

KANDIDATNUMMER(E)/NAVN:

Filip Engan Ræder, Serghii Dykyy

DATO:

FAGKODE:

IDATT2003

STUDIUM:

Dataingeniør

ANT SIDER/BILAG:

12 /

FAGLÆRER(E) :

TITTEL :

Studentrapport om mappeprosjekt vår 2025

SAMMENDRAG:

*En kort oppsummering over hva denne rapporten handler om, de viktigste resultatene og konklusjonene.*

*De fleste vil lese dette sammendraget for å vurdere om det er interessant nok til å sette seg ned og lese hele rapporten. Bruk derfor god tid på å formulere sammendraget på en slik måte at du trigger leseren sin interesse til å lese hele rapporten.*

*Sammendraget skal kunne leses isolert, uten at man behøver å ha kjennskap til rapporten forøvrig.*

*Denne oppgaven er en besvarelse utført av student(er) ved NTNU.*

# Deklarasjon om KI-hjelpemidler

Har det i utarbeidingen av denne rapporten blitt anvendt KI-baserte hjelpemidler?

☒ Nei

☐ Ja

Hvis ja: spesifiser type av verktøy og bruksområde under.

---

## Tekst

☐ **Stavekontroll.** Er deler av teksten kontrollert av:  
*Grammarly, Ginger, Grammarbot, LanguageTool, ProWritingAid, Sapling, Trinkai.ai* eller lignende verktøy?

☐ **Tekstgenerering.** Er deler av teksten generert av:  
*ChatGPT, GrammarlyGO, Copy.AI, WordAi, WriteSonic, Jasper, Simplified, Rytr* eller lignende verktøy?

☐ **Skriveassistanse.** Er en eller flere av ideene eller fremgangsmåtene i oppgaven foreslått av:  
*ChatGPT, Google Bard, Bing chat, YouChat* eller lignende verktøy?

Hvis ja til anvendelse av et tekstverktøy - spesifiser bruken her:

---

## Kode og algoritmer

☐ **Programmeringsassistanse.** Er deler av koden/algoritmene som i) fremtrer direkte i rapporten eller ii) har blitt anvendt for produksjon av resultater slik som figurer, tabeller eller tallverdier blitt generert av: *GitHub Copilot, CodeGPT, Google Codey/Studio Bot, Replit Ghostwriter, Amazon CodeWhisperer, GPT Engineer, ChatGPT, Google Bard* eller lignende verktøy?

Hvis ja til anvendelse av et programmeringsverktøy - spesifiser bruken her:

---

## Bilder og figurer

☐ **Bildegenerering.** Er ett eller flere av bildene/figurene i rapporten blitt generert av: *Midjourney, Jasper, WriteSonic, Stability AI, Dall-E* eller lignende verktøy?

Hvis ja til anvendelse av et bildeverktøy - spesifiser bruken her:

---

## Andre KI-verktøy

☐ **Andre KI-verktøy.** har andre typer av verktøy blitt anvendt? Hvis ja spesifiser bruken her:

**V**

Jeg er kjent med NTNUs regelverk: *Det er ikke tillatt å generere besvarelse ved hjelp av kunstig intelligens og levere den helt eller delvis som egen besvarelse.* Jeg har derfor redegjort for all anvendelse av kunstig intelligens enten i) direkte i rapporten eller ii) i dette skjemaet

-----  
*Underskrift/Dato/Sted*

## INNHold

### Innholdsfortegnelse

<b>1</b>	<b>Introduksjon</b>	<b>1</b>
1.1	Bakgrunn	1
1.2	Kravspesifikasjon	1
1.3	Avgrensninger	2
1.4	Begreper/Ordliste	2
<b>2</b>	<b>Teori</b>	<b>3</b>
<b>3</b>	<b>Metode</b>	<b>4</b>
3.1	Utviklingsprosess	4
3.2	Verktøy	4
3.3	Bruk av KI verktøy	4
<b>4</b>	<b>Resultat</b>	<b>4</b>
4.1	Teknisk Design	4
4.2	Implementasjon	4
4.3	Testing	5
4.4	Utrulling til sluttbruker (deployment)	5
<b>5</b>	<b>Drøfting</b>	<b>5</b>
5.1	Drøfting av løsning/design	5
5.2	Drøfting av prosess	6
5.3	Drøfting av bruken av KI-verktøy	6
<b>6</b>	<b>Konklusjon - erfaring</b>	<b>6</b>

