

## Opgaver lektion 15

<https://github.com/benn-christensen/Prog2Lektion15>

### Opgave 1

Person klassen indeholder en metode `printName` der ikke kompilerer.

1. Lav et nyt Interface `NameFormatter` med en passende metode, så koden kompilerer om der bliver udskrevet Mikael L. Christensen til konsollen hvis du kører main metoden.
2. Kald `printName` metoden med et lambda udtryk, så der bliver udskrevet M. Christensen til konsollen.

### Opgave 2

Brug `findFirst` metoden fra `PeopleArray` klassen til at løse følgende opgaver.

1. Find den første person i listen af personer med alderen 44
2. Find den første person i listen af personer, hvis navn starter med S
3. Find den første person i listen af personer, hvis navn indeholder mere end et 'i'
4. Find den første person i listen af personer, hvis alder er lig længden af navnet.

Lav en metode `findAll` i `PeopleArray` klassen. Metoden skal kunne kaldes med følgende statement.

```
List<Person> people = peopleArray.findAll(p -> p.getAge() > 30);
```

Brug `findAll` metoden til at løse følgende opgaver

1. Find alle personer, hvis navn indeholder et 'i'
2. Find alle personer, hvis navn starter med S
3. Find alle personer, hvis navn har længde 5
4. Find alle personer, hvis alder er mindst 6 og mindre end 40

Java har et funktionelt generisk interface `Predicate<T>` som har en metode `boolean test(T t)`, erstat `PeoplePredicate` med `Predicate<T>` i metoderne `findFirst` og `findAll`

**Ekstraopgave:** kan man lave `PeopleArray` klassen generisk, så man kan bruge den med andre klasser end `Person` klassen?

### Opgave 3

1. Udskriv en linie for hver løber med `name` og `lapTime` ved at bruge `List.forEach()` metoden med en `Consumer` lambda.
2. Som 1, men udskriv kun løberne med `lapTime < 30`.
3. Sorter løberne stigende efter `lapTime` ved at bruge `List.sort()` metoden med en `Comparator` lambda. Udskriv løberne.

### Opgave 4

1. Kør programmet og bemærk den exception, som kastes. Hvad er problemet?
2. Erstat den fejlende for-sætning med en løkke, som anvender en iterator til at fjerne løbere med `lapTime >= 40`. Udskriv listen med løberne.
3. Lav en metode, som fjerner løbere fra en liste af løbere vha. et lambda udtryk. Metoden skal implementeres vha. en iterator.  
Metodens signatur:  
`/**`  
`* Removes runners that satisfies the given filter.`  
`* Returns true, if any runner is removed.`

\*/

public static boolean removeIf(List runners, Predicate filter)

4. Brug metoden fra delopgave 3. til at fjerne løbere med lapTime  $\geq 40$ .
5. Brug metoden List.removeIf() og en Predicate lambda til at fjerne løbere med lapTime  $\geq 40$ .