

Tour de France 2025 Øvelse

Læringsmål:

- At kunne vælge og bruge passende kollektionstyper fra Java Collections Framework
- At kunne bruge interfaces og deres implementeringer
- At implementere unit testing

Krav:

Der skal implementeres funktionalitet, der opfylder følgende user stories:

US1 Afsluttende Hold:

Som cykelanalytiker ønsker jeg en liste over hold, som havde cyklister, der afsluttede, så jeg kan...

US2 Hold og Medlemmer:

Som cykelanalytiker ønsker jeg en liste over hold og deres afsluttende medlemmer, så jeg kan...

US3 Holdmedlemmer:

Som cykelanalytiker ønsker jeg en liste over holdmedlemmer for et bestemt hold, så jeg kan...

Et brugerinterface skal ikke implementeres. De implementerede klasser skal testes ved at implementere unit tests.

Data til øvelsen findes i csv-filen "tdffinishers2025.csv", som indeholder en liste over alle afsluttende cyklister fra Tour de France-løbet 2025.

Et eksempel vises her:

```
Rank,Rider,Rider No.,Team,Times,Gap,B,P,Year,Distance (km),Number of stages,ResultType,TotalSeconds,GapSeconds
1,TADEJ POGACAR,1,uae team emirates xrg,76h 00' 32'',-,52'',,2025,3323,21,time,273632,0
2,JONAS VINGEGAARD HANSEN,11,team visma | lease a bike,76h 04' 56'',+,00h 04' 24'',34'',,2025,3323,21,time,273896,264
3,FLORIAN LIPOWITZ,72,red bull - bora - hansgrohe,76h 11' 32'',+,00h 11' 00'',04'',,2025,3323,21,time,274292,660
4,EDGAR OSCAR ONLEY,191,team picnic postnl,76h 12' 44'',+,00h 12' 12'',04'',,2025,3323,21,time,274364,732
5,FELIX GALL,151,decathlon ag2r la mondiale team,76h 17' 44'',+,00h 17' 12'',,2025,3323,21,time,274664,1032
6,TOBIAS HALLAND JOHANNESSEN,221,uno-x mobility,76h 20' 46'',+,00h 20' 14'',,2025,3323,21,time,274846,1214
7,KÉVIN VAUQUELIN,131,arkea-b&b hotels,76h 23' 07'',+,00h 22' 35'',,2025,3323,21,time,274987,1355
8,PRIMOZ ROGLIC,71,red bull - bora - hansgrohe,76h 26' 02'',+,00h 25' 30'',,2025,3323,21,time,275162,1530
9,RFM HFAIV 31 FF ENIACATTON - FASVPOST 76h 28' 34'',+,00h 28' 02'',2025,3323,21,time,275314,1682
```

Implementering:

En klasse `Cyclist` skal implementeres, som repræsenterer en linje data i filen. Kun navn, hold og rank er nødvendigt.

En klasse `TDFFileReader` skal implementeres, som har bl.a. metode:

```
public List<Cyclist> readfile()
```

Der anbefales at bruge `Scanner` klassens `hasNext()` og `nextLine()` metoder til at indlæse linjer fra filen:

```
public class TDFFileReader implements AutoCloseable{
    private final Scanner scanner;

    public TDFFileReader(String fileName) throws FileNotFoundException {
        File file = new File(fileName);
        scanner = new Scanner(file);
    }

    public List<Cyclist> readFile() {
        // TO DO
    }

    @Override
    public void close() throws Exception {
        scanner.close();
    }
}
```

En klasse `TDFFinishers` skal implementeres, som har følgende offentlige metoder:

`public TDFFinishers(List<Cyclist> cyclists)`
constructor

`getTeams()`
som returnerer en samling af hold (en `ArrayList` skal ikke anvendes).

`getTeamsWithMembers()`
som returnerer en samling af alle hold sammen parrede med deres medlemmer.

`getTeamMembers(String team)`
som returnerer en samling af navnene på cyklisterne på holdet (en `ArrayList` skal ikke anvendes).

Overvej hvilke typer collections som skal anvendes.

Oplysninger som kan bruges til test :

Der er 23 hold.

Team "TEAM VISMA | LEASE A BIKE" har følgende medlemmer:

"EDOARDO AFFINI",
"JONAS VINGEGAARD HANSEN",
"MATTEO JORGENSEN",
"SEPP KUSS",
"SIMON PHILIP YATES",
"TIESJ BENOOT",
"VICTOR CAMPENAERTS",
"WOUT VAN AERT"