Mentimeter

IN1010 uke 2

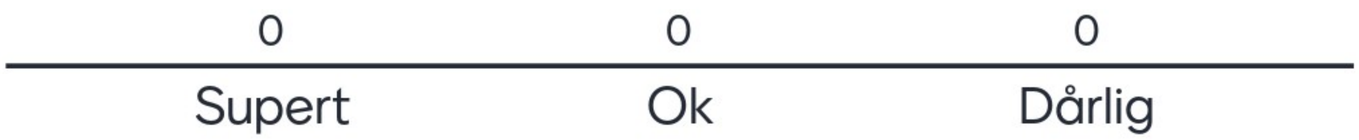
Gruppe 1

Dagens plan

- Repetisjon
- Oppsummering av ukens tema
- Tips til oblig 1
- Oppgaver

Hvordan går det? sånn egentlig

Mentimeter



Hvordan er overgangen til java?

Mentimeter

Repetisjon: kompilering og kjøring

- Kompilere er å gjøre om kode til byte, lettere sakt sjekke syntaksfeil
- "javac filnavn.java" eller "javac *.java"
- Kjøre program "java filnavn"
- Husk! trenger kun å kjøre klasser med main metode

Klasser

- 1 fil per klasse, klassenavnet skal være lik filnavn
- Forskjell på main og konstruktør
  - main er metoden som kjøres når man kjører programmet
  - alle klasser trenger ikke main
  - konsstruktør brukes typisk i klasser som skal bli objekter
  - trenger konstruktør dersom klassen tar inn parametere
- Alle klasser trenger ikke main-metode men java-program (kjørbar fil) trenger

Public eller private

- Trenger ikke public eller private foran class
- Metoder og instansvariabler kan ha private som standard
- MEN dersom dere skal bruke variabelen eller metoden utenfor klassen så bruk public
  - testene i oblig bruker variablene direkte, så derfor er variablene public

Mentimeter

```java
class Klasse{ //trenger ikke public eller private foran

    //instansvariabler. Setter alle til private.
    //MEN setter til public hvis den skal brukes i en annen klasse
    public int tall;
    public int id;
    public static int fellesVariabel = 0;
    private String tekst;
    private int[] array;

    public Klasse(int tall, String text){
        this.tall = tall; //bruk this foran instansvariabel dersom instansvariabel og parameter har samme navn

        id = fellesVariabel; //deklarerer id oppe, men initialiserer den i konstruktør
        fellesVariabel++; //neste objekt av klasse vil ha id 1 fordi det er en statisk variabel

        tekst = text;
        array = new int[5];

    }

    public void metode1(){ //Trenger IKKE return siden retur-type er void
        System.out.println(x: "metode som printer ut noe");
    }

    public String metode2(String tekst1){//MÅ ha med return av type String
        return tekst1;
    }
```

Klasse som skal brukes som et objekt. Har (typisk) instansvariabler, konstruktør og metoder

Mentimeter

```java
class Hovedprogram {
    Run | Debug
    public static void main(String[] args) {
        Klasse a = new Klasse(tall: 4, text: "Hallo");
        Klasse b = new Klasse(tall: 6, text: "hei");
        int tall = a.tall; //hovedprogram har tilgang til a sin tall-variabel siden tall er public i Klasse
        String tekst = a.tekst //!!! har ikke tilgang til tekst fordi tekst er private. Får feil

        System.out.println(a.id); //printer 0
        System.out.println(b.id); //printer 1

        printHalla(); //trenger ikke å skrive Hovedprogram.printHalla() siden metoden er i samme klasse
    }

    public static void printHalla(){ //en static metode
        //en metode som kan brukes i Hovedprogram sin main-metode
        System.out.println(x: "halla");
    }
}
```

Hovedprogram med main og static metoder

Mentimeter

Løkker

- Vanlig for-løkke
  - brukes når vi vil gå fra et start tall til slutt tall. eks 0-10
  - kan bruke tallet i til å løpe igjennom array
- For-each løkke
  - brukes når vi vil gå igjennom alle elementer i en liste.
  - får ikke indeksen da som i en vanlig løkke men objektet/elementet i lista

```java
String array[] = {"ost", "egg", "melk"};
for (int i = 0; i < array.length; i++) {
    System.out.println(array[i]);
}

for (String string : array) {
    System.out.println(string);
}
```

