IN1010 Gruppetime – Uke 12

Dagens tema: Rekursjon

Hva er rekursjon?

- Enkelt forklart: En *rekursiv* metode er en metode som *kaller på seg selv*
- En rekursiv metode trenger et *basistilfelle* og et *rekursjonssteg* (kan så klart være flere)
  - Rekursjonssteg: Gjør no endringer og kall på metoden igjen, med de nye endringene
  - Basistilfelle: Noe som må oppnås før metoden kan slutte å kalle på seg selv. Når vi når basistilfellet så slutter metoden å kalle på seg selv
- En rekursiv metode kan minne om while-løkker
  - Noe gjøres helt til basistilfellet er nådd
  - Metoder som kan gjøres iterativt kan gjøres rekursivt, og motsatt (for det meste)

```java
class Rekursjonstest {
    public static void main(String[] args) {
            tellNed(5);
    }
    public static void tellNed(int n){
            if (n < 0) return;
            System.out.println(n);
            tellNed(n-1);
    }
}
```

```java
class Iterasjonstest {
    public static void main(String[] args) {
            tellNed(5);
    }
    public static void tellNed(int n) {
            while(n >= 0) {
                System.out.println(n);
                n--;
            }
    }}
```

Call stack

- Når man kaller på en metode, legges metodekallet på en *call stack*
  - Repetisjon: En stack er en datastruktur hvor det siste elementet som legges til, er det første som fjernes
- En rekursiv metode vil legge metodekallene sine på call stacken, og når vi har nådd basistilfellet vil vi gå «tilbake» i call stacken (alle elementene poppes fra stacken)
  - Det siste metodekallet poppes først og det første metodekallet poppes sist

Stack

Output

```java
public static void main(String[] args) {
    tellOpp(5);
}

public static void tellOpp(int tall) {
    if (tall == 0) {
        return;
    }
    tellOpp(tall - 1);
    System.out.println(tall);
}
```

Stack

Output



```
public static void main(String[] args) {
    tellOpp(5);
}

public static void tellOpp(int tall) {
    if (tall == 0) {
        return;
    }
    tellOpp(tall - 1);
    System.out.println(tall);
}
```

Stack

Output



tellOpp(5)



```
public static void main(String[] args) {
    tellOpp(5);
}

public static void tellOpp(int tall) {
    if (tall == 0) {
        return;
    }
    tellOpp(tall - 1);
    System.out.println(tall);
}
```

Stack

Output



tellOpp(5)

```java
public static void main(String[] args) {
    tellOpp(5);
}

public static void tellOpp(int tall) {
    if (tall == 0) {
        return;
    }
    tellOpp(tall - 1);
    System.out.println(tall);
}
```



```java
public static void main(String[] args) {
    tellOpp(5);
}

public static void tellOpp(int tall) {
    if (tall == 0) {
        return;
    }
    tellOpp(tall - 1);
    System.out.println(tall);
}
```

Stack

tellOpp(5)

tellOpp(4)

Output

```java
public static void main(String[] args) {
    tellOpp(5);
}

public static void tellOpp(int tall) {
    if (tall == 0) {
        return;
    }
    tellOpp(tall - 1);
    System.out.println(tall);
}
```

Stack

tellOpp(5)

tellOpp(4)

Output

```java
public static void main(String[] args) {
    tellOpp(5);
}

public static void tellOpp(int tall) {
    if (tall == 0) {
        return;
    }
    tellOpp(tall - 1);
    System.out.println(tall);
}
```

Stack

tellOpp(5)

tellOpp(4)