### Figurliste

Figur 1	Use Case diagram	2
Figur 2	Klassediagram som viser	5

### Tabelliste

Tabell 1	Begreper og ordliste	3
----------	----------------------	---

*[Denne rapporten inneholder ferdigdefinerte **stiler** som du/dere kan benytte for de mest vanlige avsnittene. Følgende stiler er definert:*

*Heading 1/Overskrift 1      Overskrift på nivå 1*

*Heading 2/Overskrift 2      Overskrift på nivå 2*

*Heading 3/Overskrift 3      Overskrift på nivå 3*

*Brødtekst      Standard tekst i et avsnitt. Benytt denne for all "vanlig" tekst*

*Definition      Benyttes hovedsakelig i avsnittet "TERMINOLOGI"*

*References      Benyttes i REFERANSER-avsnittet.*

*Comment      Denne grønne teksten. Fjern all tekst av denne typen i rapporten.]*

## 1 INTRODUKSJON

### 1.1 Bakgrunn

*[Dette er første kapitlet i den faglige rapporten. Det bør behandle bakgrunnen for oppgaven, eventuell oppdragsgiver, oppsummering av problemstillingen og/eller oppgaven som skal løses.]*

Mappeoppgaven som rapporten handler om er delt inn i to deler, å lage et stige spill ved hjelp av java, mer spesifikt javafx, og for den andre delen bruke grunnlaget fra den første delen til å lage enda ett brettspill. Videre er oppgaven delt inn i tre innleveringer, som har hensikten til å gi gruppen tilbakemelding på sitt arbeid og sette noen mål for studentene å orientere seg rundt. Mappeoppgaven er en stor del av programmering 2 og den ble gitt av læreren.

### 1.2 Kravspesifikasjon

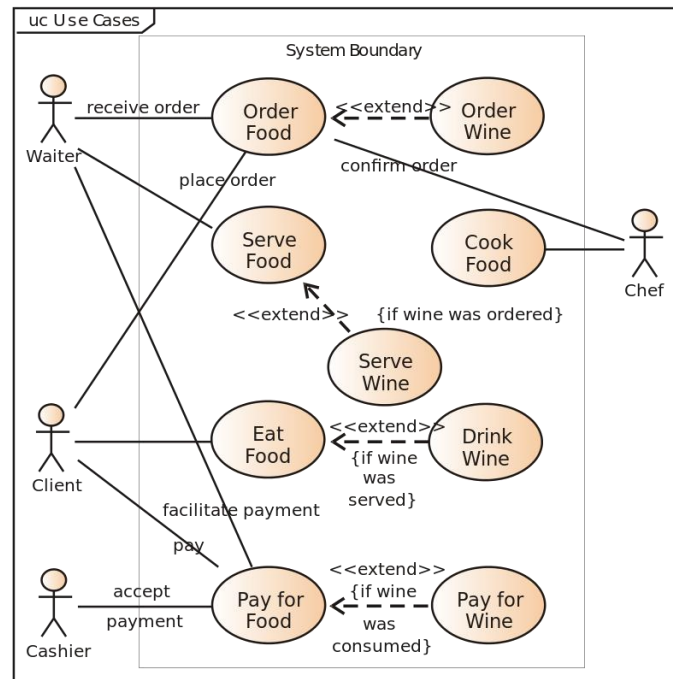
De funksjonelle kravene går ut på den første delen av prosjektet. Som basis for alle andre krav er at det første brettspillet skal være et stigespill, men dette er ikke alt. For det første skal brukerne kunne velge mellom flere varianter av stigespill, det vil si at stigespillet skal være bygd opp godt nok og med løs nok kobling at det samme grunnlaget skal kunne virke i flere situasjoner. Det skal helst være en variant med 90 felt. Deretter skal stigespillet ha kapasitet for opp til fem forskjellige spillere, men samtidig skal brukeren kunne velge hvor mange spillere er med i en runde. Til slutt skal spillerene kunne velge sine brikker fra et forhåndsdefinert utvalg, samt gi dem et navn. Selve spillet har sine egne funksjonelle krav. Hver bevegelse er bestemt av summen til to terningkast, og avhenging av hvilket felt spilleren lander på skal spillet utføre en spesifikk handling (Stige opp, Stige ned, osv.). Spillet avsluttes når en spiller når den siste ruten og blir kåret vinneren.

[Her beskriver du både de **funksjonelle** kravene og de **ikke-funksjonelle kravene** til løsningen du skal utvikle.

