



---

# Modul 6

## Brukerinput og debugger

*Uke6Xtra*

# Why Java?

## ..for example among the top programmings languages for AI

### 3. AI roles that use Java

Java remains a cornerstone in AI development, offering robust frameworks and libraries that help with the integration of AI into various platforms, from big data systems to mobile applications. Here are some of the AI job roles that use Java:

- **Big data engineer:** They use Java frameworks, which provide tools for distributed processing and machine-learning algorithms, to construct and manage AI pipelines that handle large volumes of data.
- **Enterprise AI developer:** Java libraries such as Deeplearning4j (DL4J) or H2O to create and integrate AI models into current enterprise Java systems. This enables businesses to tap into AI features without investing in more complex infrastructure.
- **Android AI developer:** They use tools like TensorFlow Lite—an optimized version of TensorFlow for mobile—to incorporate AI-driven functionalities into Android applications.
- **AI infrastructure engineer (Java backend for AI):** They develop APIs in Java to expose AI models or features to applications. They might also create pipelines and monitoring systems using Java to ensure that AI models operate smoothly in production environments.
- **Software engineer (AI integration with Java systems):** They integrate AI models or libraries like scikit-learn or spaCy into these Java-based applications through Java APIs, allowing them to use AI capabilities.



And now...for something completely different



---

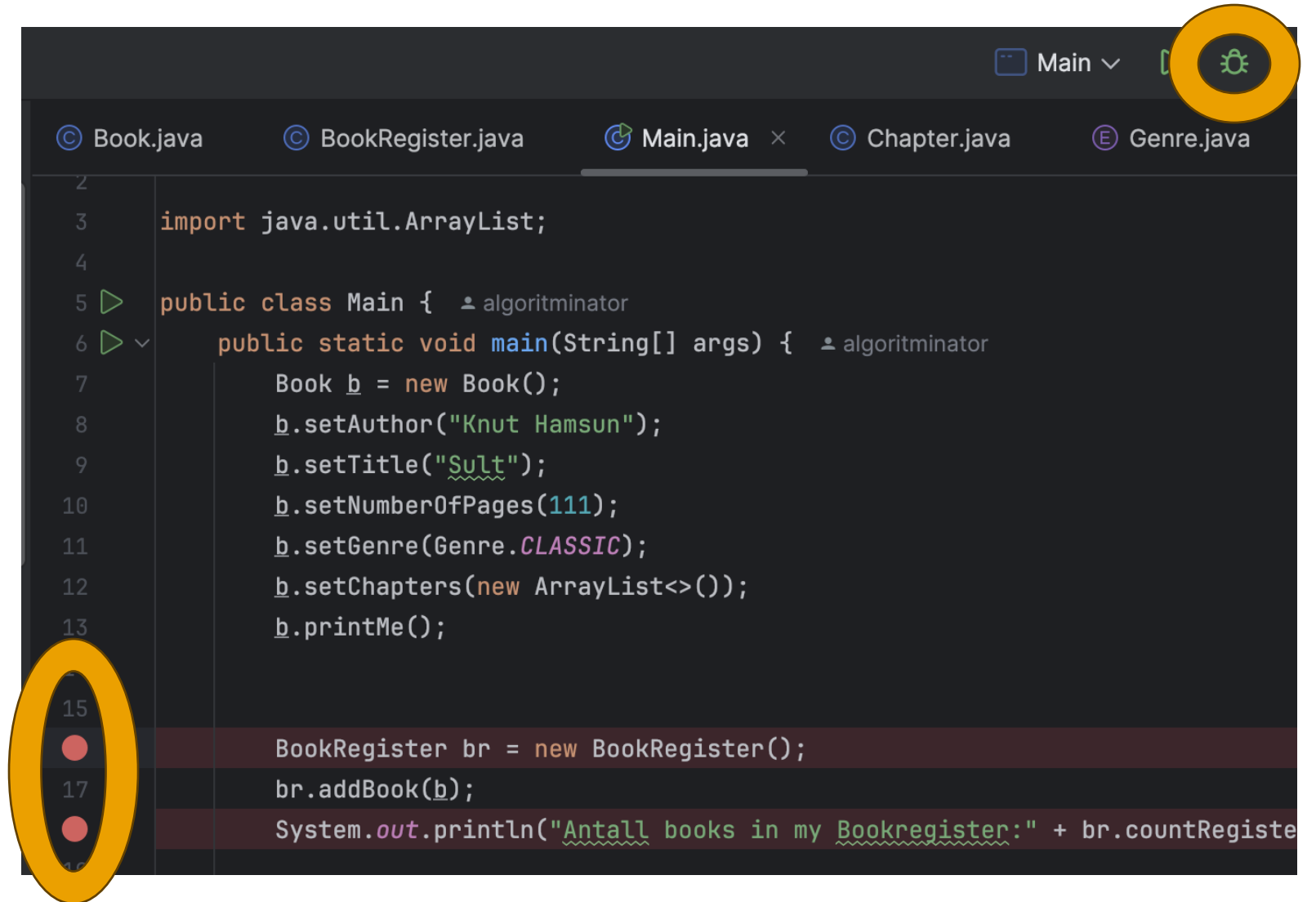
**Demo :**  
**java is**  
**pass-by-**  
**value**

