

Lag deg en oversikt over hva følgende ord/begreper betyr:

- Refaktorere

Refaktorerere er å endre på koden litt etter litt uten at det skal påvirke så mye på selve funksjonaliteten til koden. Med små endringer og tester reduserer men risikoen for at koden feiler.

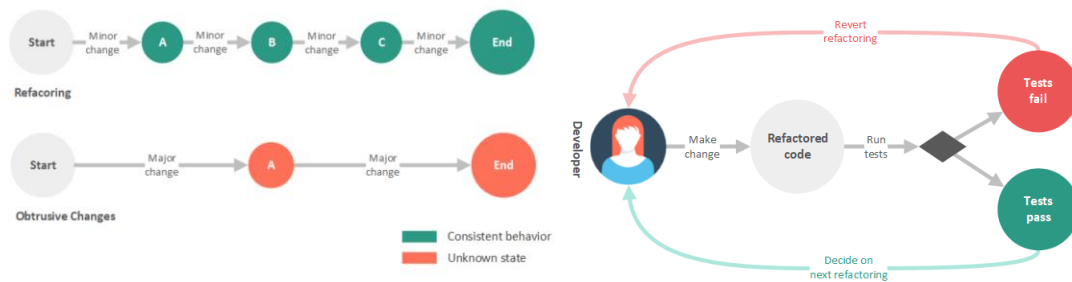
Eksempler på refaktoring:

endre/forenkle klasse navnet til noe som passer bedre.

Endre på en funksjon i en klasse til å legge til flere arver og klasser

Når man skriver en kode å ser at to klasser har mye av det samme, så kan man sette en felle klasse og la de to klassene arve fra den klassen.

Tok med to eksempel bilde som forklarer prosessen.



- Static (variabel, metode)

Når noe er satt som statisk betyr det at det tilhører til selve klassen og ikke instansen objektet er i.

Det vil si om man lager en statisk variabel, vil alle instansene av den klassen ha tilgang til den variabelen.

Lager man statisk metode kan man bruke metoden uten og initialisere et objekt av klassen.

Eksempel på statisk variabel er `Math.PI` siden dette er en final variabel i matte klassen til java.

Eksempel på statisk metode er `Math.pow()` hvor dette er en metode i matte klassen til java.

- Final (variabel, metode, klasse)

Final kan bli sett som en konstant og skrives alltid med store bokstaver.

Variabel: i en final variabel kan ikke endres etter den er gitt en verdi

Metode: en final metode kan ikke overloads(endring av metodens funksjon) av underklasser

Klasse: disse klassene kan man ikke lage under klasser til

- Abstract (klasse, metode)

Klasse: dette er klasser man ikke kan initialisere(kan ikke skrive «type variabelNavn = new type»)

Metode: dette er metoder som ikke trenger en indre funksjon, så klasser som arver fra denne må lage egen implementasjon.

- Interface

Interface ligner mye på abstract forskjellen er at et interface ikke kan ha noen funksjoner i metodene, så man kan se på dette som en blueprint av hvordan under klasser skal se ut

## Oppgave 1.2 – Sammenligning

Sammenlignet med Even Langrbråten.

getSurfaceGravity

Even hadde med mellom regninger av variabler. Før han returnerte den endelige verdien. Jeg derimot skrev hele formelen i retur statementet.

getSmallestPlanet

Even brukte en for loop for og gå igjennom alle planetene, mens han lagret det største til en midlertidig variabel. Han sjekket først om massen var forskjellige og vis de var satt han den midlertidige variabelen til den minste. Hvis massen var lik sjekket han at radiusen var forskjellig og satt den midlertidig variabelen til den minste. Etter han hadde gått igjennom alle planetene returnerte han den midertidig variabelen.

Jeg bruket en for each loop til og gå igjennom alle planetene. Jeg valgte og bare sjekke masse. Gjøre som even lagret minste i en midertidig variabel og returnerte den mot slutten.

getMassInKg

vi gjorde det samme som i surface gravity. utenom metode navn mitt metode navn var mer beskrivende.

### Oppgave 1.3 - Klassediagram

