Velkommen!

•••

Johanna
johannph på mattermost
johannph@uio.no på mail!

Kort: Praktisk informasjon

- Undervisningstilbud
  - https://www.uio.no/studier/emner/matnat/ifi/IN1010/v21/undervisningstilbud/
  - Jeg har konkrete spørsmål/problemer med min kode -> Labtime!
  - Jeg vil ha mer liveprogrammering -> Plenumstime!
  - Jeg vil jobbe med andre (og kanskje en kjapp recap av forelesning) -> Gruppetime!
  - Jeg vil ha en recap av de vanskeligste konseptene fra forelesning -> Repetisjonsgruppe!
- Skriv alt dere lurer på i chatten enten til everybody eller bare til meg 😊
  - Si i fra hvis dere faller av! Enten hvis noe er vanskelig, eller hvis dere glemte å følge med! Bare si i fra!
  - Vi har god tid, så det er masse tid til å gå gjennom ting flere ganger
- Start på oblig6 ASAP!
  - Viktig å skjønne logikken

Oblig 5

Jeg vet det var skikkelig vanskelig.

Dere har bare fått en liten innføring i tråder, og tråder er et vanskelig tema. Det er ikke forventet at dere skal bli pro på tråder etter in1010, det finnes et eget fag for det på ifi: IN3030 – Effektiv parallellprogrammering.

Dere kan bli kjempegode utviklere uten å kunne en dritt om tråder!

Før eksamen anbefaler jeg å lese kapittelet om tråder i boka.

Hjelp, jeg får det ikke til

Ikke vær too hard on yourself ❤️

Dere har litt tid før eksamen (etter vi er ferdig med alt pensum) på å repetere stoffet.

Veldig mange tar opp in1010! Programmering blir lettere jo mer man gjør det og mange trenger bare litt ekstra tid! Men gjør uansett deres beste dette semesteret.

Det at dere fortsatt henger litt med er så sinnsykt bra jobba, dere burde være skikkelig stolte.

Repetisjon denne uka

Hva er rekursjon?

Vi bryter opp komplekse oppgaver i mindre/enklere oppgaver

Rekursjon: samme operasjon flere ganger

Hvordan implementere rekursjon?

1. Basis case
    a. Vi må passe på at vi ikke får uendelig rekursjon!
    b. Basis caset gjør ikke et rekursivt kall!
2. Hvert rekursive kall må gjøre beregningen litt enklere
    a. Slik at vi til slutt treffer basis caset!

TIPS: Hvis du synes rekursjon er litt vanskelig så LES BOKA KAP 13, det gjelder å få litt feelingen for rekursjon. Men som regel er det mye enklere enn man kanskje tror! Et annet tips er å tegne hva vi vil at skal skje!

Løkke vs rekursjon

```
15  class EnkelIterasjon{
16      public static void main(String[] args) {
17          System.out.println("Iterasjon:");
18          skrivTallIterasjon(5);
19      }
20      public static void skrivTallIterasjon(int n){
21          for(int i = n; i >= 0; i--){
22              System.out.println(i);
23          }
24      }
```

OBS

Hva printes her?
Send Johanna en direktemelding i chatten.

Løkke vs rekursjon

```java
class EnkelIterasjon{
    public static void main(String[] args) {
        System.out.println("Iterasjon:");
        skrivTallIterasjon(5);
    }
    public static void skrivTallIterasjon(int n){
        for(int i = n; i >= 0; i--){
            System.out.println(i);
        }
    }
```

```
jonbon@jons-macbook-pro uke12 % java EnkelIterasjon
Iterasjon:
5
4
3
2
1
0
```

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

1. Basis case
    a. Vi må passe på at vi ikke får uendelig rekursjon!
    b. Basis caset gjør ikke et rekursivt kall!
2. Hvert rekursive kall må gjøre beregningen litt enklere
    a. Slik at vi til slutt treffer basis caset!

Løkke vs rekursjon

```java
class EnkelIterasjon{
    public static void main(String[] args) {
        System.out.println("Iterasjon:");
        skrivTallIterasjon(5);
    }
    public static void skrivTallIterasjon(int n){
        for(int i = n; i >= 0; i--){
            System.out.println(i);
        }
    }
```

```
jonbon@jons-macbook-pro uke12 % java EnkelIterasjon
Iterasjon:
5
4
3
2
1
0
```

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

Basis case

1. Basis case
    a. Vi må passe på at vi ikke får uendelig rekursjon!
    b. Basis caset gjør ikke et rekursivt kall!
2. Hvert rekursive kall må gjøre beregningen litt enklere
    a. Slik at vi til slutt treffer basis caset!

Løkke vs rekursjon

```java
class EnkelIterasjon{
    public static void main(String[] args) {
        System.out.println("Iterasjon:");
        skrivTallIterasjon(5);
    }
    public static void skrivTallIterasjon(int n){
        for(int i = n; i >= 0; i--){
            System.out.println(i);
        }
    }
```

```
jonbon@jons-macbook-pro uke12 % java EnkelIterasjon
Iterasjon:
5
4
3
2
1
0
```

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

Enklere kall hver gang

1. Basis case
    a. Vi må passe på at vi ikke får uendelig rekursjon!
    b. Basis caset gjør ikke et rekursivt kall!
2. Hvert rekursive kall må gjøre beregningen litt enklere
    a. Slik at vi til slutt treffer basis caset!

Løkke vs rekursjon

```java
class EnkelIterasjon{
    public static void main(String[] args) {
        System.out.println("Iterasjon:");
        skrivTallIterasjon(5);
    }
    public static void skrivTallIterasjon(int n){
        for(int i = n; i >= 0; i--){
            System.out.println(i);
        }
    }
```

```
jonbon@jons-macbook-pro uke12 % java EnkelIterasjon
Iterasjon:
5
4
3
2
1
0
```

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

Hva printes her?
Send Johanna en direktemelding i chatten.

Løkke vs rekursjon

```java
class EnkelIterasjon{
  public static void main(String[] args) {
    System.out.println("Iterasjon:");
    skrivTallIterasjon(5);
  }
  public static void skrivTallIterasjon(int n){
    for(int i = n; i >= 0; i--){
      System.out.println(i);
    }
  }
```

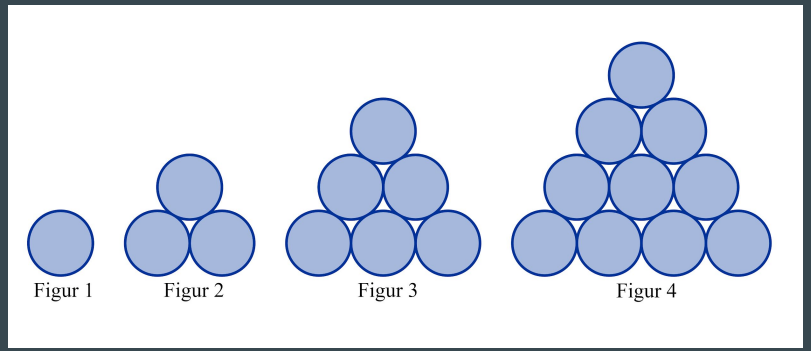
```
jonbon@jons-macbook-pro uke12 % java EnkelIterasjon
Iterasjon:
5
4
3
2
1
0
```