COLLABORATIVE PROGRAMMING



# Feilsøking

<https://www.jetbrains.com/help/idea/debugging-your-first-java-application.html>

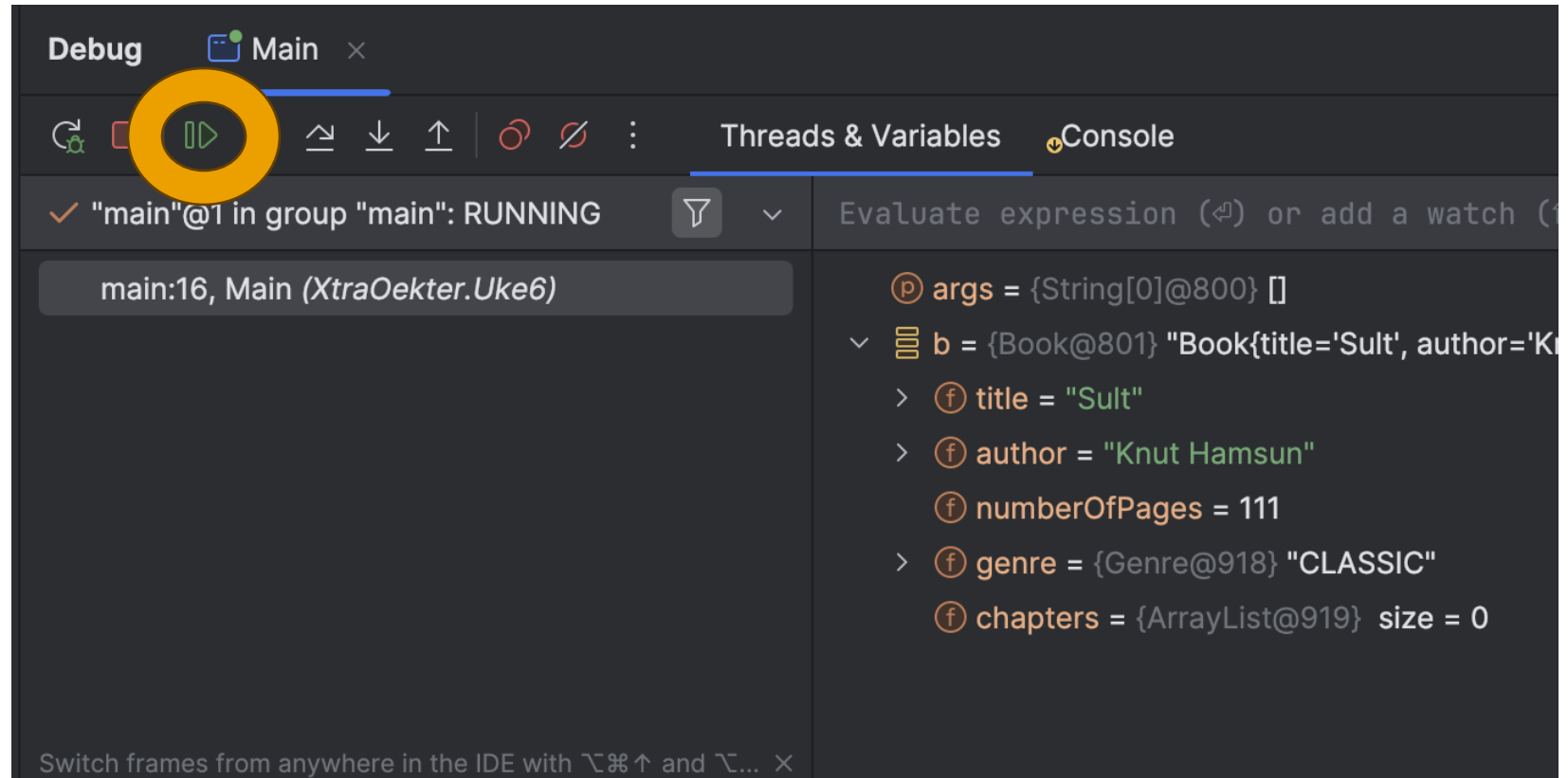


---