Array

- int[] array = new int[5]
  - vanlig array med fast størrelse.
  - tomme plasser er 0 dersom listen er int, ellers er tomme plasser null
  - int[] array = {1,2,3}
  - har ingen metoder
- Arraylist<Integer> liste = new Arraylist<>()
  - merk skriver Integer i stedet for int
  - dynamisk. ingen fast størrelse
  - importere fra java.util.ArrayList
  - har flere metoder som add(element), size(), get(indeks)

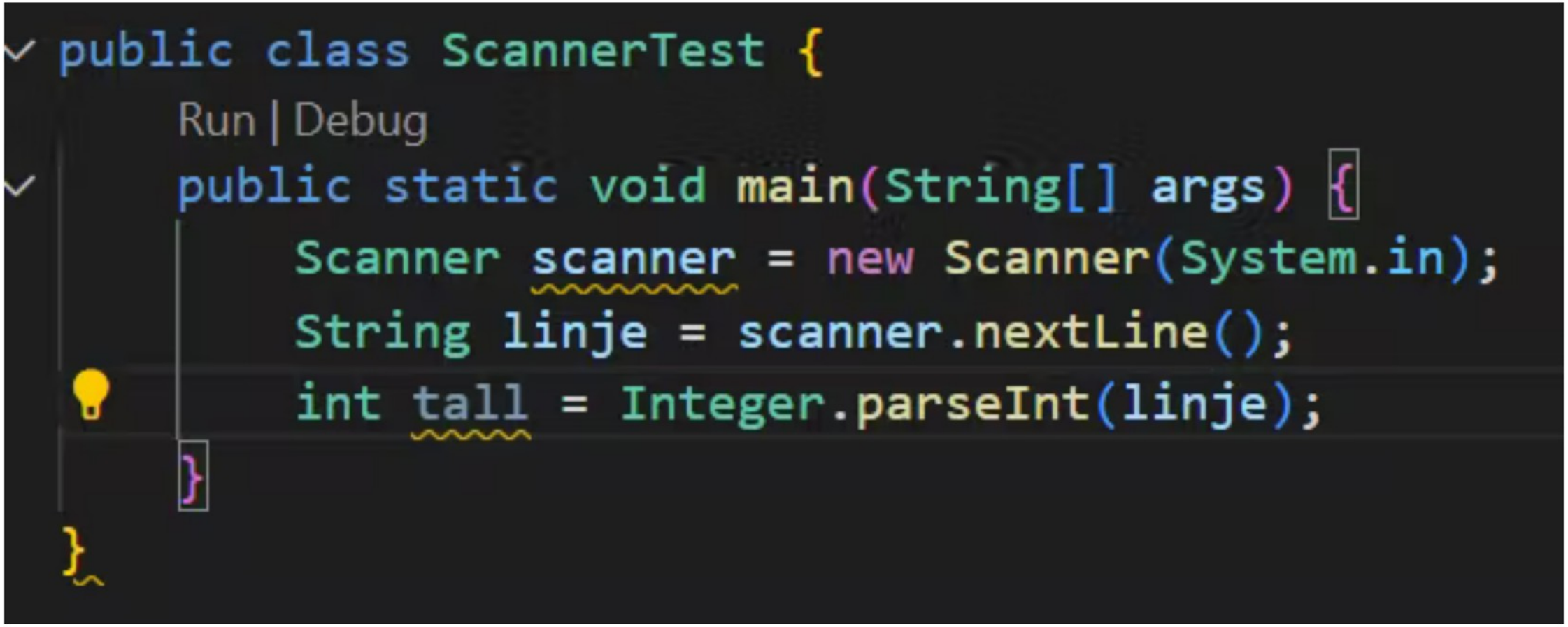
Mentimeter

Oppsummering av ukens tema

Scanner

- må importeres fra java.util
- Konstruktør i Scanner kan ta inn filnavn, System.in (terminal) eller string
- bruker next() for neste ord i filen/terminal/streng
- hasNext() for å sjekke om det er flere ord i fil/terminal/streng
- vi har nextLine(), nextInt(), nextBoolean() og mange fler

