GRUPPE 1

Velkommen til
IN1010 - Uke 1

Plan for dagen

- Tilgangsnivåer
- Static
- Lese og skrive til fil
- Håndtere unntak (try/catch)
- Wrapper klasser
- Oppgaver

Tilgangsnivåer

```
Run | Debug
public static void main(String[] args) {

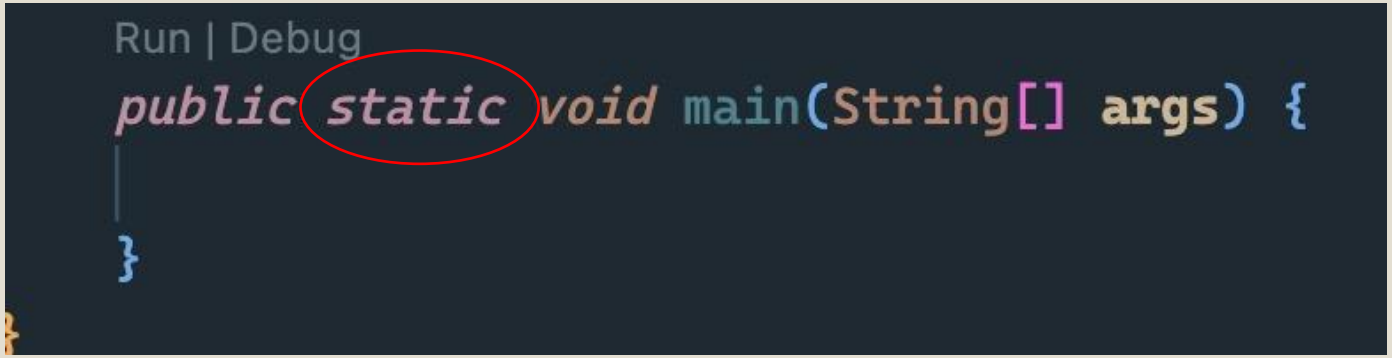
}
```

```
private class eksempel {

}
```

- Sier noe om hvor klasser/metoder/variabler er tilgjengelig fra

Static



- Metode/variabel tilhører klassen og ikke et objekt

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

public class LeseFil {

    Run | Debug
    public static void main(String[] args) throws FileNotFoundException {

        String tekst = "";
        File fil = new File("tekst.txt");
        Scanner scanner = new Scanner(fil);

        while (scanner.hasNext()) {
            tekst = scanner.nextLine();
            System.out.println(tekst);
        }
        scanner.close();
    }
}
```

Lese fra fil

- Husk import
- Må kaste eller håndtere feil
- Husk å lukk scanner

Skrive til fil

- Husk import
- Må kaste eller håndtere feil
- Husk å lukke PrintWriter

```java
import java.io.PrintWriter;
import java.io.FileNotFoundException;

public class SkriveFil {
    Run | Debug
    public static void main(String[] args) throws
    FileNotFoundException {
        PrintWriter utfil = new PrintWriter("utfil.txt");

        utfil.println("Tekst her");
        utfil.close();
    }
}
```

Håndtering av unntak (try/catch)

- Med throws må feil kastes helt opp til main metoden
- Putt try rundt der feilen kan oppstå
- Programmet fortsetter å kjøre etter catch

```java
public class TryCatch {
    Run | Debug
    public static void main(String[] args) throws Exception {

    }
}
```

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

public class TryCatch {
    Run | Debug
    public static void main(String[] args) {
        String tekst = "";
        File fil = new File("tekst.txt");

        try {
            Scanner scanner = new Scanner(fil);

            while (scanner.hasNext()) {
                tekst = scanner.nextLine();
                System.out.println(tekst);
            }
            scanner.close();
        } catch (FileNotFoundException e) {
            System.out.println("Fant ikke fil");
        }
    }
}
```

Wrapper klasser

- Brukes i stedet for de primitive datatypene for å kalle på metoder og for beholdere

```java
import java.util.Scanner;

public class Wrapper {
    Run | Debug
    public static void main(String[] args) {
        int tall1;
        int tall2;

        Scanner scanner = new Scanner(System.in);

        tall1 = Integer.parseInt(scanner.nextLine());
        tall2 = Integer.parseInt(scanner.nextLine());

        System.out.println(tall1 + tall2);

        scanner.close();
    }
}
```

Oppgaver