IN1010 gruppetime

04.03.2026

Plan for dagen

Quizlet!

Mer om beholdere

Dobbeltlenket lister

Iterable

Quizlet Live!

Beholdere

```
public class Spilleliste {}
```

```
public class Spilleliste {

    private class Sang {

    }

}
```

Er Spilleliste en generisk klasse??

Nei. Den kan bare holde på sanger.

```java
public class Spilleliste {

    private class Sang {
        String tittel;
        String artist;
        int lengde;
        private Sang(String tittel, String artist, int lengde) {
            this.tittel = tittel;
            this.artist = artist;
            this.lengde = lengde;
        }

    }

}
```

```java
public class Spilleliste {

    Sang første = null;
    Sang siste = null;

    private class Sang {
        String tittel;
        String artist;
        int lengde;

        Sang neste = null;
        Sang forrige = null;

        private Sang(String tittel, String artist, int lengde) {
            this.tittel = tittel;
            this.artist = artist;
            this.lengde = lengde;
        }
    }
}
```

La oss legge til noen metoder!

1. Legge til sanger
2. Finne sanger
3. Fjerne sanger
4. Finne total lengde på spillelisten

```
public class Spilleliste {

    Sang første = null;
    Sang siste = null;

    private class Sang {
        String tittel;
        String artist;
        int lengde;

        Sang neste = null;
        Sang forrige = null;

        private Sang(String tittel, String artist, int lengde) {
            this.tittel = tittel;
            this.artist = artist;
            this.lengde = lengde;
        }
    }
}
```

1. Legge til sanger

```
public class Spilleliste {

    Sang første = null;
    Sang siste = null;

    private class Sang {
        String tittel;
        String artist;
        int lengde;

        Sang neste = null;
        Sang forrige = null;

        private Sang(String tittel, String artist, int lengde) {
            this.tittel = tittel;
            this.artist = artist;
            this.lengde = lengde;
        }
    }

    //Her legger vi til metoder
}
```

```java
public void leggTilSang(String tittel, String artist, int lengde) {



}
```

```java
public void leggTilSang(String tittel, String artist, int lengde) {

    Sang nySang = new Sang(tittel, artist, lengde);


}
```

```java
public void leggTilSang(String tittel, String artist, int lengde) {

    Sang nySang = new Sang(tittel, artist, lengde);

    //Håndtering hvis listen er tom:

    //Håndtering hvis det bare er ett element i listen:

    //Håndtering av alt annet:

}
```

```java
public void leggTilSang(String tittel, String artist, int lengde) {

    Sang nySang = new Sang(tittel, artist, lengde);

    //Håndtering hvis listen er tom:
    if (første == null) {
        første = nySang;
        siste = nySang;
        return;
    }

}
```

```java
public void leggTilSang(String tittel, String artist, int lengde) {

    Sang nySang = new Sang(tittel, artist, lengde);

    //Håndtering hvis listen er tom:
    if (første == null) {
        første = nySang;
        siste = nySang;
        return;
    }

}
```

```java
public void leggTilSang(String tittel, String artist, int lengde) {

    Sang nySang = new Sang(tittel, artist, lengde);

    //Håndtering hvis listen er tom:
    if (første == null) {
        første = nySang;
        siste = nySang;
        return;
    }

    //Håndtering hvis det bare er ett element i listen:
    if (første == siste) {
        første.neste = nySang;
        nySang.forrige = første;
        siste = nySang;
        return;
    }

    //Håndtering av alt annet:

}
```

```java
public void leggTilSang(String tittel, String artist, int lengde) {

    Sang nySang = new Sang(tittel, artist, lengde);

    //Håndtering hvis listen er tom:
    if (første == null) {
        første = nySang;
        siste = nySang;
        return;
    }

    //Håndtering hvis det bare er ett element i listen:
    if (første == siste) {
        første.neste = nySang;
        nySang.forrige = første;
        siste = nySang;
        return;
    }

    //Håndtering av alt annet:
    siste.neste = nySang;
    nySang.forrige = siste;
    siste = nySang;
}
```

2. Finne sang

```
public Sang finnSang(String tittel) {

    //Håndtering hvis listen er tom:

    //Iterere gjennom listen:

}
```

```java
public Sang finnSang(String tittel) {

    //Håndtering hvis listen er tom:
    if (første == null) {
        System.out.println(x: "Listen er tom!");
        return null;
    }

    //Iterere gjennom listen:
}
```

```java
public Sang finnSang(String tittel) {

    //Håndtering hvis listen er tom:
    if (første == null) {
        System.out.println(x: "Listen er tom!");
        return null;
    }

    //Iterere gjennom listen:
    Sang currSang = første;

    while (currSang != null) {
        if (currSang.tittel == tittel) {
            return currSang;
        }
        currSang = currSang.neste;
    }
    System.out.println(x: "Fant ikke sangen...");
    return null;
}
```