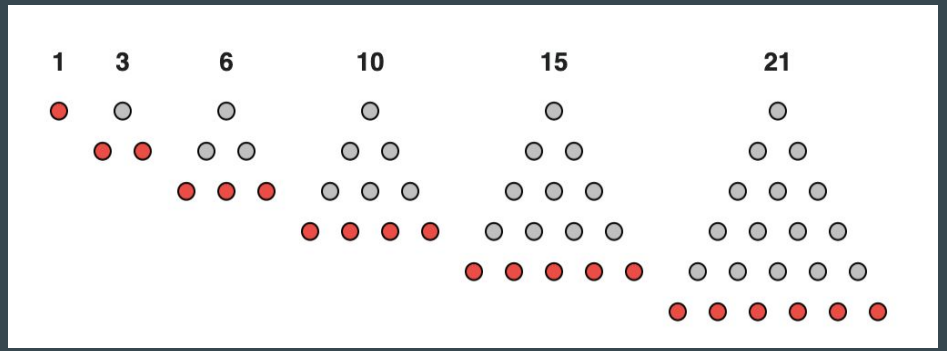
```java
class EnkelRekursjon {
  public static void main(String[] args) {
    System.out.println("Rekursjon:");
    skrivTallRekursjon(5);
  }
  public static void skrivTallRekursjon(int n){
    if (n < 0) {
      return;
    }
    System.out.println(n);
    skrivTallRekursjon(n-1);
```

```
jonbon@jons-macbook-pro uke12 % java EnkelRekursjon
Rekursjon:
5
4
3
2
1
0
```

Eksempel: rekursjon trekanttall

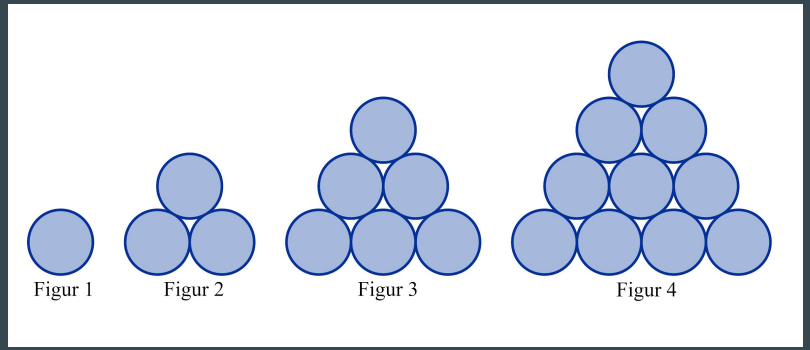
Hvor mange sirkler består trekanten av totalt, dersom hver kant er x sirkler?
Antall sirkler totalt kalles trekanttallet til trekanten.

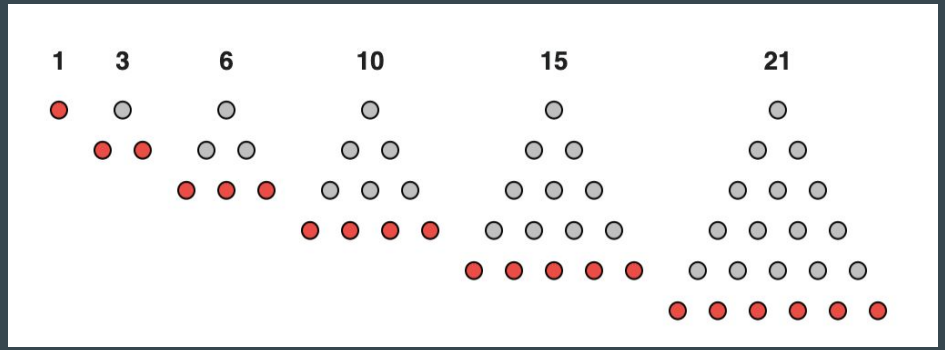




Eksempel: rekursjon trekanttall

Hvor mange sirkler består trekanten av totalt, dersom hver kant er x sirkler?
Antall sirkler totalt kalles trekanttallet til trekanten.





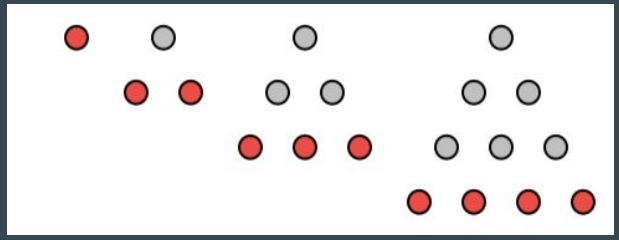
1 1+2=3 3+3=6 6+4=10

Det er veldig enkelt å regne ut dersom vi vet trekanttallet til den forrige trekanten!

Eksempel: rekursjon trekanttall

For å regne ut trekanttallet til trekant 2 trenger vi bare vite trekanttallet til trekant 1 og så må vi plusse på 2.

(trekanttallet til trekant 1) + 2 = x
1 + 2 = 3

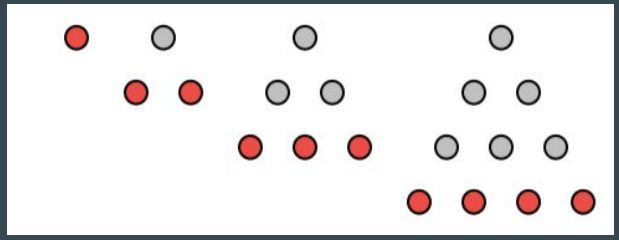


1 1+2=3 3+3=6 6+4=10

Eksempel: rekursjon trekanttall

For å regne ut trekanttallet til trekant 3 trenger vi bare vite trekanttallet til trekant 2.

(trekanttallet til trekant 2) + 3 = x
3 + 3 = 6

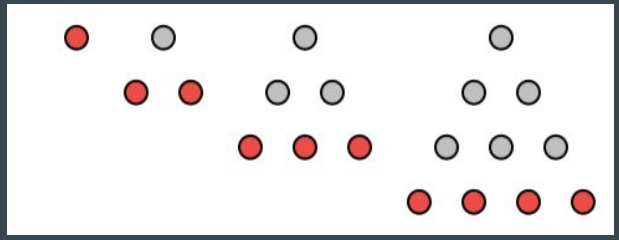


1 1+2=3 3+3=6 6+4=10

Eksempel: rekursjon trekanttall

For å regne ut trekanttallet til trekant 4 trenger vi bare vite trekanttallet til trekant 3.

(trekanttallet til trekant 3) + 4 = x
6 + 4 = 10



1 1+2=3 3+3=6 6+4=10

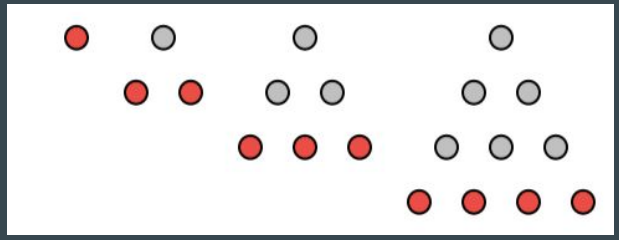
Eksempel: rekursjon trekanttall

For å regne ut trekanttallet til trekant y trenger vi bare vite trekanttallet til trekanten før y.

Siden vi vet at trekanttallet til trekant 1 = 1 så kan vi regne oss fram til hvilket som helst trekanttall.

Send Johanna melding i chatten:

- Hva er trekanttallet til trekant 6?
- Hvordan kan vi regne oss fram til det ved å bare vite trekanttallet til trekant 1?

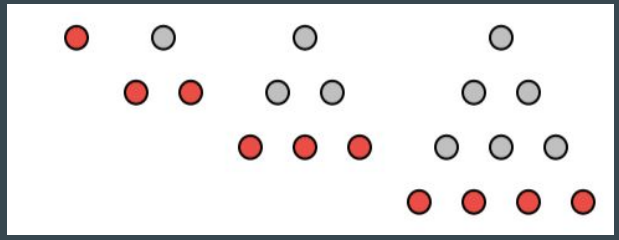


1 1+2=3 3+3=6 6+4=10

Eksempel: rekursjon trekanttall

Finne trekanttallet til trekant 6:

- For å regne ut trekanttallet til trekant 6 trenger vi bare vite trekanttallet til trekant 5!