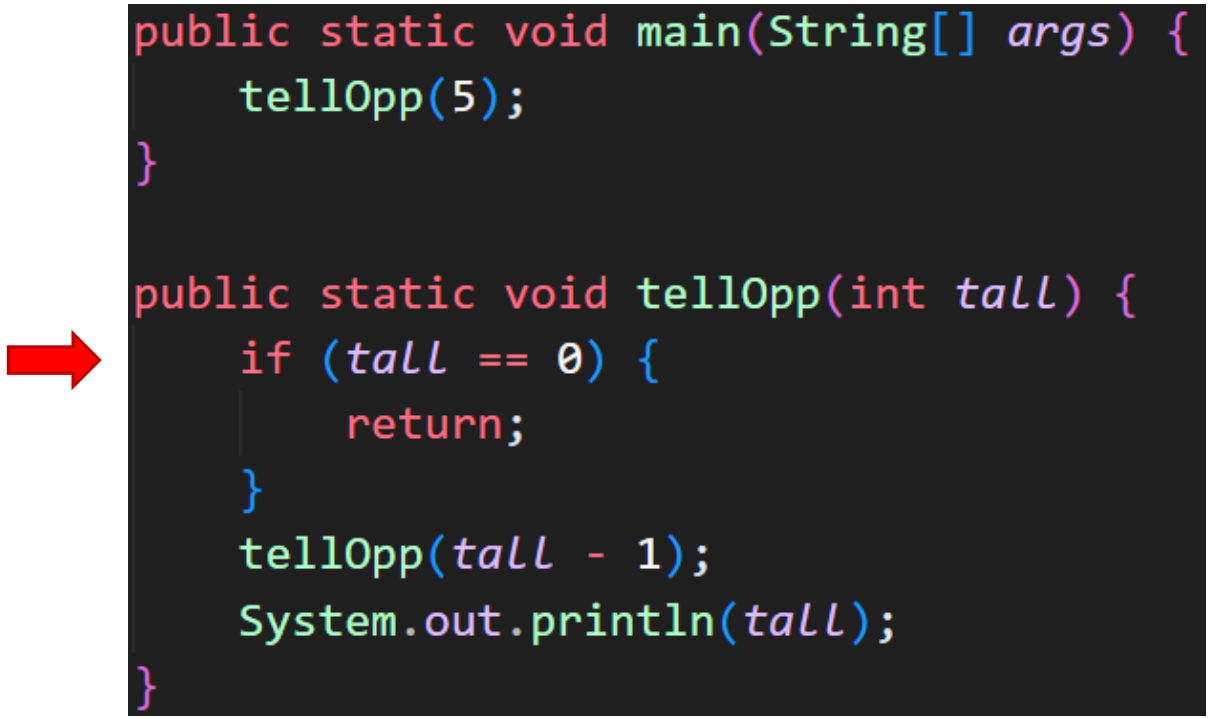
tellOpp(3)

Output

```java
public static void main(String[] args) {
    tellOpp(5);
}

public static void tellOpp(int tall) {
    if (tall == 0) {
        return;
    }
    tellOpp(tall - 1);
    System.out.println(tall);
}
```

Stack	Output
tellOpp(5)	
tellOpp(4)	
tellOpp(3)	



```java
public static void main(String[] args) {
    tellOpp(5);
}

public static void tellOpp(int tall) {
    if (tall == 0) {
        return;
    }
    tellOpp(tall - 1);
    System.out.println(tall);
}
```

Stack

tellOpp(5)

tellOpp(4)

tellOpp(3)

tellOpp(2)

Output

```java
public static void main(String[] args) {
    tellOpp(5);
}

public static void tellOpp(int tall) {
    if (tall == 0) {
        return;
    }
    tellOpp(tall - 1);
    System.out.println(tall);
}
```

Stack

tellOpp(5)

tellOpp(4)

tellOpp(3)

tellOpp(2)

Output

```java
public static void main(String[] args) {
    tellOpp(5);
}

public static void tellOpp(int tall) {
    if (tall == 0) {
        return;
    }
    tellOpp(tall - 1);
    System.out.println(tall);
}
```

Stack

tellOpp(5)

tellOpp(4)

tellOpp(3)

tellOpp(2)

tellOpp(1)

Output

```java
public static void main(String[] args) {
    tellOpp(5);
}

public static void tellOpp(int tall) {
    if (tall == 0) {
        return;
    }
    tellOpp(tall - 1);
    System.out.println(tall);
}
```

Stack

tellOpp(5)

tellOpp(4)

tellOpp(3)

tellOpp(2)

tellOpp(1)

Output

```java
public static void main(String[] args) {
    tellOpp(5);
}

public static void tellOpp(int tall) {
    if (tall == 0) {
        return;
    }
    tellOpp(tall - 1);
    System.out.println(tall);
}
```

Stack

tellOpp(5)

tellOpp(4)

tellOpp(3)

tellOpp(2)

tellOpp(1)

tellOpp(0)