3. Fjern sang

```
public Sang fjernSang(String tittel) {

    //Håndtering hvis sangen ikke er i listen:

    //Håndtering hvis det er den den eneste sangen

    //Håndtering hvis det er den første

    //Håndtering hvis det er den siste

    //Ellers...

}
```

```java
public Sang fjernSang(String tittel) {

    Sang sangSomSkalFjernes = finnSang(tittel);

    //Håndtering hvis sangen ikke er i listen:
    if (sangSomSkalFjernes == null) {
        return null;
    }

    //Håndtering hvis det er den den eneste sangen

    //Håndtering hvis det er den første

    //Håndtering hvis det er den siste

    //Ellers...

}
```

```java
public Sang fjernSang(String tittel) {

    Sang sangSomSkalFjernes = finnSang(tittel);

    //Håndtering hvis sangen ikke er i listen:
    if (sangSomSkalFjernes == null) {
        return null;
    }

    //Håndtering hvis det er den den eneste sangen
    if (sangSomSkalFjernes == første && sangSomSkalFjernes == siste) {
        første = null;
        siste = null;
        return sangSomSkalFjernes;
    }
    //Håndtering hvis det er den første

    //Håndtering hvis det er den siste

    //Ellers...

}
```

```java
public Sang fjernSang(String tittel) {

    Sang sangSomSkalFjernes = finnSang(tittel);

    //Håndtering hvis sangen ikke er i listen:
    if (sangSomSkalFjernes == null) {
        return null;
    }

    //Håndtering hvis det er den den eneste sangen
    if (sangSomSkalFjernes == første && sangSomSkalFjernes == siste) {
        første = null;
        siste = null;
        return sangSomSkalFjernes;
    }
    //Håndtering hvis det er den første
    if (sangSomSkalFjernes == første) {
        sangSomSkalFjernes.neste.forrige = null;
        første = sangSomSkalFjernes.neste;
        sangSomSkalFjernes.neste = null;
        return sangSomSkalFjernes;
    }

    //Håndtering hvis det er den siste

    //Ellers...
```

```java
public Sang fjernSang(String tittel) {

    Sang sangSomSkalFjernes = finnSang(tittel);

    //Håndtering hvis sangen ikke er i listen:
    if (sangSomSkalFjernes == null) {
        return null;
    }

    //Håndtering hvis det er den den eneste sangen
    if (sangSomSkalFjernes == første && sangSomSkalFjernes == siste) {
        første = null;
        siste = null;
        return sangSomSkalFjernes;
    }
    //Håndtering hvis det er den første
    if (sangSomSkalFjernes == første) {
        sangSomSkalFjernes.neste.forrige = null;
        første = sangSomSkalFjernes.neste;
        sangSomSkalFjernes.neste = null;
        return sangSomSkalFjernes;
    }

    //Håndtering hvis det er den siste
    if (sangSomSkalFjernes == siste) {
        sangSomSkalFjernes.forrige.neste = null;
        siste = sangSomSkalFjernes.forrige;
        sangSomSkalFjernes.forrige = null;
        return sangSomSkalFjernes;
    }

    //Ellers...

}
```

```java
public Sang fjernSang(String tittel) {

    Sang sangSomSkalFjernes = finnSang(tittel);

    //Håndtering hvis sangen ikke er i listen:
    if (sangSomSkalFjernes == null) {
        return null;
    }

    //Håndtering hvis det er den den eneste sangen
    if (sangSomSkalFjernes == første && sangSomSkalFjernes == siste) {
        første = null;
        siste = null;
        return sangSomSkalFjernes;
    }
    //Håndtering hvis det er den første
    if (sangSomSkalFjernes == første) {
        sangSomSkalFjernes.neste.forrige = null;
        første = sangSomSkalFjernes.neste;
        sangSomSkalFjernes.neste = null;
        return sangSomSkalFjernes;
    }

    //Håndtering hvis det er den siste
    if (sangSomSkalFjernes == siste) {
        sangSomSkalFjernes.forrige.neste = null;
        siste = sangSomSkalFjernes.forrige;
        sangSomSkalFjernes.forrige = null;
        return sangSomSkalFjernes;
    }

    //Ellers...
    sangSomSkalFjernes.forrige.neste = sangSomSkalFjernes.neste;
    sangSomSkalFjernes.neste.forrige = sangSomSkalFjernes.forrige;
    sangSomSkalFjernes.neste = null;
    sangSomSkalFjernes.forrige = null;
    return sangSomSkalFjernes;
}
```