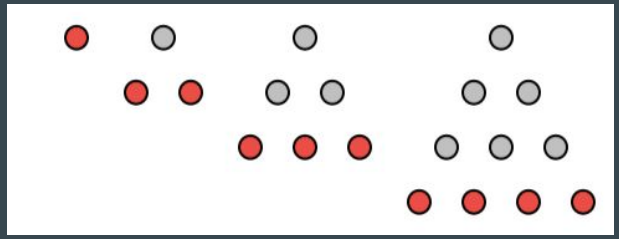


1 1+2=3 3+3=6 6+4=10

Eksempel: rekursjon trekanttall

Finne trekanttallet til trekant 6:

- For å regne ut trekanttallet til trekant 6 trenger vi bare vite trekanttallet til trekant 5!
- For å regne ut trekanttallet til trekant 5 trenger vi bare vite trekanttallet til trekant 4!

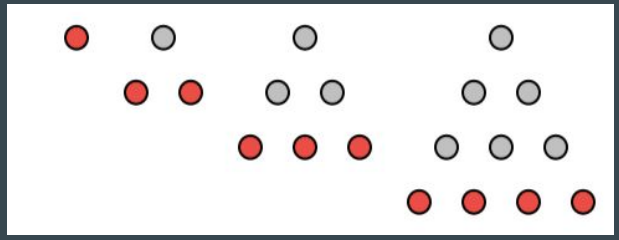


1 1+2=3 3+3=6 6+4=10

Eksempel: rekursjon trekanttall

Finne trekanttallet til trekant 6:

- For å regne ut trekanttallet til trekant 6 trenger vi bare vite trekanttallet til trekant 5!
- For å regne ut trekanttallet til trekant 5 trenger vi bare vite trekanttallet til trekant 4!
- For å regne ut trekanttallet til trekant 4 trenger vi bare vite trekanttallet til trekant 3!

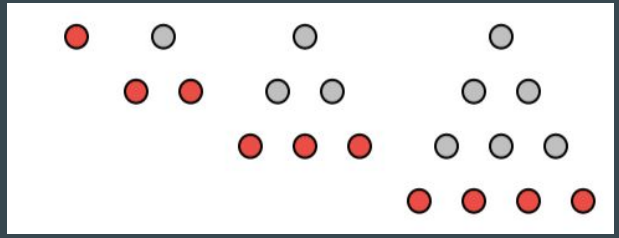


1 1+2=3 3+3=6 6+4=10

Eksempel: rekursjon trekanttall

Finne trekanttallet til trekant 6:

- For å regne ut trekanttallet til trekant 6 trenger vi bare vite trekanttallet til trekant 5!
- For å regne ut trekanttallet til trekant 5 trenger vi bare vite trekanttallet til trekant 4!
- For å regne ut trekanttallet til trekant 4 trenger vi bare vite trekanttallet til trekant 3!
- For å regne ut trekanttallet til trekant 3 trenger vi bare vite trekanttallet til trekant 2!

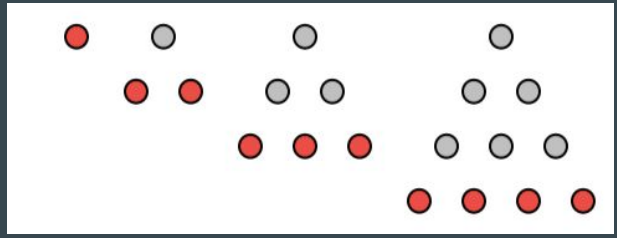


1 1+2=3 3+3=6 6+4=10

Eksempel: rekursjon trekanttall

Finne trekanttallet til trekant 6:

- For å regne ut trekanttallet til trekant 6 trenger vi bare vite trekanttallet til trekant 5!
- For å regne ut trekanttallet til trekant 5 trenger vi bare vite trekanttallet til trekant 4!
- For å regne ut trekanttallet til trekant 4 trenger vi bare vite trekanttallet til trekant 3!
- For å regne ut trekanttallet til trekant 3 trenger vi bare vite trekanttallet til trekant 2!
- For å regne ut trekanttallet til trekant 2 trenger vi bare vite trekanttallet til trekant 1!

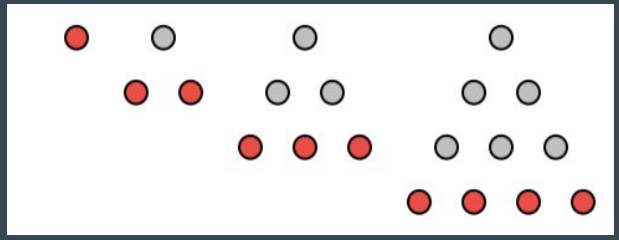


1 1+2=3 3+3=6 6+4=10

Eksempel: rekursjon trekanttall

Finne trekanttallet til trekant 6:

- For å regne ut trekanttallet til trekant 6 trenger vi bare vite trekanttallet til trekant 5!
- For å regne ut trekanttallet til trekant 5 trenger vi bare vite trekanttallet til trekant 4!
- For å regne ut trekanttallet til trekant 4 trenger vi bare vite trekanttallet til trekant 3!
- For å regne ut trekanttallet til trekant 3 trenger vi bare vite trekanttallet til trekant 2!
- For å regne ut trekanttallet til trekant 2 trenger vi bare vite trekanttallet til trekant 1!
- Trekanttallet til trekant 1 vet vi jo! Det er 1!

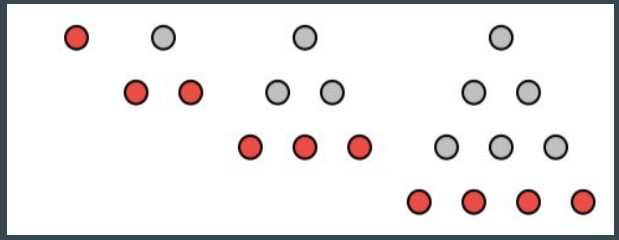


1 1+2=3 3+3=6 6+4=10

Eksempel: rekursjon trekanttall

Finne trekanttallet til trekant 6:

- For å regne ut trekanttallet til trekant 6 trenger vi bare vite trekanttallet til trekant 5!
- For å regne ut trekanttallet til trekant 5 trenger vi bare vite trekanttallet til trekant 4!
- For å regne ut trekanttallet til trekant 4 trenger vi bare vite trekanttallet til trekant 3!
- For å regne ut trekanttallet til trekant 3 trenger vi bare vite trekanttallet til trekant 2!
- For å regne ut trekanttallet til trekant 2 trenger vi bare vite trekanttallet til trekant 1! 1 + 2 = 3!
- Trekanttallet til trekant 1 vet vi jo! Det er 1!

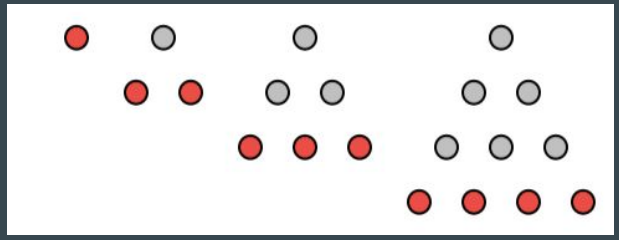


1 1+2=3 3+3=6 6+4=10

Eksempel: rekursjon trekanttall

Finne trekanttallet til trekant 6:

- For å regne ut trekanttallet til trekant 6 trenger vi bare vite trekanttallet til trekant 5!
- For å regne ut trekanttallet til trekant 5 trenger vi bare vite trekanttallet til trekant 4!
- For å regne ut trekanttallet til trekant 4 trenger vi bare vite trekanttallet til trekant 3!
- For å regne ut trekanttallet til trekant 3 trenger vi bare vite trekanttallet til trekant 2! 3 + 3 = 6!
- For å regne ut trekanttallet til trekant 2 trenger vi bare vite trekanttallet til trekant 1! 1 + 2 = 3!
- Trekanttallet til trekant 1 vet vi jo! Det er 1!