# Feilsøking

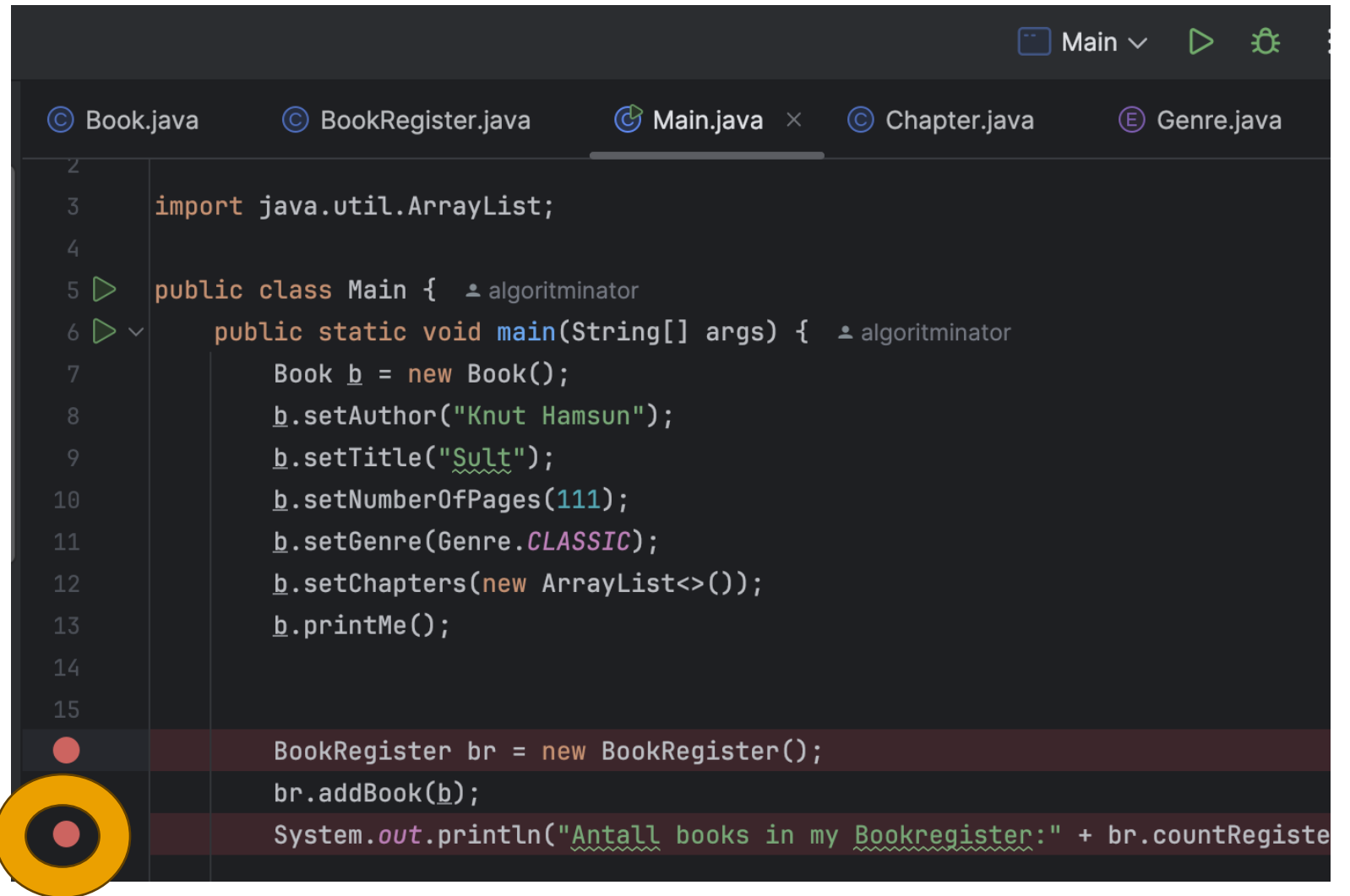
[https://www.jetbrains.com  
/help/idea/debugging-your-  
first-java-application.html](https://www.jetbrains.com/help/idea/debugging-your-first-java-application.html)

