Spørsmål?

- Skriv de i chat!
  - Henrik svarer om han kan (i chat)
  - Stilles videre om det er behov
- Still det muntlig!
  - Rekk opp hånden / Skriv at du vi vil snakke
- Bruk reaksjoner + «Slower» / «Faster» for å vise hvordan dere henger med
- Kode blir lagt ut etter timen

Bytt navn hvis dere vil være anonyme

Uke 12

Design og Testing

Plan

- Design
- Rekursjon
- Testing

Hva er et godt design? (fra IN1030):

- En god utforming gjør den jobben den er ment å gjøre.
- En god utforming er enkel og elegant.
    - Eleganse innebærer å finne akkurat riktig abstraksjonsnivå.
- En god utforming er gjenbrukbar, utvidbar og enkel å forstå.
- Et godt objekt har et lite og veldefinert ansvarsområde.
- Et godt objekt skjuler implementasjonsdetaljer fra andre objekter

Rekursjon

https://www.uio.no/studier/emner/matnat/ifi/IN1010/v20/grupper/gruppe-13/uke-8/rekursjon.pdf

Rekursjon enkelt eksempel

Overloading? - Yes/No

Samme metode navn, forskjellige parametere

Skriv ut tall (nedover) fra n til 1

5, 4, 3, 2, 1

Warning: flere måter å gjøre dette på

Testing

- Vil ikke ha feil - Exceptions, Errors, Logiske feil
- Unit Tests + Integration tests
- Gjøre programmeringen raskere
  - Vi skal skrive testene før vi skriver ‘’main’’ koden vår

Testing

Ha som mål å knekke programmet,

- Hva kan gå galt?
- Hva må være riktig?

Det kan være lurt å skrive testene før selve programmet, da dette gjør at du vil måtte tenke på eventuelle utfordringer før man begynner med kodingen - lite gjennomtenkt kode fører fort til spagettikode.

```
static void sjekk(Object forventet, Object faktisk, String testmelding) {
        if (forventet.equals(faktisk)) {
                sjekkPasserte();
        }
        else {
                sjekkFeilet(testmelding);
                System.out.println("  > Forventet verdi: "+forventet);
                System.out.println("  > Faktisk verdi: " + faktisk);
          }
    }
```

```
static void sjekkPasserte() {
        antallTester++;
        antallPasserte++;
        System.out.println("- Test " + antallTester + ": OK");
}
static void sjekkFeilet(String testmelding) {
        antallTester++;
        antallFeil++;
        System.out.println("- Test " + antallTester + " feilet: " + testmelding);
}
```

```
try {
      liste.hent(0);
      sjekkFeilet("hent(0) paa tom liste skulle kastet
 unntak");
}
catch(UgyldigListeIndeks e) {
      sjekkPasserte();
}
```

Live-Koding

Fokus på Testing og Rekursjon

Oppgave 1

- Skriv testen til den rekursive metoden leggSammen2DArray
- Skriv den rekursive metoden leggSammen2DArray. (PS: Husk at det er lov med hjelpemetoder)
- Bruk testen du skrev i deloppgave 1.A, for å teste 1.B

Oppgave 1

```
class RekursiveMetoder{
  /**
  *Retuerner summen av alle tallene i int 2D array.
  *Dette gjøres rekursivt
  *@param array: int[][] tallen som skal summeres
  *@return summen av alle tallen i array
  */
  public static int leggsammen2DArray(int[][] array){
  }
}
```

Live-Koding

Oppgave 2

- Skriv testen til de rekursive metoden hentStringRiktigVei og hentStringFeilVei
- Skriv de rekursive metoden hentStringRiktigVei og hentStringFeilVei.
- Bruk testene du skrev i deloppgave 2.A, for å teste 2.B

Oppgave 2

```
class RekursiveMetoder{
  /**
  *Legger til alle stringen i en liste til en liste i riktig rekkefølge
  *@param ord: alle stringen som skal konkantineres
  *@return en string av alle ordene i lista i riktig rekkefølge
  */


  public static String hentStringRiktigVei(List<String> ord){
  }
  /**
  *Legger til alle stringen i en liste til en liste i feil rekkefølge
  *@param ord: alle stringen som skal konkantineres
  *@return en string av alle ordene i lista i feil rekkefølge
  */
  public static String hentStringFeilVei(List<String> ord){
  }
}
```

Live-Koding

Oppgave 3

Konsept: finnTekst er en metode som rekursivt finner en setning i et 2D array av stringer. Det finnes noen regler som denne metoden bruker:

- En setning kan kun gå til høyre eller nedover i et array
- Hvis det finnes String til høyre for nåværende posisjon finnes det ikke en String nedenfor
- Hvis det finnes en String nedenfor for nåværende posisjon finnes det ikke en String til høyre
- En tekst streng starter alltid med indeks [0][0]

Dette	null	null
er	en	null
null	sti	!

Her	er	null	null
null	en	tekst	null
null	null	tenk!	nul
null	null	null	null

Oppgaver:

- Skriv testen til den rekursive metoden finnTekst
- Skriv den rekursive metoden finnTekst.
- Bruk testen du skrev i deloppgave 3.A, for å teste 3.B

```
class RekursiveMetoder{
  /**
  *Leter nedover og til høyere i arrayet for å finne en tekst . Dette løses
rekursivt
  *@param ord:  arrayet som innholder alle strengene
  *@param indeks1: bestemmer om man skal "gå opp eller ned" i arrayet
  *@param indeks2: bestemmer om man skal "gå til høyere eller venstere" i arrayet
  *@return en tekst.
  */
  public static String finnTekst(String[][] ord, int indeks1, int indeks2){
  }
}
```

Live-Koding