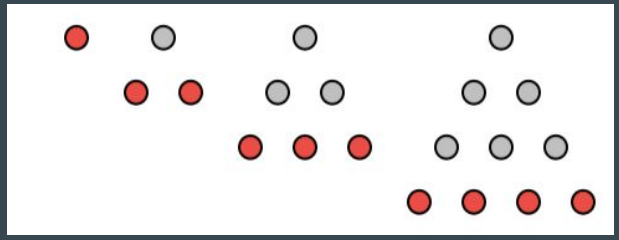


1 1+2=3 3+3=6 6+4=10

Eksempel: rekursjon trekanttall

Finne trekanttallet til trekant 6:

- For å regne ut trekanttallet til trekant 6 trenger vi bare vite trekanttallet til trekant 5!
- For å regne ut trekanttallet til trekant 5 trenger vi bare vite trekanttallet til trekant 4!
- For å regne ut trekanttallet til trekant 4 trenger vi bare vite trekanttallet til trekant 3! 6 + 4 = 10!
- For å regne ut trekanttallet til trekant 3 trenger vi bare vite trekanttallet til trekant 2! 3 + 3 = 6!
- For å regne ut trekanttallet til trekant 2 trenger vi bare vite trekanttallet til trekant 1! 1 + 2 = 3!
- Trekanttallet til trekant 1 vet vi jo! Det er 1!

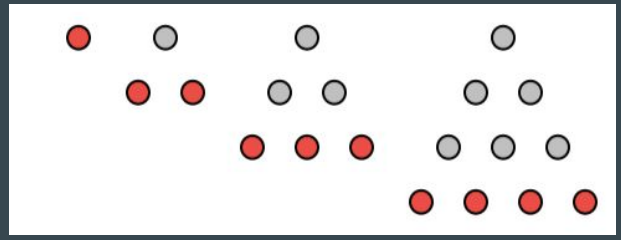


1 1+2=3 3+3=6 6+4=10

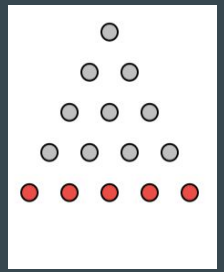
Eksempel: rekursjon trekanttall

Finne trekanttallet til trekant 6:

- For å regne ut trekanttallet til trekant 6 trenger vi bare vite trekanttallet til trekant 5!
- For å regne ut trekanttallet til trekant 5 trenger vi bare vite trekanttallet til trekant 4! 10 + 5 = 15!
- For å regne ut trekanttallet til trekant 4 trenger vi bare vite trekanttallet til trekant 3! 6 + 4 = 10!
- For å regne ut trekanttallet til trekant 3 trenger vi bare vite trekanttallet til trekant 2! 3 + 3 = 6!
- For å regne ut trekanttallet til trekant 2 trenger vi bare vite trekanttallet til trekant 1! 1 + 2 = 3!
- Trekanttallet til trekant 1 vet vi jo! Det er 1!



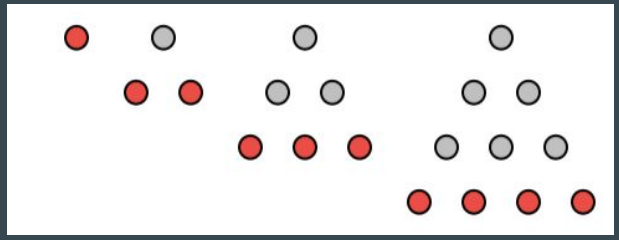
1 1+2=3 3+3=6 6+4=10



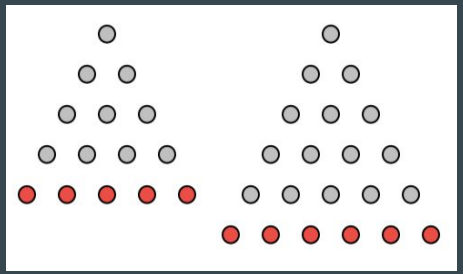
Eksempel: rekursjon trekanttall

Finne trekanttallet til trekant 6:

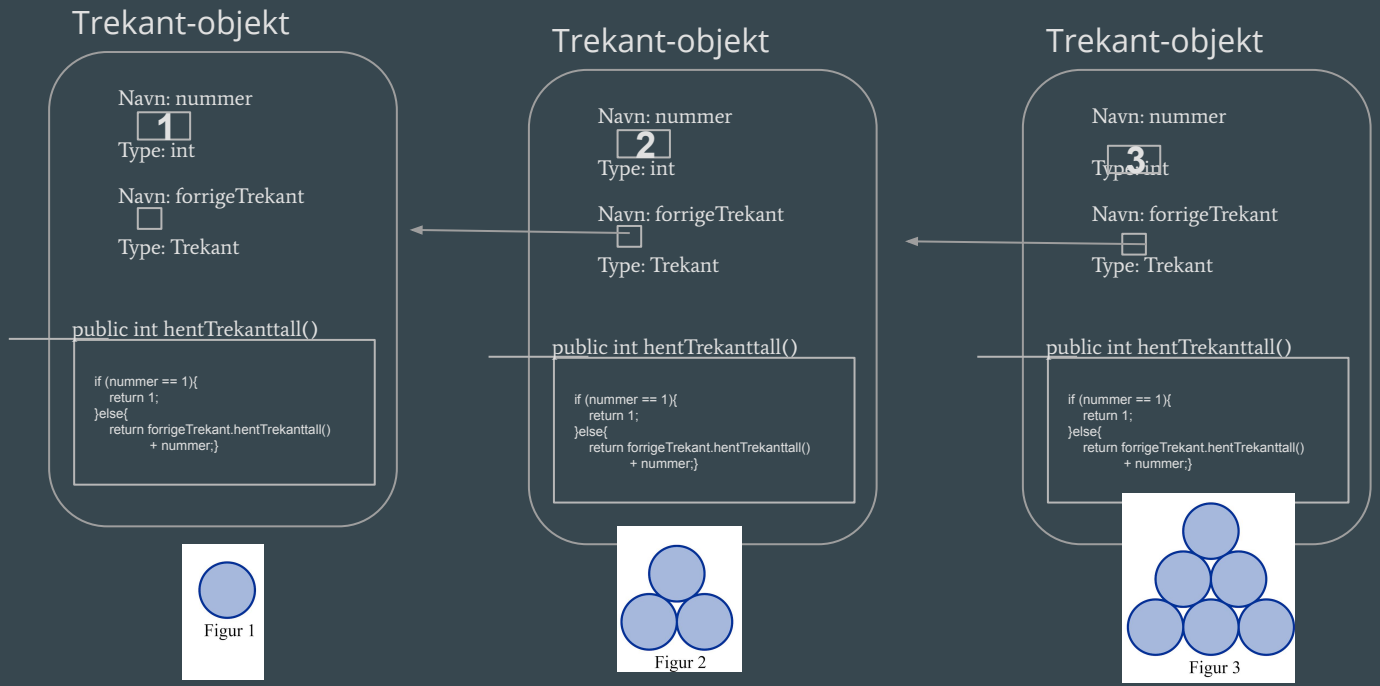
- For å regne ut trekanttallet til trekant 6 trenger vi bare vite trekanttallet til trekant 5! 15 + 6 = 21
- For å regne ut trekanttallet til trekant 5 trenger vi bare vite trekanttallet til trekant 4! 10 + 5 = 15!
- For å regne ut trekanttallet til trekant 4 trenger vi bare vite trekanttallet til trekant 3! 6 + 4 = 10!
- For å regne ut trekanttallet til trekant 3 trenger vi bare vite trekanttallet til trekant 2! 3 + 3 = 6!
- For å regne ut trekanttallet til trekant 2 trenger vi bare vite trekanttallet til trekant 1! 1 + 2 = 3!
- Trekanttallet til trekant 1 vet vi jo! Det er 1!



