

# INF1010 - Innleveringsoppgave 1

Maksimum 6 poeng

Temaet for denne uka er objektorientert programmering og generiske (parametriserte) klasser.

## Mus- og rottejakt

I denne oppgaven skal du lage klassene **Mus**, **Rotte**, **Katt**, **Bol** og **Oblig1**. Programmet skal simulere en katt som jakter på gnagere. Det er anbefalt at du tester klassene underveis.

### Mus

- En **Mus** må ha et **navn**.
- En **Mus** kan ha to tilstander. Den kan enten være **levende** eller **dod**. En mus vil starte som **levende**, men om den blir angripet av katten, vil denne tilstanden endre seg til **dod**.
- Dersom en **Mus** endrer tilstand, skal dette skrives ut til skjermen. For eksempel: **Musen Jerry gikk fra aa vaere levende til aa vaere dod**.
- En **Mus** må ha en metode som sier om den er død eller ikke.

Husk at variable skal være **private**, og man skal kun endre variablene ved hjelp av **public** metoder i **Mus**. Dette gjelder også de øvrige klassene.

### Rotte

- Ei **Rotte** må ha et **navn**.
- Ei **Rotte** kan ha tre tilstander. Den kan enten være **levende**, **skadet** eller **dod**. Ei rotte vil starte som **levende**, men om den blir angrepet av katten, vil rottas tilstand endre seg til **skadet**. Hvis ei **skadet** rotte blir angrepet, vil rottas tilstand endre seg til **dod**.
- Dersom ei **Rotte** endrer tilstand, skal dette skrives ut til skjermen. For eksempel: **Rotta Ronny gikk fra aa vaere levende til aa vaere skadet**.

- Ei **Rotte** må ha en metode som sier om den er død eller ikke.

*Hint: Rotte og Mus bør ha en metode for å bli angrepet.*

## **Bol<T>**

- Denne klassen skal være generisk. Det vil si at den skal ta inn en klassep-arameter som skal si noe om hva som skal bo i det aktuelle bolet. I denne oppgaven skal du lage et **Bol** for mus og et for rotter i **main**-klassen.
- Et **Bol** skal ha plass til *ett* dyr.
- Det skal være en metode for å sjekke om bolet er tomt.
- Det skal være en metode for å sette inn et nytt objekt. *OBS: Det skal kun være mulig å sette inn et nytt objekt dersom det ikke er noen som bor der fra før av.*
- Det skal være metoder for å hente ut objektet som ligger der.

Du trenger ikke å lage metode for å slette objektet som ligger der. Det vil si at det ikke er problematisk dersom det ligger en død mus eller rotte i bolet.

## **Katt**

- En **Katt** må ha et **navn**.
- En katt kan gå på jakt. Katten kan bare gjøre ett angrep per jakttur. Katten liker best mus, så den vil alltid sjekke i musebolet først.
  - Dersom musebolet er bebodd og musen er levende, vil den gjøre et angrep på musen og avslutte jakten. Det skal da skrives ut til skjerm at katten har gjort et angrep på musen. For eksempel: **Katten Tom gjorde et angrep på musen Jerry.**
  - Om musebolet enten er tomt eller musen som ligger der er død, vil den gå videre til rottebolet. Dersom rotta er levende eller skadet, vil den gjøre et angrep og avslutte jakten. Det skal da skrives ut til skjerm at katten har gjort et angrep på rotta. For eksempel: **Katten Tom gjorde et angrep på rotta Ronny.**
  - Dersom rottebolet er tomt eller rotta som er der er død, skal det skrives ut en melding om at det ikke fantes noen tilgjengelige gnagere, og jakten avsluttes. For eksempel: **Katten Tom fant ingen gnagere. Jakten avsluttes.**

*Hint: Det kan være lurt å ha en metode **jakt** som tar inn de to bolene som parameter. For eksempel:*

```
public void jakt(Bol<Mus> musebol, Bol<Rotte> rottebol)
```

## Oblig1

Dette skal være `main`-klassen til innleveringen. I denne klassen skal du gjøre følgende:

1. Lag `Bol<Mus>` musebol og `Bol<Rotte>` rottebol.
2. Lag en katt.
3. La katten gå på jakt og sjekk at det skrives ut at katten ikke fant noen gnagere.
4. Opprett ei rotte.
5. Sett rotta du lagde i forrige punkt inn i rottebolet.
6. La katten gå på jakt på nytt. Sjekk at rotta får status som skadet.
7. Lag en mus.
8. Sett musen inn i musebolet.
9. Lag en ny mus.
10. Prøv å sett inn musen i musebolet. Her burde du få en feilmelding om at det allerede er fullt.
11. La katten gå på jakt igjen. Sjekk at katten angriper musen og musen endrer tilstand.

En riktig løsning av oppgaven skal gi følgende utskrift når programmet kjøres:

```
Katten Tom fant ingen gnagere.  Jakten avsluttes.  
Katten Tom gjorde et angrep på rotta Ronny.  
Rotta Ronny gikk fra aa vaere levende til aa vaere skadet.  
Bolet er allerede fullt.  
Katten Tom gjorde et angrep på musen Jerry.  
Musen Jerry gikk fra aa vaere levende til aa vaere dod.
```

## Datastrukturtegning

Lag en tegning av datastrukturen som viser tilstanden til objektene etter å ha kjørt programmet over. Du kan bruke et bilteredigeringsprogram eller tegne for hånd. Om du tegner for hånd, kan du scanne den inn eller ta bilde av den. Er du usikker på hvordan du skal tegne datastrukturer, kan du finne det [her](#).

*Informasjon om hva som skal leveres og hvordan finner du i et eget dokument publisert samme sted som oppgaveteksten. Du finner det også på neste side.*

## Krav til innleveringen

1. Klassenavn og filnavn skal være identiske.
2. Klassenavn skal skrives med stor forbokstav.
3. Variabelnavn skal ha liten forbokstav.
4. Oppgaven må kunne kompilere og kjøre på IFI sine maskiner.
5. .class filer skal ikke leveres.
6. Ikke bruk æ, ø eller å i .java-filene (heller ikke som kommentarer eller utskrift).
7. Filene skal inneholde gode kommentarer som forklarer hva programmet gjør.
8. Programmet skal inneholde gode utskriftssetninger som gjør det enkelt for bruker å forstå.
9. Metodenavn skal skrives med liten forbokstav.
10. Koden skal være riktig indendert. Er du usikker, se Appendix J i Big Java.
11. Hver klasse skal ligge i sin egen .java-fil.

## Fremgangsmåte for innleveringer i INF1010

1. Lag en fil som heter README.txt. Følgende spørsmål skal være besvart i filen:
  - 1) Hvordan synes du innleveringen var? Hva var enkelt og hva var vanskelig?
  - 2) Hvor lang tid (ca) brukte du på innleveringen?
  - 3) Samarbeidet du med noen under innleveringen? Hvis ja, skriv brukernavn på den/de du samarbeidet med.
  - 4) Var det noen oppgaver du ikke fikk til? Hvis ja:
    - a) Hvilke(n) oppgave er det som ikke fungerer i innleveringen?
    - b) Hvorfor tror du at oppgaven ikke fungerer?
    - c) Hva ville du gjort for å få oppgaven til å fungere hvis du hadde mer tid?
2. Logg inn på Devilry.
3. Lever de 5 .java-filene, bilde av datastrukturen og README.txt i *samme innlevering*. Dersom du ønsker å levere igjen før fristen skal alle filene leveres på nytt, selv om kun noen har blitt endret.
4. Husk å trykke lever og sjekk deretter at innleveringen din er komplett.