Mentimeter

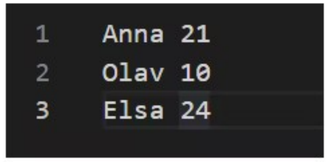


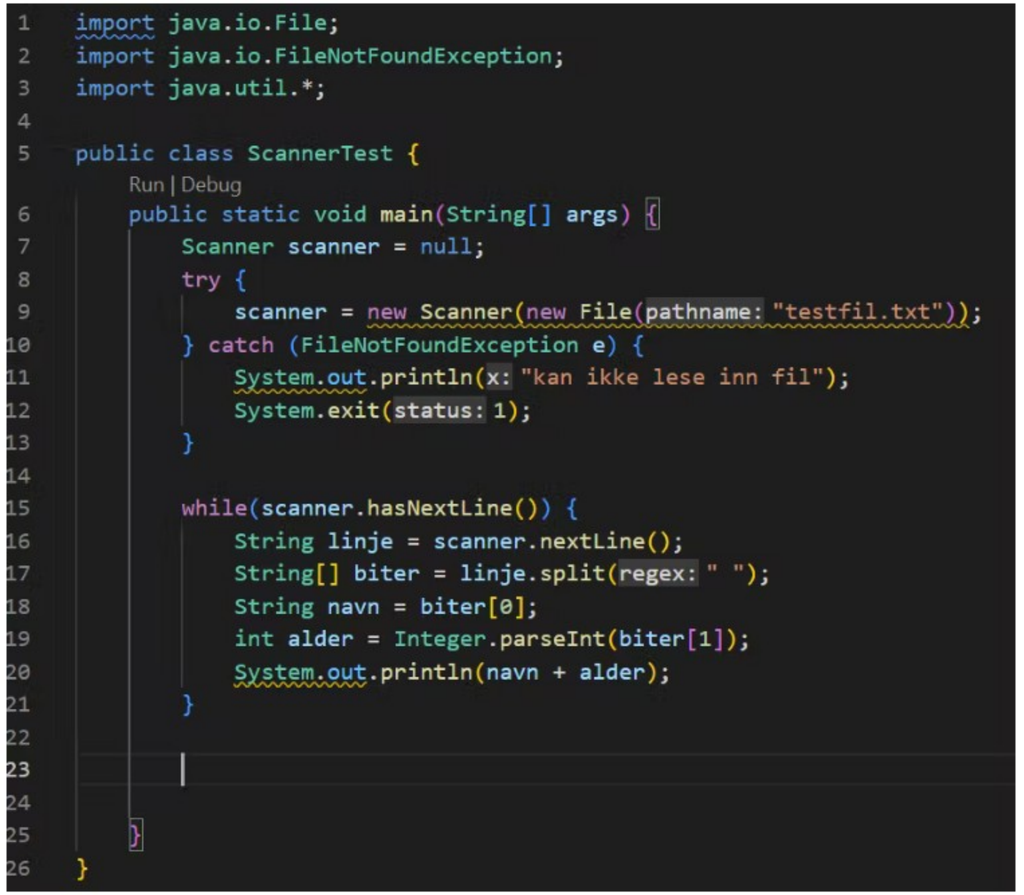
Lese et tall fra terminal og gjøre string om til int

Mentimeter

Lese fra fil

- kan hende filen som skal bli lest ikke finnes
- Må derfor si til systemet hva den skal gjøre
- beste er å ha try-catch for å gi feilmelding dersom filen ikke finns
- til venstre er testfil.txt





