

1 Gamla tentauppgifter (från 021211)

- a) Skriv en funktion `successiveAverages` som tar en lista av flyttal (`real`) och returnerar en annan lista av flyttal vars element är medelvärdet av successiva par av element ur den första listan.

Exempel: `successiveAverages [2.5,3.5,0.0,10.0] = [3.0,1.75,5.0]` eftersom 3.0 är medelvärdet av 2.5 och 3.5, 1.75 är medelvärdet av 3.5 och 0.0 etc.

Du får själv bestämma hur funktionen skall bete sig om argumentet endast har ett element eller är tomt. Det beteendet måste dock vara en naturlig specialisering av beteendet för längre listor.

- b) En enkel databas (tabell) är lagrad på en fil. Filen innehåller omväxlande rader med nycklar och tabellvärden. Det tabellvärde som hör till en viss nyckel ligger i raden efter nyckeln.

Skriv en funktion `searchFile` med två argument `file` och `key`, där `file` är namnet på filen och `key` är nyckeln som du skall söka efter (en sträng). Om nyckeln hittas skall värdet av funktionen vara `SOME s`, där `s` är tabellvärdet – annars `NONE`.

Du måste jämföra nycklarna allteftersom du läser filen. Du får alltså inte t.ex. läsa in hela filens innehåll till en lista först och därefter söka i listan. Du måste också se till att du stänger alla strömmar som du öppnar.

- c) Koda en abstrakt datatyp `account` som representerar ett bankkonto. Förutom saldot (ett heltal) skall datatypen innehålla kontonummer (en sträng) och spara information om ändringar av kontot så att man kan göra ett kontoutdrag.

Primitiver enligt nedan skall finnas. Här ger jag bara en kortfattad beskrivning av vad funktionerna skall göra. Du får själv skriva en fullständig specifikation.

- `makeAccount(id) : string -> account`
Skapar ett kontoobjekt med angivet kontonummer och saldo = 0.
- `updateAccount(acct, amount) : account*int -> account`
Ger tillbaka ett nytt kontoobjekt där saldot uppdaterats med angivet belopp.
- `resetTransactions(acct) : account -> account`
Ger tillbaka ett nytt kontoobjekt där informationen om ändringar (men inte saldot) nollställts.
- `getBalance(acct) : account -> int`
Hämta saldot i kontoobjektet.
- `getId(acct) : account -> string`
Hämta kontonumret i kontoobjektet.
- `getTransactions(acct) : account -> int list`
Hämta en lista av alla uppdateringar till kontot, i kronologisk ordning.

2 Begrepp – repetition

- Abstrakt Datatyp
- Ackumulator
- Anonym funktion
- Argument
- Array
- Basfall
- Defensiv programmering
- Funktion
- Högre ordningens funktion
- Identifierare
- Imperativ Programmering
- Invariant
- Iteration (Svansrekursion)
- Konstruktör (Tag/Etikett)
- Lista
- Mönstermatchning
- Nyckel
- Polymorf typ
- Postcondition
- Precondition
- Predikat
- Primitiv
- Referens
- Rekursion
- Rekursionsvariant
- Returvärde
- Selektor
- Sidoeffekt
- Ström
- Taggad datatyp
- Terminering
- Tupel
- Typ
- Undantag (Exception)
- Uppräkningstyp
- Uttryck
- Värde