*Dersom det er gitt en kravspesifikasjon vil du kunne hente det meste av informasjon fra denne. Husk at du her IKKE skal beskrive noen av de valg du har gjort i prosjektet, eller det du konkret har utviklet i prosjektet.*

*Bruk her gjerne **UML-diagrammer** som **Use-Case**, **Aktivitetsdiagram** osv for å beskrive krav til funksjonalitet (NB! Uten å dra inn **hvordan** du/dere har løst det.).*

*Når du senere skriver **drøfting** og **konklusjon**, må du henvise tilbake til dette kapitlet og svare på om løsningen du har levert svarer på kravspesifikasjonen].*



Figur 1 Use Case diagram

### 1.3 Avgrensninger

[Er det gitt noen avgrensninger/begrensninger i oppgaven? Beskriv i så fall disse her. Avgrensninger kan f.eks. være teknologier dere **må** bruke, el.l. Dersom prosjektet ikke har noen avgrensninger, kan dette kapittelet utelates.]

Oppgaven har to avgrensninger. Den første er at begge spillene skal ikke være nettverksbaserte, det vil si at alle spillerne skal spille på samme datamaskin. Den andre begrensningen sier at spillene skal ha et grafisk brukergrensesnitt utviklet ved hjelp av JavaFX uten å bruke FXML. Kravene fører også til at spillene skal kjøres på en pc og ikke en smarttelefon.

### 1.4 Begreper/Ordliste

[Når man utvikler programvare for en kunde, er det viktig å etablere en felles forståelse for begreper/terminologi/ord som benyttes **av kunden**. Det er derfor svært vanlig å lage en "ordliste" og/eller en "Domene modell". Denne ordlisten er også et svært godt utgangspunkt for å finne frem til hvilke mulige **klasser** det kan være aktuelt å implementere i løsningen når denne skal utvikles. Bruk tid på denne slik at du har en god forståelse for begrepene.

Her kan man også bruke klasse-diagrammer for å illustrere hvordan **begreper** henger sammen. NB! Klasser i dette diagrammet er **ikke** klassene du har valgt å implementere i prosjektet.]

Begrep (Norsk)	Begrep (Engelsk)	Betyding/beskrivelse
Object orienterte språk	Object oriented language	En viss type programmeringsspråk som baserer seg på objekter og deres samspill i motsetning til oppbygningen av funksjonell programmering.
Lager	Storage	....
Konto	Account	En konto til en kunde i banken. En konto kan ha en balanse, og man kan sette penger inn eller ta ut penger fra konto
Transaksjon	Transaction	Når et beløp tas ut av en konto og føres inn på en annen konto, utføres en transaksjon. En transaksjon har en dato, en tid, hvilken konto beløpet trekkes fra, hvilken konto beløpet overføres til, samt beløpet.

*Tabell 1 Begreper og ordliste*

## 2 TEORI

Flere teorier ble brukt i dette prosjektet. I stor del har det vært hensiktsmessig bruk av de grunnleggende prinsippene av god koding i objekt basert programmering: innkapsling, arv, polyformisme og abstraksjon samt design prinsipper som passende kopling og kohesjon. Alle konsepter er tatt fra forelesningene i programmering 1 og programmering 2, som har gått i dybden på dem.

De grunnleggende prinsippene av OOP språk beskriver generelle retningslinjer som sier noe om kode design og de beste praksiser i alle Object orienterte språk som tar i bruk deres unike egenskaper. Innkapsling går ut på å samle kode i relevante klaser, samt kontrollere tilgang til klassene på en måte som ivaretar sikkerhet. Arv handler om å sette opp koden slik at relevante egenskaper og metoder kan bli gjenbrukt så mye som mulig, ikke kun for kodingens skyld, men også for å gjøre vedlikehold enklere. Polyformisme henger sammen med gjenbruk, men istedenfor generell arv sier polyformisme om å lage metoder som er åpne nok slik at de kan bli brukt i flere situasjoner. Til slutt betyr abstraksjon at det er best praksis å abstrahere koden og fremheve kun det viktigste slik at den skal være mer lesbar og igjen, at vedlikeholdet skal være enklere. Kopling og kohesjon er ikke en del av de fire grunnprinsippene, men de er fortsatt viktige



### 3 METODE

#### 3.1 Utviklingsprosess

[I dette kapittelet skal du fortelle hvilken prosess du/dere planla å følge. Den skal dekke prosessmodellen, hvorfor den ble valgt og hvordan den ble implementert.

Har dere jobbet i gruppe, så si noe om hvordan dere planla å organisere arbeidet (hvor ofte planla dere å møtes å jobbe?)

Hvordan planlagt prosess faktisk fungerte og hvilke endringer dere eventuelt gjorde skal beskrives i resultat-kapittelet og drøftes under Drøfting.

