

Videregående programmering - Aflevering 1

Indledning:

Jeg har valgt at implementere en løsning, der inkluderer de 3 ekstraopgaver, samt ingen begrænsning på antallet af navne.

Opbygning:

I forhold til opbygning, så har jeg prøvet at gøre det sådan, at jeg har oprettet egne funktioner for alting, der arbejder direkte med data. Det vil sige tilføjning, sletning og opdatering af tekst. Derudover har jeg lavet en trySort funktion, som sorterer elementerne, hvis muligt, og returnerer true, hvis det er lykkedes. Dermed kan jeg benytte denne funktion direkte i mine if-statements, hvilket jeg hovedsageligt benytter i forbindelse med automatisk sortering og i forhold til at udskrive fejlbeskeder.

Koden under eventhandlerne har jeg prøvet at holde simpel, omend lidt lang i nogle tilfælde. Det vil sige, at jeg så vidt muligt har holdt mig til sammenligninger, kald af mine andre funktioner, samt ændringer af enkelte variabler og tekst.

Det synes jeg sikrer en nogenlunde overskuelig kode.

Derudover har jeg prøvet at sikre en ensartet og forståelig navngivning igennem programmet.

Diskussion:

I forbindelse med ekstraopgave 1, så fungerer slet-knappen imellem kasserne ved at slette linjen, hvorpå tekst-cursoren sidst har stået i den sidst aktive kasse.

Hvad angår sortering til ekstraopgave 2, så foretrækker jeg, at brugeren selv kan manipulere inputtet, så det er nemmere at følge med i, hvad der sker. Derfor har jeg implementeret det sådan, at automatisk

sortering er slået til fra start og angivet ved en knap. Man kan slå det fra og til ved at klikke på knappen, hvorefter man kan benytte sortering manuelt, og man kan ændre retningen af autosorteringen ved at benytte de andre knapper med autosortering slået til.

Det skal nævnes, at selvom programmet starter med autosortering slået til, så kan man ikke slå det til igen, hvis der er 0 elementer i array og liste. Det er for at undgå unødvendig sortering, når der ikke er nogle elementer.

Derudover vil autosortering, når funktionen slås til, gå tilbage til den opførsel, den havde før - altså om der skal sorteres forlæns eller baglæns.

I forhold til ekstraopgave 3, hvor jeg har valgt at bruge Lists, så har jeg lavet en løsning, der inkluderer en ekstra tekstboks, der viser indholdet af listen.

Der vil derfor være steder med unødvendig eller ikke optimal kode, da jeg har prøvet at holde array-delen og liste-delen separat i de enkelte metoder. Under normale omstændigheder ville man selvfølgelig ikke gøre dette, og jeg har kommenteret i koden, hvor det er tilfældet.

Da mit array og liste er globale variabler, så ville man også kunne arbejde direkte på dem uden at sende parametre til metoderne i visse tilfælde. Særligt i tilfældet `textRemove` er parametret `numToRemove` det samme som den globale variabel `currentLine`. Jeg benytter dog parametret i et enkelt tilfælde, i forbindelse med `tryParse`, og har valgt at sende det med i et tilfælde, hvor `currentLine` kunne være benyttet direkte.

Et alternativ kunne være at lave en overloaded funktion, der kan kaldes uden argument.

Konklusion:

Jeg synes, at jeg har fået implementeret en rimelig god og overskuelig løsning, der opfylder kriterierne til opgaven.

Der er plads til forbedringer, særligt hvis programmet skulle bruges af andre, og et sted hvor koden kunne forbedres er med yderligere fejlhåndtering. Et eksempel kunne være en kontrol af, hvorvidt et indtastet navn er gyldigt, og ikke kun om der er indtastet noget.

Det ville også give mening at arbejde på at beskytte eller eliminere de globale variabler for at øge sikkerheden i programmets drift.