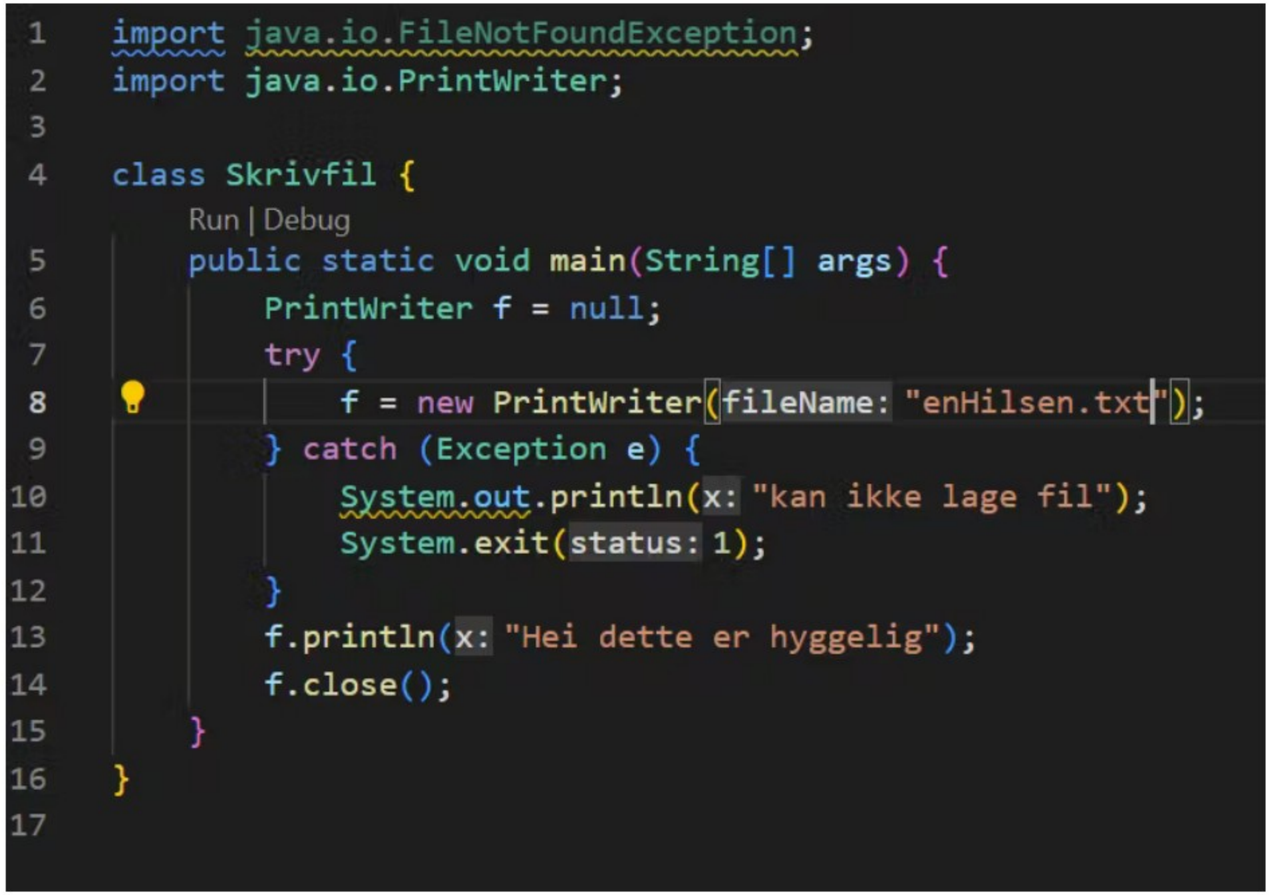
```java
import java.io.File;
import java.io.FileNotFoundException;
import java.util.*;

public class ScannerTest {
    Run | Debug
    public static void main(String[] args) {
        Scanner scanner = null;
        try {
            scanner = new Scanner(new File(pathname: "testfil.txt"));
        } catch (FileNotFoundException e) {
            System.out.println(x: "kan ikke lese inn fil");
            System.exit(status: 1);
        }

        while(scanner.hasNextLine()) {
            String linje = scanner.nextLine();
            String[] biter = linje.split(regex: " ");
            String navn = biter[0];
            int alder = Integer.parseInt(biter[1]);
            System.out.println(navn + alder);
        }


    }
}
```

program som leser fil og printer ut navn og alder som står i filen. glemte scanner.close()



skrive til fil

Håndtere feil

- Noen ganger krever java at vi håndterer feil som oppstår under kjøring
- kan fanges og håndteres med try .. catch som sett tidligere
- throws er en annen måte men dårligere.

Tips til oblig 1

- start tidlig. NÅ
- Les gjennom hele oppgaven nøye
- Husk å kompiler og kjør testene jevnlig
  - husk testprogrammet har gitte navn på variablene. For eksempel skal Celle ha variabel levende for å vise om den er død eller i live
  - testprogrammet forutser at dere har skrevet alle metodene. For å teste en og en metode så kan dere kommentere ut testene
  - som dere ikke har lagd metode for ennå (vis eksempel)
- Bruk metodene dere har laget! det er mulig å bruke en metode fra samme klasse i en annen metode
- levende er boolean. if (levende = true) er stygt. Skriv heller if (levende)
- Spør gruppelærer eller andre studenter, og bruk lab

Array i java er...

Mentimeter

Hvilken er riktig syntaks?

Mentimeter

0	0	0	0	0
liste = new array[6]	int liste = new int(6)	int[] liste = new int[6]	liste = new int[6]()	int[] liste = {1,2,3, 4,5,6}
✗	✗	✓	✗	✓

For å lese inn fil trenger jeg

Mentimeter

Diskusjonsoppgave

Skrive ned så mange forskjeller mellom lister i python og arrays i java som dere kommer på (mellom 30-60 sek). Etterpå skrive ned likheter. Diskuter og sammenlign i mindre grupper før vi tar det i plenum.

Skrive ned så mange forskjeller mellom lister i python og arrays i java som dere kommer på. Etterpå skrive ned likheter.