I en typisk mappe-oppgave der dere har jobbet gjennom flere stadier/deler av prosjektet, og fått tilbud om tilbakemelding, bør dette beskrives som en del av prosessen (altså **at** dere jobbet i f.eks. 3 deler med muntlig tilbakemelding etter hver del).]

#### 3.2 Verktøy

[Beskriv verktøy du/dere har benyttet for å løse prosjektet. Lag gjerne en tabell med navn på verktøy, versjon og hva verktøyet er benyttet til. Få med **samlige** verktøy (IDE, versjonskontroll, prosjektplanlegging/gjennomføring osv.)]

Verktøy	Bruksområde	Versjon
Github	Versjonskontroll	2025 Februar-
VScode	Koding	1.98
Teams	Digital kommunikasjon	25016.1904.3401.2239-

Tabell 2 Verktøy

#### 3.3 Bruk av KI verktøy

### 4 RESULTAT

#### 4.1 Teknisk Design

[Kapittelet om teknisk design beskriver det store bildet av valgt løsning. For et programvareutviklingsprosjekt vil dette vanligvis inneholde systemarkitekturen (klient-server, sky, databaser, tjenester, desktop-applikasjon osv.); både hvordan det ble løst, og, enda viktigere, hvorfor denne arkitektur ble valgt]

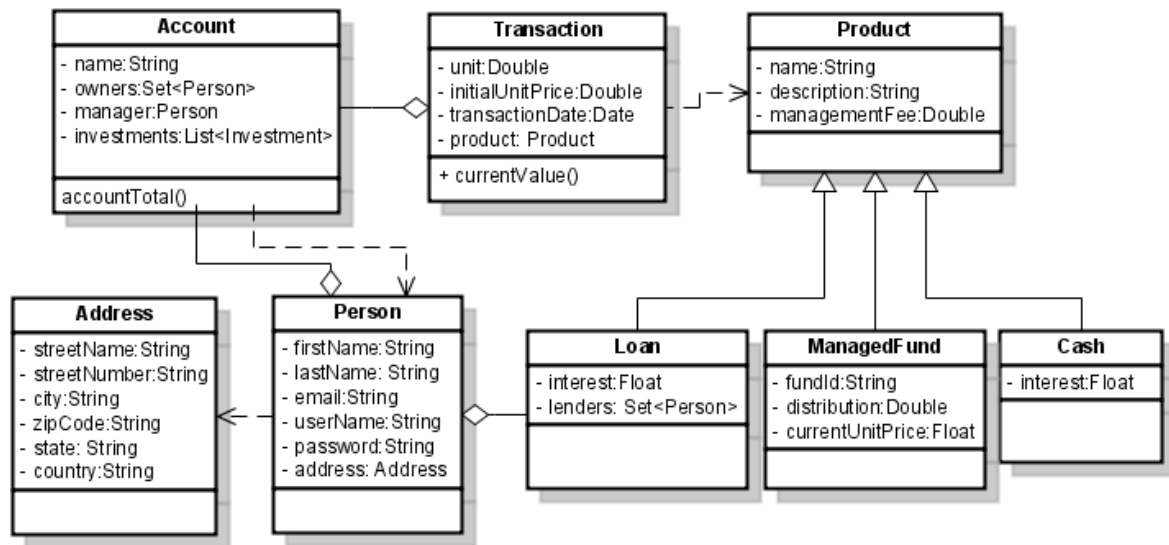
#### 4.2 Implementasjon

[Her beskriver du de tekniske detaljene til løsningen du har endt opp med. Hvilke eksterne biblioteker og rammeverk, APIer osv. har du/dere benyttet?

Vis med **klassediagrammer** hvordan klassene i løsningen din henger sammen. Husk at du kan vise flere klassediagrammer for å vise ulike sider ved løsningen (kun klasser som benyttes til brukerinteraksjon f.eks., eller kun klasser som utgjør forretningslogikken).

Bruk **aktivitetsdiagramm(er)** for å beskrive logikken/flyten i løsningen din. Fint også om du kan vise hvordan objekter av klassene dine **samhandler** for å løse **de mest sentrale** oppgavene i løsningen, da i form av **sekvensdiagramm(er)**.

Klassene dine bør beskrives i forhold til hvilken **rolle** og **ansvar** de har, men det er ikke nødvendig å beskrive hver enkelt metode eller felt til klassen. De står uansett dokumentert i Javadoc'en din.]



Figur 2 Klassediagram som viser...

