

KANDIDATNUMMER(E)/NAVN:

Elias André Willumsen
Harald Theodor Helland Velde

DATO:	FAGKODE:	STUDIUM:	ANT SIDER/BILAG:
28.02.25	IDATT2003	Dataingeniør, Programmering 2	14 /

FAGLÆRER(E) :

Majid Rouhani
Atle Olsø

TITTEL :

Group project report: Boardgame

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 først.

Dette prosjektet, utført av gruppe 10, forklarer fremgangsmåten, evaluasjonen og konklusjonen av utviklingsprosessen av et brettspill i løpet av et semester.

Prosjektet sikter på å utvikle to spill med samme struktur. Det minste fullførte produktet skal ha implementert *Stigespillet*, og på utvidet funksjonalitet skal *Ludo* bli implementert.

Etter en utviklingsprosess i tre faser, er dette resultatet.

[skrive mer her senere]

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

Deklarasjon om KI-hjelpeverktøy

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

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, Trinka.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:

Vi har ofte konsultert med diverse språkmodeller som ChatGPT for å gi en base vi kan utvide, samt til bli forklart feilmeldinger og løsninger som vi leser godt igjennom, før en eventuell implementasjon.

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:



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

1	Introduksjon	1
1.1	Bakgrunn	1
1.2	Kravspesifikasjon	1
1.3	Avgrensninger	1
1.4	Begreper/Ordliste	2
2	Teori	2
3	Metode	3
3.1	Utviklingsprosess	3
3.2	Verktøy	3
3.3	Bruk av KI verktøy	4
4	Resultat	4
4.1	Teknisk Design	4
4.2	Implementasjon	7
4.3	Testing	7
4.4	Utrulling til sluttbruker (deployment)	7
5	Drøfting	8
5.1	Drøfting av løsning/design	8
5.2	Drøfting av prosess	8
5.3	Drøfting av bruken av KI-verktøy	8
6	Konklusjon - erfaring	8

Figurliste

Figur 1 Use Case diagram	1
Figur 2 Klassediagram som viser	7

Tabelliste

Tabell 1 Begreper og ordliste	2
-------------------------------------	---

[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

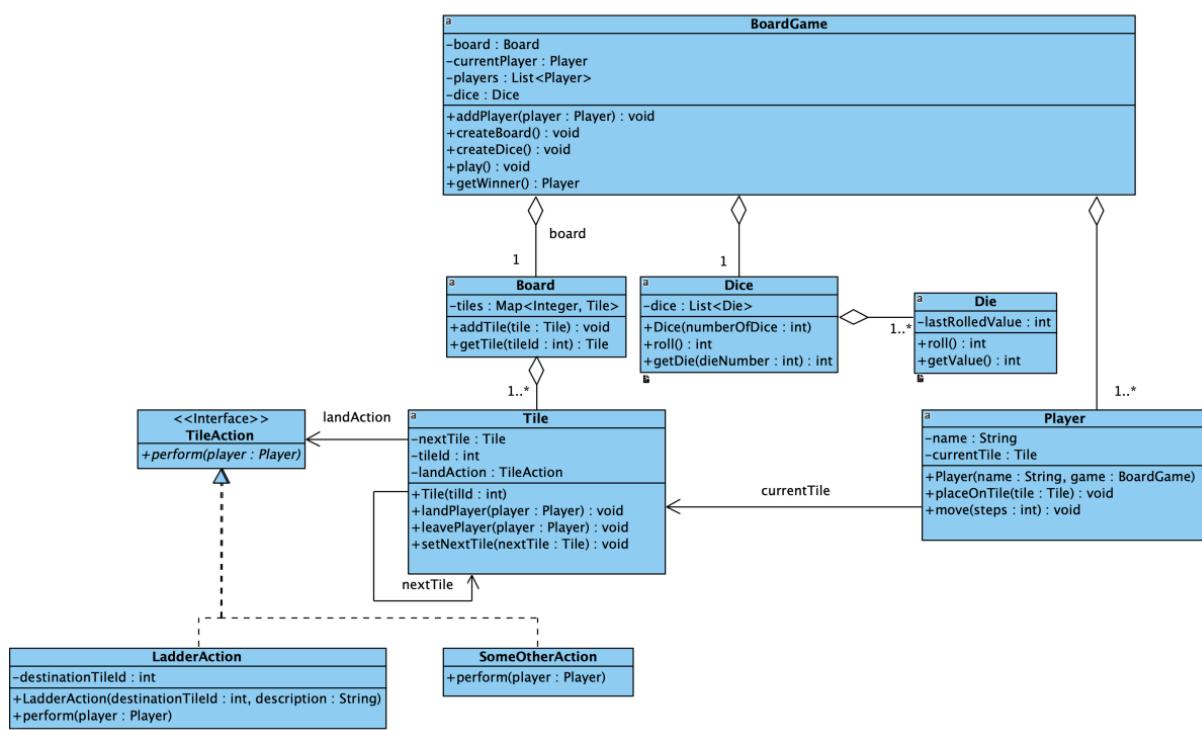
[Dette er første kapitlet i den faglige rapporten. Det bør behandle bakgrunnen for oppgaven, eventuell oppdragsgiver, oppsummering av problemstillingen og/eller oppgaven som skal løses.]

