

# Eksamens Rapport

## Unntakshåndtering:

Jeg har benyttet try-catch blokker for å fange opp SQLExceptions ved alle databaseoperasjoner, ett eksempel på dette er ved innsetting av nye samlekort. Dette sikrer at eventuelle feil ved databaseforbindelsen eller SQL-spørringene blir håndtert, og gir meg muligheten til å logge feilmeldingene og informere brukeren om feilen.

Jeg har også gjort sånn at om brukeren skriver inn ugyldige verdier i brukergrensesnittet, f.eks å skrive inn en sport som ikke finnes i databasen vil det komme en feilmelding om ugyldig sport ved bruk av dennekode strofen:

```
} else {  
  
    JOptionPane.showMessageDialog(mainFrame, "Ugyldig sport", "Error",  
JOptionPane.ERROR_MESSAGE);  
}
```

## Innkapsling:

I klassen Databasehandler har jeg laget private metoder og variabler for å sikre at de ikke kan endres direkte utenfor klassen. Dette sikrer at all tilgang og endring av data går gjennom de offentlige metodene jeg har definert, noe som gir bedre kontroll.

## Arv:

Jeg har brukt arv for å forenkle og gjenbruke kode.

For eksempel, i klassen Fotballkort som arver fra klassen «Kort», har jeg definert spesifikke egenskaper og metoder som gjelder for fotballkort, mens de generelle egenskapene og metodene er definert i superklassen Kort. Dette gjør koden mer oversiktlig og vedlikeholdbar, samt reduserer mengden duplisert kode.

## Forutsetninger og Uklarheter

Under utviklingen av denne applikasjonen har jeg antatt at alle nødvendige tabeller og kolonner finnes i databasen og har riktig struktur i henhold til spesifikasjonene. Eventuelle uklarheter i oppgaveteksten, slik som spesifikke krav til felter og formater, har jeg forsøkt å tolke på best mulig måte for å oppfylle oppgavens mål.