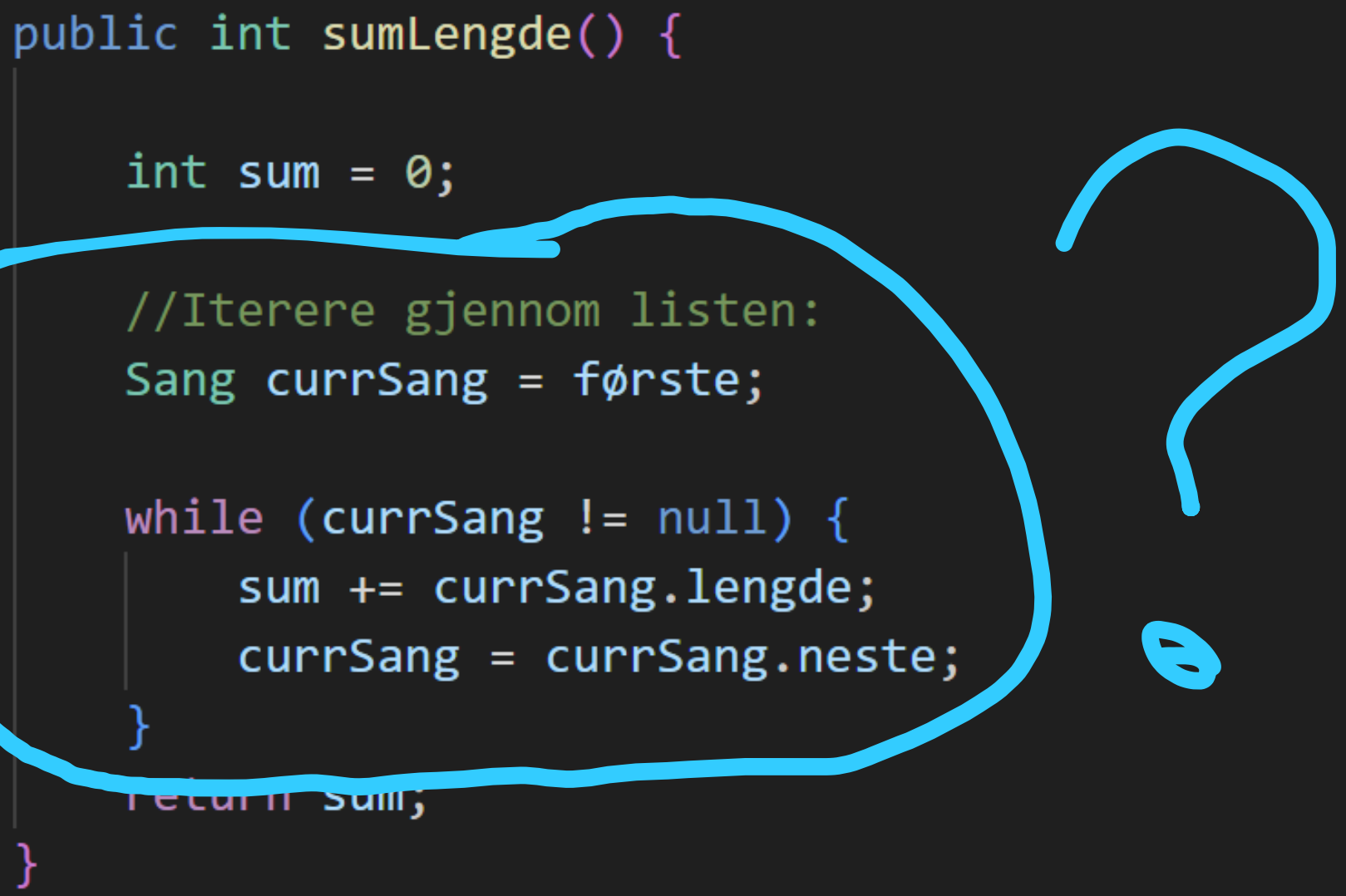
4. Finne total lengde

```
public int sumLengde() {

    int sum = 0;

    //Iterere gjennom listen:

}
```


public int sumLengde() {
int sum = 0;
//Iterere gjennom listen:
Sang currSang = første;
while (currSang != null) {
sum += currSang.lengde;
currSang = currSang.neste;
}
}

La oss implementere Iterable!

**Etter pausen*

Legge til og *fjerne* i dobbeltlenket liste:

Legge til:

1. Håndter om listen er tom
2. Håndter om listen bare har ett element
3. Ellers...

Fjerne:

**Tips: ha en metode for å hente elementet*

1. Håndter om elementet **er** i listen
2. Håndter om det er det eneste elementet
3. Håndter om du fjerner *første*
4. Håndter om du fjerner *siste*
5. Ellers...

La oss implementere Iterable!

Hvorfor??

Krav for Iterable-klasser

Beholder-klassen:

1. Ha en metode for å returnere en instans av et iterator-objekt

Iterator-klassen:

1. hasNext()
2. next()

Takk for meg

Jeg skal vise oppgaver her på skjermen!