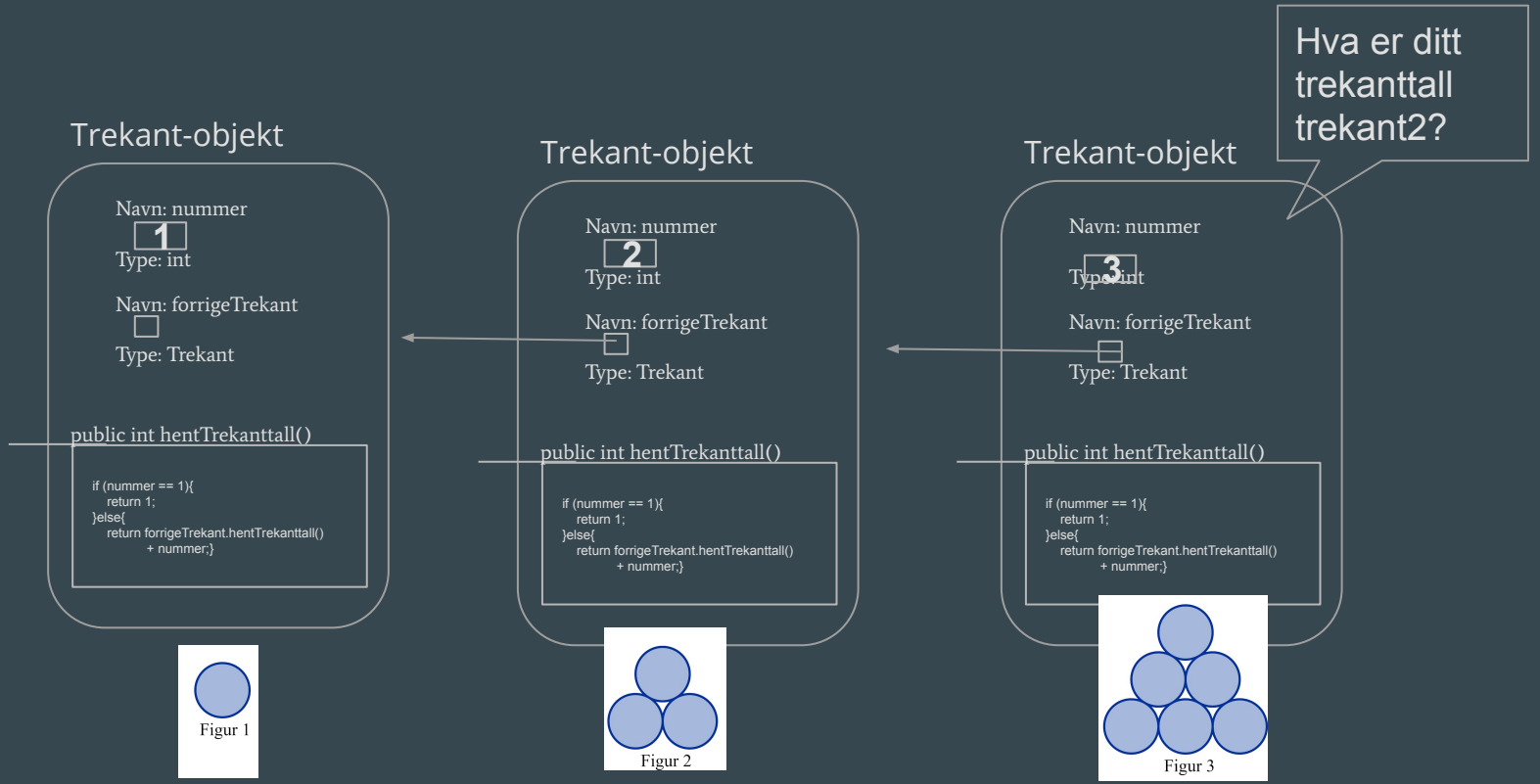
1 1+2=3 3+3=6 6+4=10



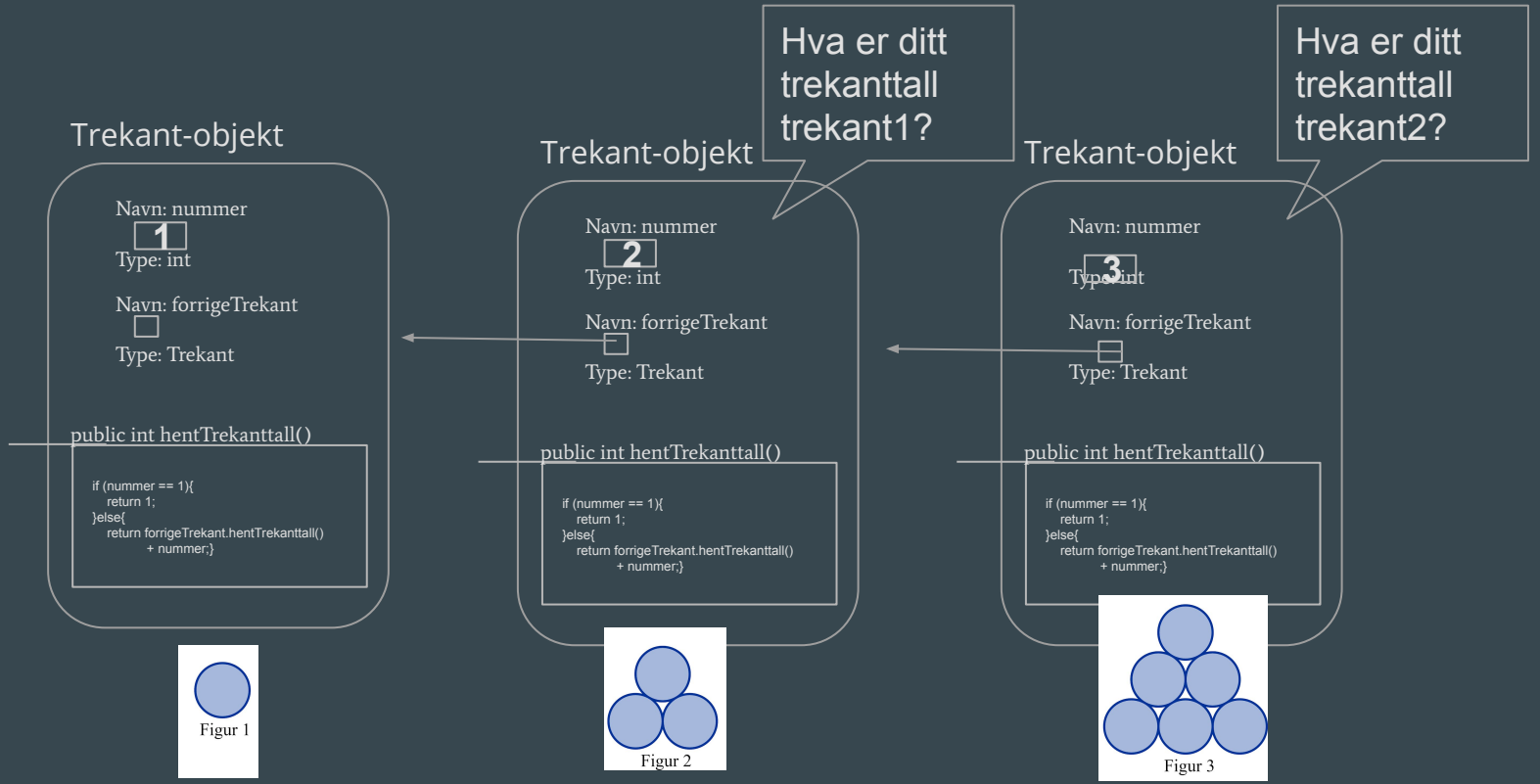
Rekursjon trekanttall



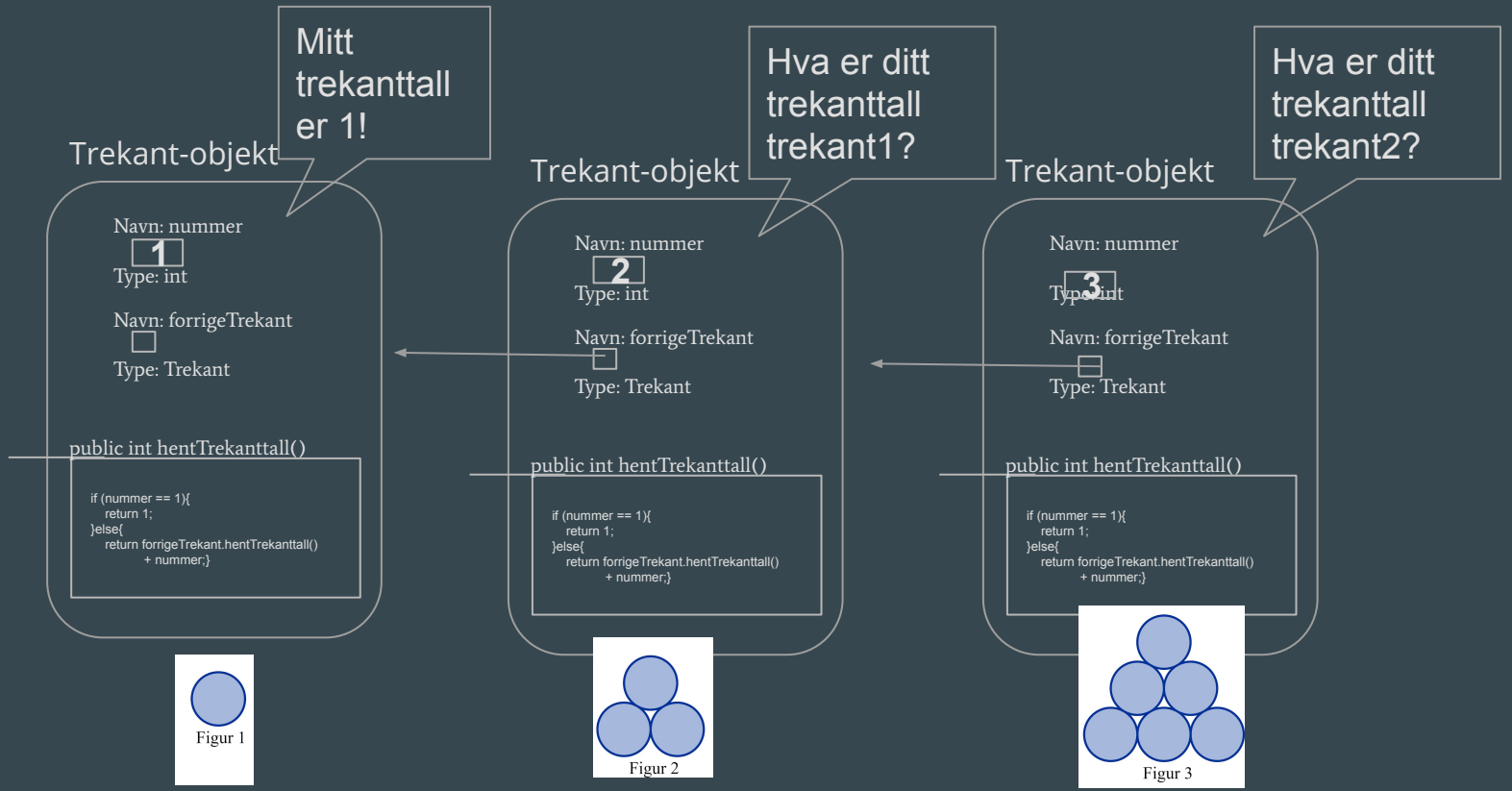
Rekursjon trekanttall



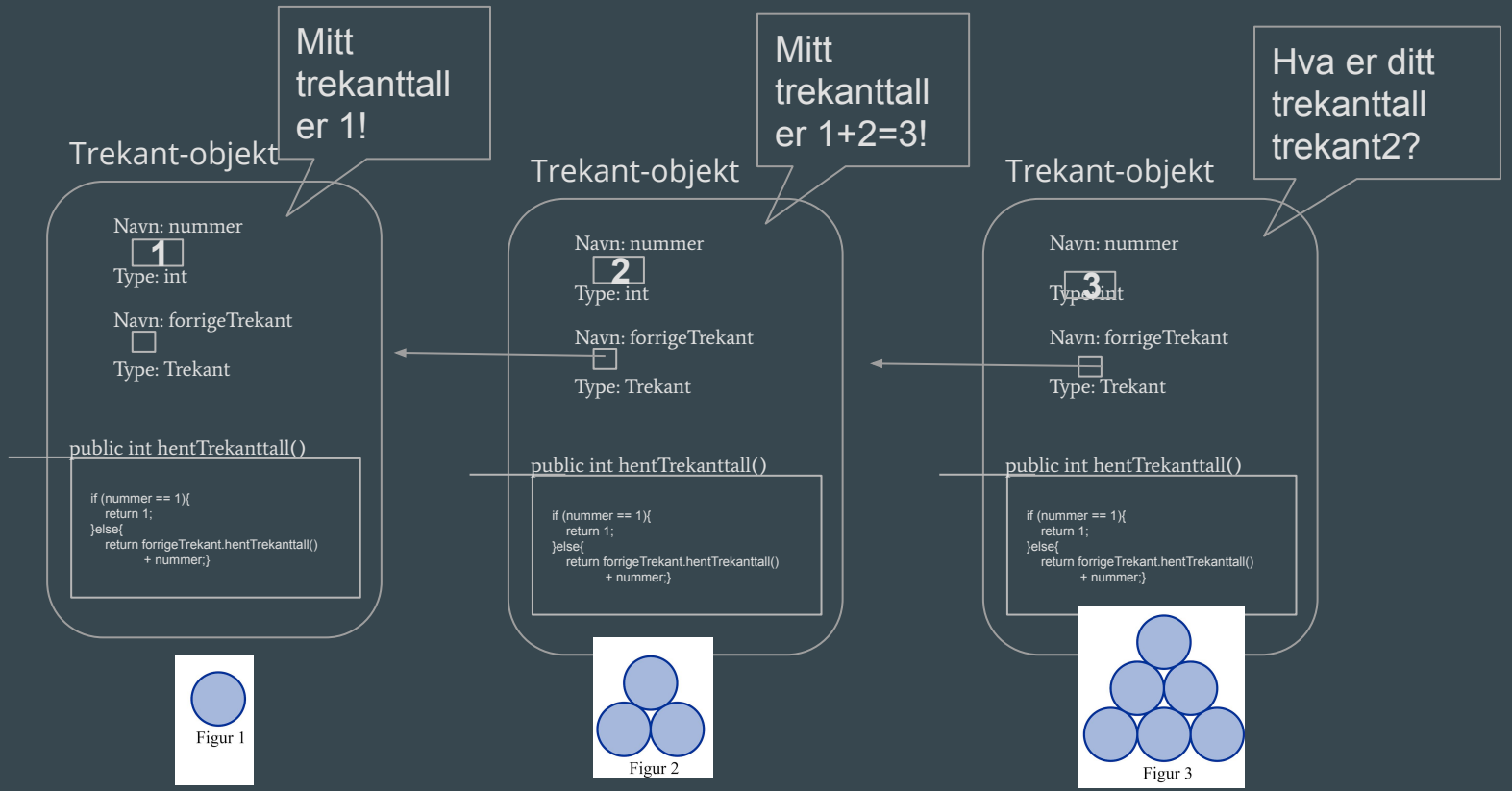
Rekursjon trekanttall



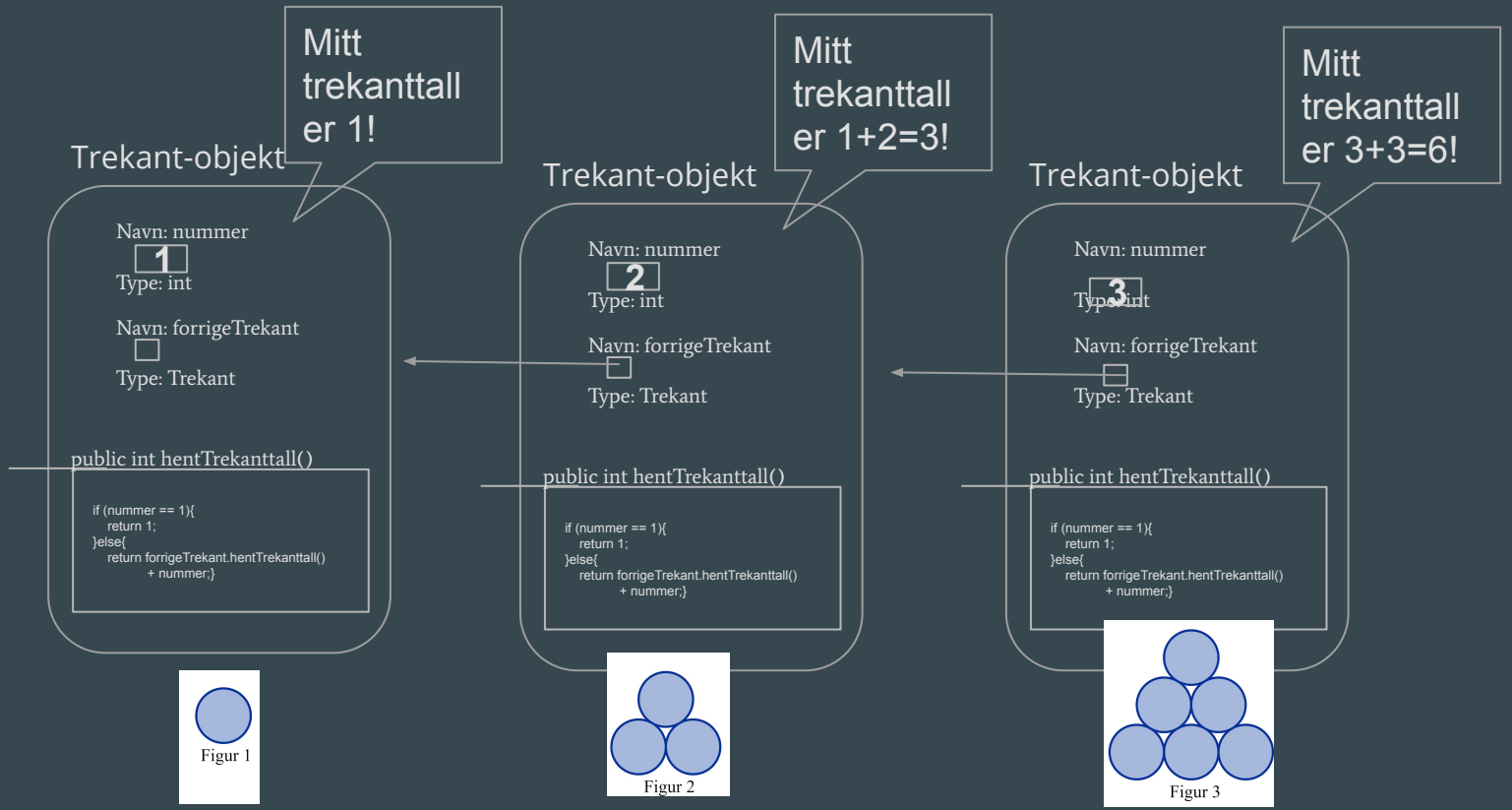
Rekursjon trekanttall



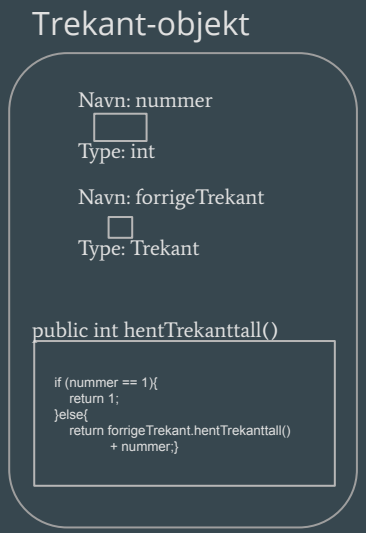
Rekursjon trekanttall



Rekursjon trekanttall



Rekursjon trekanttall

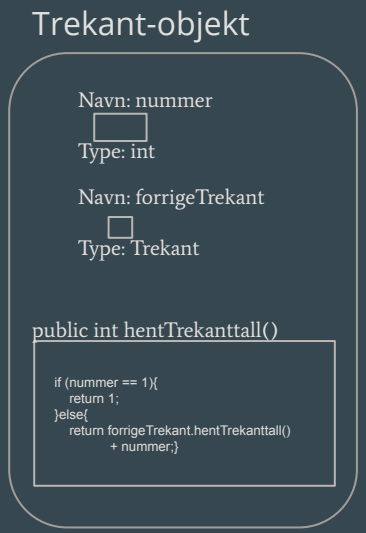


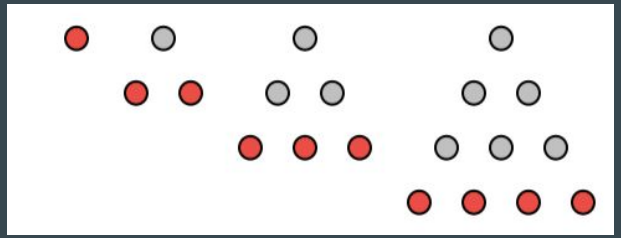

```java
class Trekant{
    private int nummer;
    private Trekant forrigeTrekant;