### 4.3 Testing

[Beskriv hvordan løsningen din har blitt testet, både i form av enhetstesting, men også **brukertesting**. Hvor mange brukere har testet løsningen? Hvordan ble brukertesting gjennomført? Vis også eventuelt resultatene fra brukertesting her. (svar på spørreskjema el.l.)

Enhetstesting: si noe om hvilke klasser du/dere har valgt å skrive tester for, og hva **strategien** som dere/du har lagt opp til for å sikre best mulig testet kode.]

### 4.4 Utrulling til sluttbruker (deployment)

[Her beskriver du hvordan programvaren din gjøres tilgjengelig for sluttbruker. Rulles den ut å en web-server? I så fall hvordan? Lages det en desktop-applikasjon som bruker kan dobbeltklikke på for å starte? Eller kjøres applikasjonen fra Maven (mvn javafx:run) Osv.]

## 5 DRØFTING

### 5.1 Drøfting av løsning/design

[Her oppsummerer du/dere oppgaven. Hvor langt kom du/dere (resultat)? Hva fikk du/dere ikke gjort i forhold til oppgaveteksten? Hva var de store utfordringene/problemene du/dere møtte, etc..

Spesielt viktig er det å **drøfte din egen løsning i forhold til det du har lært om gode prinsipper for design av programvare** (robust kode, kodelstil, designprinsipper osv) som beskrevet i teori-kapittelet. I en godt skrevet rapport, er det ingen teorier som beskrives under teori-kapittelet som ikke drøftes under drøfting-kapittelet.

Husk å være **konkret**: Det holder ikke å skrive «Jeg/vi har designet en løsning som er i tråd med prinsippene om coupling og cohesion». Du/dere må «**bevise**» **hvorfor** dere kan påstå dette. Altså hente eksempler fra egen kode som underbygger teoriene om god design: «I klassen ..... har vi valgt å returnere .... fra metoden.... Dette bidrar til lav kobling....»

Her bør man også gjøre seg tanker rundt kvaliteten av det arbeidet som er nedlagt.

*Er de kildene du/dere bruker pålitelige, er det sprik mellom forskjellige kilder (og i så fall hvorfor), er det andre forhold som kan være med å gjøre noen av de vurderinger og valg du/dere har gjort usikre?]*

## 5.2 Drøfting av prosess

*[Fulgte dere prosessen som dere planla (og beskrev under kapittelet «Metode»)? Var det lurt, eller ikke? Hva fungerte bra hva fungerte mindre bra? Hva ville du/dere ha gjort annerledes neste gang?]*

## 5.3 Drøfting av bruken av KI-verktøy

*[Dersom du/dere benyttet KI-verktøy i denne oppgaven, drøft kort erfaringene dine/deres. Hva var KI-verktøyene nyttige for å løse? Hvilke svakheter oppdaget du/dere?]*

*Har du/dere ikke benyttet KI-verktøy dropper du dette kapittelet.]*

# 6 KONKLUSJON - ERFARING

*[Overbevisninger /erfaring som en er kommet fram til på grunnlag av det presenterte materialet.*

- Fikk du realisert hele problemstillingen fra kapittel «1.2 Kravspesifikasjon»?*
- Hva ville du ha gjort annerledes dersom du kunne begynn på nytt?*
- Hva slags begrensninger kan en forvente når en bruker løsningen?*
- Hva skal tas opp i fremtidige arbeid dersom du eller noen andre ville ha tatt utvikling videre?]*

## REFERANSER

*[Forfatter, årstall, tittel på bok eller artikkel, navn på tidsskrift eller forlag/utgiver, nr. eller dato for tidsskrift, sted som det vises til eller refereres fra i oppgaven.*

*Konkret for programmeringsemner: Regner med at du/dere kommer til å måtte slå opp litt i læreboka, så den er en innlysende referanse. Dersom du/dere i tillegg benytter internett, så list URL'er til sidene du/dere har benyttet.*

*Her er en god guide til hvordan oppgi referanser og hvordan referere til de (benyttes av IEEE): <https://www.bath.ac.uk/publications/library-guides-to-citing-referencing/attachments/ieee-style-guide.pdf> ]*

- [1] "Objects First With Java", Sixth edition, av Barnes og Kölling. ISBN ....
- [2] http://.....
- [3] Osv.

## VEDLEGG

*[Materiell som er utarbeidet eller innsamlet i tilknytning til rapporten, men som det ikke er naturlig eller hensiktsmessig å ta inn i hoveddelen, som feks brukerveiledning, skal tas inn som vedlegg.*

*Vedleggene skal være nummererte og ha en overskrift.*

*Har du/dere ingen vedlegg, så droppes dette kapitlet.]*