

Programmeringsuppgift

Syftet med uppgiften är att bedöma hur bekväm du är med grundläggande programmeringskonstruktioner.

Tekniska krav

Vi föredrar om uppgiften görs i Java men det är helt okej att göra uppgiften i det språk du känner dig mest bekväm i. Om du väljer Java skall uppgiften göras i en av Javas LTS versioner. Den ska lämnas in som en zip med källkoden eller som en länk till ett github-repo, och innehålla ett byggsript som bygger ett huvudprogram som kan användas för att exekvera koden med godtyckligt testdata.

Om du inte vill lösa det i Java, och väljer att lösa det i ett annat språk, tänk på att använda ditt språks konstruktioner för generaliseringar och kompositioner.

Du får använda standardbibliotek.

Uppgiften

Uppgiften går ut på att skriva ett program som utser en vinnare från en lista av deltagare av en turnering. Listan innehåller resultatet av 3 olika tävlingar, där deltagare kan vara med i en eller flera av tävlingarna. Du ska läsa av resultatet från en text-fil och returnera vinnaren av hela turneringen. För att vinna så måste en deltagare ha varit med i alla 3 tävlingarna och ha den snabbaste totala tävlingstiden (summan av tiden för varje tävling). Tänk på att försöka lösa problemet objektorienterat och som att detta är en del av ett större program.

Varje rad i filen representerar en tid för en deltagare i en tävling och består av:

- Namn på deltagaren
- Unikt ID
- Starttid
- Sluttid
- Tävlingstyp (race type)

Dessa är separerade med kommatecken och inga mellanslag, se nedan exempel:

```
Steve Carell,3100693,14:07:10,14:14:05,1000m
```

Det finns 3 olika typer av tävlingar: 1000m, eggRace och sackRace. Alla deltagare kan vara med i alla tävlingarna, men vissa deltagare har enbart deltagit i några av tävlingarna. Om en deltagare är med i flera tävlingar dyker de upp på flera rader i filen, med samma namn och id, se nedan exempel:

```
Steve Carell,3100693,14:07:10,14:14:05,1000m  
Steve Carell,3100693,12:16:11,12:19:08,eggRace
```

Du vill returnera namnet och ID:t på vinnaren samt deras genomsnittliga tid. Om det är flera personer som har fått samma tid, returnera alla vinnare.

Glöm inte att validera de olika värdena och ge användbara felmeddelanden när någonting går fel i programmet. Om inläsningen av en rad går fel, logga informationen om raden och gå vidare i filen. Det ska alltså gå att läsa in en fil med fel i sig och ändå få ut en vinnare.

Det finns en exempel-fil som du kan använda dig av som skickas i samband med detta dokument, race-results.txt. Den innehåller diverse fel som hade kunnat dyka upp i en fil, men ska alltså fortfarande returnera en vinnare. Det ska gå att läsa av andra filer som har samma format som race-results.txt, filen är bara ett exempel.

Antaganden

Du kan inte anta att varje rad är korrekt formaterad. Returnera ett tydligt felmeddelande när den är felformaterad. Däremot kan du anta att inga av värdena innehåller ett kommatecken.

Validera värdena enligt följande regler:

- Ett namn innehåller bara bokstäver och mellanslag
- Ett ID innehåller bara siffror
- Tid är skriven på format HH:mm:ss
- Det finns tre typer av tävlingar: 1000m, eggRace, sackRace

Tänk på att det kan finnas fler saker du vill validera i ditt program som inte nämns ovan. Till exempel om två deltagare har olika namn men samma id så finns det ett fel i filen och programmet borde returnera ett felmeddelande.

Tips:

- Enkel kod är bra kod
- Små (fungerande) commits ofta
- Att fråga är positivt
- Tester är användbara