    public Trekant(int nummer, Trekant forrigeTrekant){
        this.nummer = nummer;
        this.forrigeTrekant = forrigeTrekant;
    }
    public int hentTrekanttall(){
        if (nummer == 1){
            return 1;
        }else{
            return forrigeTrekant.hentTrekanttall() + nummer;
        }
```

Rekursjon trekanttall

```java
class Trekant{
    private int nummer;
    private Trekant forrigeTrekant;

    public Trekant(int nummer, Trekant forrigeTrekant){
        this.nummer = nummer;
        this.forrigeTrekant = forrigeTrekant;
    }
    public int hentTrekanttall(){
        if (nummer == 1){
            return 1;
        }else{
            return forrigeTrekant.hentTrekanttall() + nummer;
        }
```

Basis case

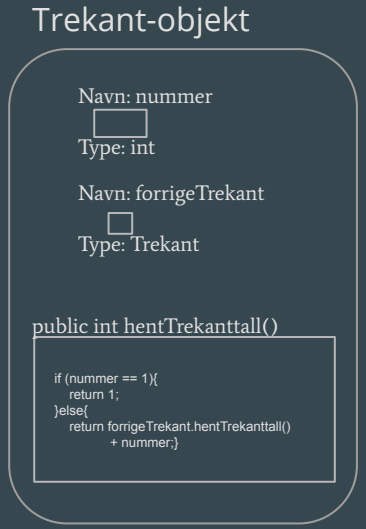


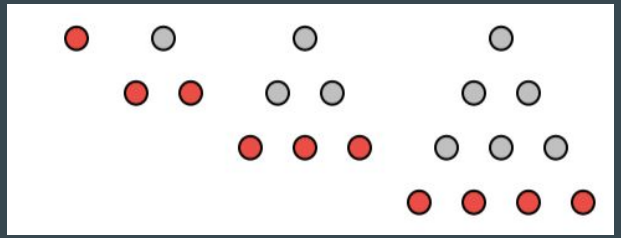


Rekursjon trekanttall

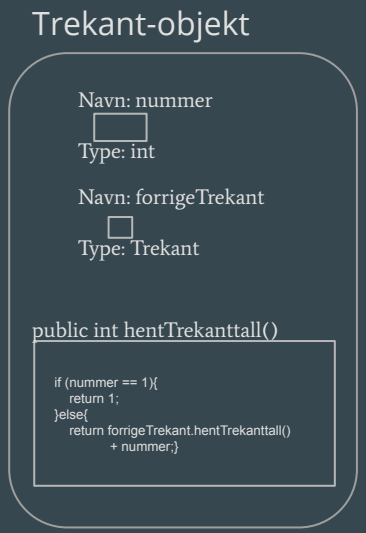
```java
class Trekant{
    private int nummer;
    private Trekant forrigeTrekant;

    public Trekant(int nummer, Trekant forrigeTrekant){
        this.nummer = nummer;
        this.forrigeTrekant = forrigeTrekant;
    }
    public int hentTrekanttall(){
        if (nummer == 1){
            return 1;
        }else{
            return forrigeTrekant.hentTrekanttall() + nummer;
        }
```