---



# Feilsøking

<https://www.jetbrains.com/help/idea/debugging-your-first-java-application.html>



The screenshot shows an IDE window with several tabs: Book.java, BookRegister.java, Main.java (active), Chapter.java, and Genre.java. The code in Main.java is as follows:

```
2  
3 import java.util.ArrayList;  
4  
5 public class Main {  
6     public static void main(String[] args) {  
7         Book b = new Book();  
8         b.setAuthor("Knut Hamsun");  
9         b.setTitle("Sult");  
10        b.setNumberOfPages(111);  
11        b.setGenre(Genre.CLASSIC);  
12        b.setChapters(new ArrayList<>());  
13        b.printMe();  
14  
15        BookRegister br = new BookRegister();  
16        br.addBook(b);  
17        System.out.println("Antall books in my Bookregister:" + br.countRegister());  
18    }  
19 }
```

A red dot on the left margin indicates a breakpoint is set on line 15. A red arrow points from the URL on the left to this breakpoint. A yellow circle with a black center is overlaid on the breakpoint icon.

---

# Feilsøking

```
Evaluate expression (⌘) or add a watch (⇧⌘)
args = {String[0]@800} []
b = {Book@801} "Book{title='Sult', author='Knut Hamsun', numberOfPages=111, genre=CLASSIC, chapters=[]}"
  title = "Sult"
  author = "Knut Hamsun"
  numberOfPages = 111
  genre = {Genre@918} "CLASSIC"
  chapters = {ArrayList@919} size = 0
br = {BookRegister@922}
  registeredBooks = {ArrayList@927} size = 1
    0 = {Book@801} "Book{title='Sult', author='Knut Hamsun', numberOfPages=111, genre=CLASSIC, chapters=[]}"
```

*[https://www.jetbrains.com  
/help/idea/debugging-your-  
first-java-application.html](https://www.jetbrains.com/help/idea/debugging-your-first-java-application.html)*

---



# Brukerinput

## Scanner – opprette

Typen til  
referansen

Referansens  
navn



```
Scanner scanner =  
new Scanner(System.in);
```



Lage  
nytt objekt



Input til  
nytt objekt

# Brukerinput

Hvilke metoder har vi?

Returverdi	Metode	Beskrivelse
boolean	hasNext()	Har noe å returnere?
boolean	hasNextInt()	Er neste en int?
int	nextInt()	Returnere neste int
String	nextLine()	Returnere neste linje
String	next()	Returnere neste som tekst

