien til det aritmetiske uttrykket:

A screen shot of a computer code

Description automatically generated

Ved å benytte oss av en slags postordentraversering, hvor det nøstes opp i løv-/barne-noder før foreldrenodene, vil vi kunne finne den endelige summen ved å utføre en siste enkel operasjon på de to nodene som er barn til rot-noden. Vi bør ta forbehold om at det kan oppstå en situasjon hvor det bes om å dividere med 0, slik vi har gjort her.

I denne koden er det kun tillatt med heltallige noder på bakgrunn av simpelhet, men dette kan enkelt endres til å tillatte tall med desimaler.

Ved å kjøre (blant annet) koden over på treet vist i oppgave 3, får vi følgende resultater:

A black background with white text

Description automatically generated

Vi kan enkelt verifisere at denne løsningen holder vann, ved å for eksempel endre løv-noden med verdi 3 til en forelder med to løv-noder med verdier 3 og 4.

A black background with white text

Description automatically generated