Enklere kall hver gang





Rekursjon trekanttall




```java
class Trekant{
    private int nummer;
    private Trekant forrigeTrekant;

    public Trekant(int nummer, Trekant forrigeTrekant){
        this.nummer = nummer;
        this.forrigeTrekant = forrigeTrekant;
    }
    public int hentTrekanttall(){
        if (nummer == 1){
            return 1;
        }else{
            return forrigeTrekant.hentTrekanttall() + nummer;
        }
```

```java
class TestTrekant{
    public static void main(String[] args) {
        Trekant trekant1 = new Trekant(1, null);
        Trekant trekant2 = new Trekant(2, trekant1);
        Trekant trekant3 = new Trekant(3, trekant2);
        Trekant trekant4 = new Trekant(4, trekant3);
        Trekant trekant5 = new Trekant(5, trekant4);
        Trekant trekant6 = new Trekant(6, trekant5);

        System.out.println(trekant6.hentTrekanttall());
    }
```

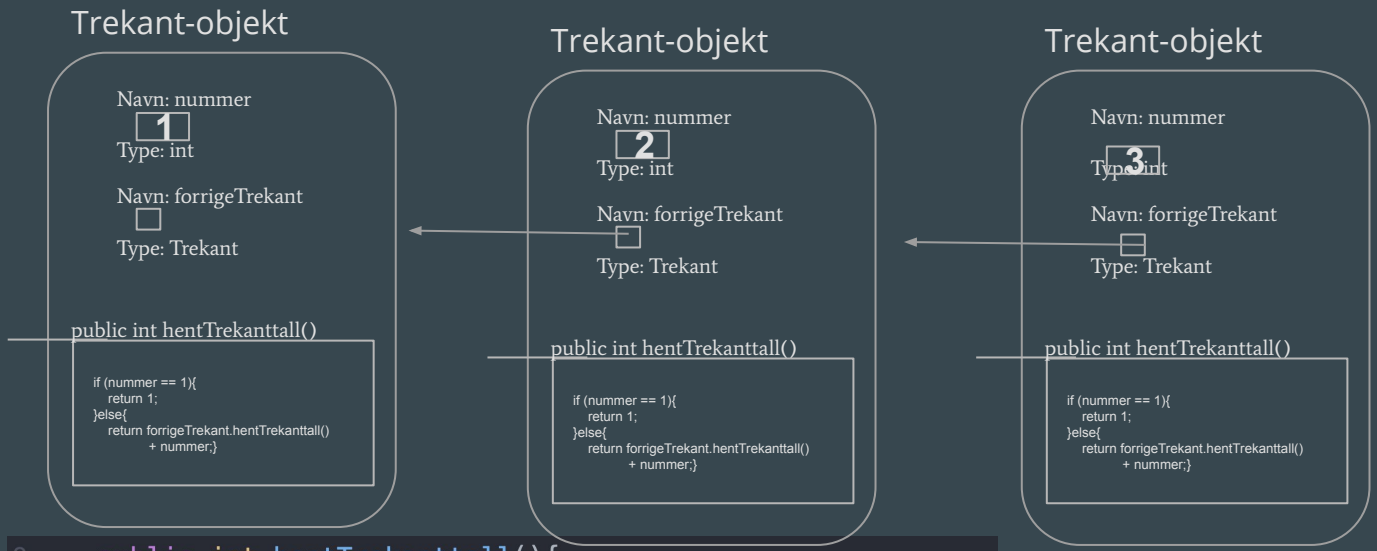
Kjøre koden

```
jonbon@jons-macbook-pro uke12 % java TestTrekant
21
```

```java
class Trekant{
    private int nummer;
    private Trekant forrigeTrekant;

    public Trekant(int nummer, Trekant forrigeTrekant){
        this.nummer = nummer;
        this.forrigeTrekant = forrigeTrekant;
    }
    public int hentTrekanttall(){
        if (nummer == 1){
            return 1;
        }else{
            return forrigeTrekant.hentTrekanttall() + nummer;
        }
```

```java
class TestTrekant{
    public static void main(String[] args) {
        Trekant trekant1 = new Trekant(1, null);
        Trekant trekant2 = new Trekant(2, trekant1);
        Trekant trekant3 = new Trekant(3, trekant2);
        Trekant trekant4 = new Trekant(4, trekant3);
        Trekant trekant5 = new Trekant(5, trekant4);
        Trekant trekant6 = new Trekant(6, trekant5);

        System.out.println(trekant6.hentTrekanttall());
    }
```

Rekursjon trekanttall




```
9   public int hentTrekanttall(){
10      if (nummer == 1){
11          return 1;
12      }else{
13          return forrigeTrekant.hentTrekanttall() + nummer;
14      }
15  }
```

Før og etter rekursivt kall

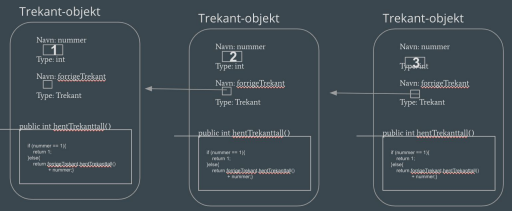
Viktig å henge med på hva som skjer FØR vs. hva som skjer ETTER et rekursivt kall!

Vi skal legge til noen print-lines i koden og printe før og etter de rekursive kallene!

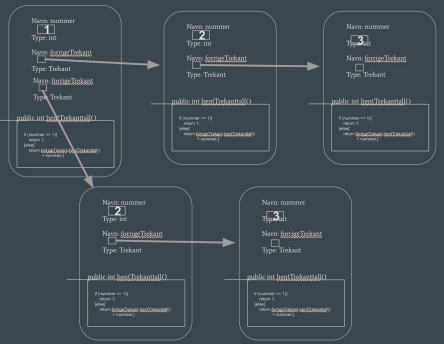
```
jonbon@jons-macbook-pro uke12 % java TestTrekant
Henter trekanttallet til trekant 6:
---------------------------------
Trekant 6 sier: Hva er ditt trekanttall, trekant 5?
Trekant 5 sier: Hva er ditt trekanttall, trekant 4?
Trekant 4 sier: Hva er ditt trekanttall, trekant 3?
Trekant 3 sier: Hva er ditt trekanttall, trekant 2?
Trekant 2 sier: Hva er ditt trekanttall, trekant 1?
Trekant 1 sier: Mitt trekanttall er 1!
Trekant 2 sier: Da er mitt trekanttall: 1+2=3!
Trekant 3 sier: Da er mitt trekanttall: 3+3=6!
Trekant 4 sier: Da er mitt trekanttall: 6+4=10!
Trekant 5 sier: Da er mitt trekanttall: 10+5=15!
Trekant 6 sier: Da er mitt trekanttall: 15+6=21!
---------------------------------
Trekanttallet til trekant 6 er: 21
```

Rekursjon kan grene utover

Vi så på dette eksempelet der strukturen ligner veldig på en lenkeliste og det rekursive kallet går i én retning:



I obligen skal dere implementere et rekursivt kall i noe som egentlig er en graf eller et tre, og det rekursive kallet går i flere retninger, det kan være flere rekursive kall i samme metode:



Jobbe selv

Jobb med hva dere vil og rekk opp hånda hvis dere trenger hjelp med noe/har spørsmål så møtes vi i breakoutroom! 😊