Output

```java
public static void main(String[] args) {
    tellOpp(5);
}

public static void tellOpp(int tall) {
    if (tall == 0) {
        return;
    }
    tellOpp(tall - 1);
    System.out.println(tall);
}
```



Stack	Output
tellOpp(5)	
tellOpp(4)	
tellOpp(3)	
tellOpp(2)	
tellOpp(1)	
tellOpp(0)	



```java
public static void main(String[] args) {
    tellOpp(5);
}

public static void tellOpp(int tall) {
    if (tall == 0) {
        return;
    }
    tellOpp(tall - 1);
    System.out.println(tall);
}
```

Stack

tellOpp(5)

tellOpp(4)

tellOpp(3)

tellOpp(2)

tellOpp(1)

Output

```java
public static void main(String[] args) {
    tellOpp(5);
}

public static void tellOpp(int tall) {
    if (tall == 0) {
        return;
    }
    tellOpp(tall - 1);
    System.out.println(tall);
}
```

Stack	Output
tellOpp(5)	1
tellOpp(4)	
tellOpp(3)	
tellOpp(2)	
tellOpp(1)	

```java
public static void main(String[] args) {
    tellOpp(5);
}

public static void tellOpp(int tall) {
    if (tall == 0) {
        return;
    }
    tellOpp(tall - 1);
    System.out.println(tall);
}
```

Stack	Output
tellOpp(5)	1
tellOpp(4)	
tellOpp(3)	
tellOpp(2)	

```java
public static void main(String[] args) {
    tellOpp(5);
}

public static void tellOpp(int tall) {
    if (tall == 0) {
        return;
    }
    tellOpp(tall - 1);
    System.out.println(tall);
}
```

Stack	Output
tellOpp(5)	1
tellOpp(4)	2
tellOpp(3)	
tellOpp(2)	

```java
public static void main(String[] args) {
    tellOpp(5);
}

public static void tellOpp(int tall) {
    if (tall == 0) {
        return;
    }
    tellOpp(tall - 1);
    System.out.println(tall);
}
```

Stack	Output
tellOpp(5)	1
tellOpp(4)	2
tellOpp(3)	

```java
public static void main(String[] args) {
    tellOpp(5);
}

public static void tellOpp(int tall) {
    if (tall == 0) {
        return;
    }
    tellOpp(tall - 1);
    System.out.println(tall);
}
```

Stack	Output
tellOpp(5)	1
tellOpp(4)	2
tellOpp(3)	3

```java
public static void main(String[] args) {
    tellOpp(5);
}

public static void tellOpp(int tall) {
    if (tall == 0) {
        return;
    }
    tellOpp(tall - 1);
    System.out.println(tall);
}
```

Stack	Output
tellOpp(5)	1
tellOpp(4)	2
	3

```java
public static void main(String[] args) {
    tellOpp(5);
}

public static void tellOpp(int tall) {
    if (tall == 0) {
        return;
    }
    tellOpp(tall - 1);
    System.out.println(tall);
}
```

Stack	Output
tellOpp(5)	1
tellOpp(4)	2
	3
	4

```java
public static void main(String[] args) {
    tellOpp(5);
}

public static void tellOpp(int tall) {
    if (tall == 0) {
        return;
    }
    tellOpp(tall - 1);
    System.out.println(tall);
}
```

Stack	Output
tellOpp(5)	1
	2
	3
	4

```java
public static void main(String[] args) {
    tellOpp(5);
}

public static void tellOpp(int tall) {
    if (tall == 0) {
        return;
    }
    tellOpp(tall - 1);
    System.out.println(tall);
}
```

Stack	Output
tellOpp(5)	1
	2
	3
	4
	5

```java
public static void main(String[] args) {
    tellOpp(5);
}

public static void tellOpp(int tall) {
    if (tall == 0) {
        return;
    }
    tellOpp(tall - 1);
    System.out.println(tall);
}
```

Stack

Output

1

2

3

4

5

```java
public static void main(String[] args) {
    tellOpp(5);
}

public static void tellOpp(int tall) {
    if (tall == 0) {
        return;
    }
    tellOpp(tall - 1);
    System.out.println(tall);
}
```

Stack

Output

1

2

3

4

5

TADA!

Livekode

Binært søketre