

JAVA20, Objektorienterad programmering och Java, inlämningsuppgift 2.

Gymet "Best Gym Ever" har problem. Deras datasystem har kraschat och går inte att använda. Konsulten som byggde systemet är bortrest och kan inte komma och laga systemet på flera veckor. Dessutom har någon postat på Flashback att Best Gym Evers system är sönder och att alla kan komma dit och träna gratis, vilket har gjort att en hel del skumma typer har dykt upp.

Nu ber receptionisten dig om hjälp. Hen har lyckats rota fram en fil från hårddisken med alla kunders namn, personnummer och när de senast betalade årsavgiften till gymmet. Hen undrar om du kan skriva ett program där man skriver in en kunds personnummer eller namn när personen dyker upp på gymmet, sen söker programmet i filen och visar upp om kunden är en nuvarande medlem (dvs om årsavgiften betalades för mindre än ett år sedan), en före detta kund (om årsavgiften betalades för mer än ett år sedan) eller om personen inte finns i filen och sålunda aldrig har varit medlem och är obehörig.

Posterna i filen vi läser från ser ut enligt följande:

```
7502031234, Anna Andersson
```

```
2017-05-03
```

```
8505132345, Per Persson
```

```
2015-12-29
```

```
...OSV...
```

Filen ligger uppladdad på Nackademins portal.

Dessutom undrar gymets personlige tränare om du kan fixa att när en betalande kund dyker upp så sparas det ner att kunden har varit och tränat, i en egen, annan fil. Detta för att personlige tränaren ska kunna fortsätta tracka hur ofta klienterna tränar. (Vi kan anta att om en kund kommer till gymmet tränar hen, även om personen egentligen bara sitter i loungen och kăkar banan). Tränaren behöver veta namn, personnummer och datum när kunden var på gymmet (så detta ska alltså sparas i hens fil).

Betygskriterier:

För att bli **Godkänd (G)** krävs att du hjälper både receptionisten och den personlige tränaren och uppfyller deras önskemål. Din lösning ska kunna:

- Läs personposter från fil.
- Skriva till fil.
- Läs in och parsar datum (tips: `LocalDate` är en bra klass för hantering av datum)
- Ha bra felhantering (relevanta felmeddelanden, exceptionhantering och try-with-resources)
- Koden ska vara enkelt läsbar och prydligt skriven.

För att bli **Väl Godkänd (VG)** måste lösningen uppfylla följande:

- Alla krav för att få G vara uppfyllda.
- Jobba testdrivet när du löser uppgiften.
- Minst 3 enhetstester måste finnas i lösningen (alltså minst 3 test-metoder, som var och en innehåller ett antal asserts), som uppfyller följande krav:
 - a. Dessa tester ska alltid kunna köras med samma resultat. (De får alltså inte vara beroende på t.ex. att man kör koden ett visst datum för att de ska bli gröna)

- b. Testerna ska vara relevanta för programmets funktionalitet och inte vara extremt simplistiska. (Ni kan alltså inte testa att $1==1$ eller liknande, och inte heller simpla getters och setters)
- c. De ska gå att köra testerna automatiskt (det är alltså inte ok att testerna väntar på input från användaren)
- d. Alla testfallen ska gå gröna när de körs.

Vidare måste programmet demonstreras på ett professionellt sätt. För att uppnå detta, innan du demonstrerar, gå noga igenom och kontrollera att programmet verkligen fungerar. Tänk igenom vad du ska säga under redovisningen och hur du ska visa upp din kod så att du framstår som den stabila programmerare du är. Du kommer att behöva visa både att programmet funkar och visa din kod. För att alla ska hinna demonstrera får din demonstration ta **max 7 minuter**.

Koden skall vara uppladdad på GitHub i ett publikt repo, som kursledaren har åtkomst till. Lägg upp en länk till ditt repo i den inlämnings-katalog på Portalen som är skapad för detta ändamål.

Denna uppgift görs individuellt. Lycka till!