---

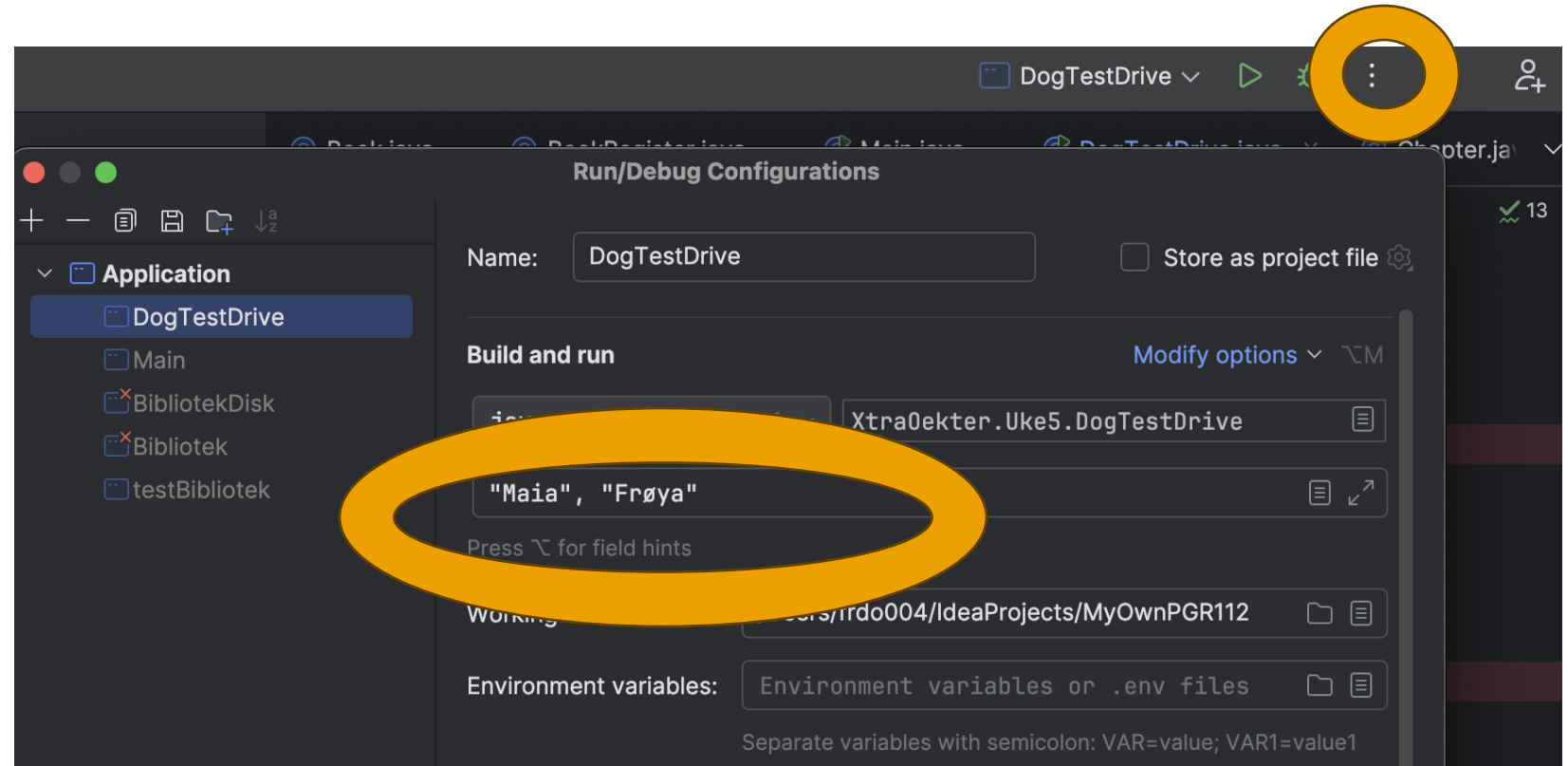
# Program-input

## Program-input

```
public static void main(String[] args){  
    ...  
}
```

# Program-input

IntelliJ injection



<https://www.jetbrains.com/help/objc/add-environment-variables-and-program-arguments.html>

---

# Gjennomgang oppgave 2 modul 5

---

## Oppgave 2

- A. Opprett et nytt prosjekt. Hent inn Book-klassen og Genre-enumen du lagde i forrige modul. Du trenger også en klasse med en main-metode.
  - B. Opprett en klasse BookRegister. Klassen skal ha en ArrayList som kan ta vare på bøker.
  - C. I BookRegister: Lag en metode for å legge inn en bok i registeret.
  - D. I BookRegister: Lag en metode som kan gi informasjon om hvor mange bøker som er lagt inn i registeret.
  - E. Opprett registeret i main-metoden, og legg inn noen bøker der. Test at innleggingen av bøker fungerer.
  - F. Lag noen metoder i registeret som kan gi informasjon om bøker i registeret:
    - Alle registrerte bøker
    - Bøker i en viss sjanger (metoden må ta imot en sjanger)
    - Bøker skrevet av en spesifikk forfatter
- Kall på metodene fra main (etter at du har lagt inn noen bøker) og sjekk om metodene fungerer tilfredsstillende.
- G. Lag en klasse Chapter som holder på informasjon om
  - Tittel på kapittelet
  - Antall sider
  - Antatt lesetid
- Utvid Book-klassen slik at den har en ArrayList som inneholder informasjon om alle kapitlene i boka.
- H. (**Vanskelig**) Lag en ny metode i bokregisteret som returnerer bøker som har en lesetid på mindre enn et gitt antall minutter.