Prosjekt Board Game er en mappevurdering som vi løser gruppevis i faget IDATT2003 – Programmering 2 ved NTNU Trondheim. Målet med mappen er å lære å implementere GUI via JavaFX, samt konsepter som MVC, designmønstre og spill-logikk.

1.2 Kravspesifikasjon

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

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



Det ferdige prosjektet skal som minimum implementere et eksisterende brettspill, omgjort fra fysisk til digitalt. Spillet skal være *Stigespillet* som skal kunne bli spilt av opptil fem personer samtidig. Det forventes at enkelte felt har handlinger tilknyttet dem slik at en spiller som lander på et felt skal for eksempel bli flyttet frem eller tilbake et bestemt antall felt.

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 utlates.]

[Tror ikke det er noen avgrensninger]

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
Brett	Board	Spillbrettet som spilleren(-e) flytter brikkene sine på. Brettet inneholder felt med visse effekter.
Felt	Tile	Et felt er en posisjon spillbrikkene kan flyttes til, med en tilhørende effekt som å hoppe til neste felt eller å vinne.
Konto	Account	<i>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</i>
Transaksjon	Transaction	<i>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.</i>

Tabell 1 Begreper og ordliste

2 TEORI

[Her beskriver du de **teoriene** og **beste praksiser** som dere har benyttet ved utvikling av en programvare, som kobling, cohesion, SOLID osv. Beskriv **hvilke teorier** og **IKKE hvordan** dere har brukte teoriene. Det hører hjemme i resultat og diskusjon.]

Her må dere referere til **kilder** (som legges til i kapittelet **Referanser**).

Når du senere skal skrive drøfting og konklusjon, må du henvise tilbake til dette kapittelet og svare på om løsningen du har levert er løst i henhold til disse teoriene/beste praksisene.

Unngå å beskrive om grunnleggende teorier som «hva er OOP», «Programmeringsspråket Java» osv.

Skriv **kun teoripunkter som du faktisk har benyttet i ditt design, og som du faktisk diskuterer i diskusjonen** ☺]

Begrep (Norsk)	Begrep (Engelsk)	Betyding/beskrivelse
Observerer Mønster	Observer Pattern	Gir klasser en måte å oppdatere hverandre, men samtidig være løst avhengige. Kan sammenliknes med undervisningen fra en lærer til elev.

MVC (-mønsteret)	MVC (-pattern)	Struktur for grafisk brukergrensesnitt der man deler inn data (Model), utseendet (View) og kontrolleren (Controller) i tre deler for å lettere kunne utvikle et grafisk design med funksjonalitet.
Lettvekt	Flyweight	En måte å gjøre at applikasjonen bruker mindre minne ved at gjentakende data blir dratt ut av alle objektene og satt inn i ett enkelt objekt. Ofte brukt for bilder i spill.

3 METODE

[I dette kapittelet skal dere beskrive hva som skal til for å kunne reproduusere resultatet dere har fått. I programvareutvikling koker det som oftest ned til prosess/metodikk, og verktøy.]

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

Navn	Versjon	Forklaring
IntelliJ	IntelliJ IDEA 2024.3.3, build IU-243.24978.46	IDE-en brukt for utviklingen av javaprojektet.
git	git version 2.39.5 (Apple Git-154) og [Elias' git-versjon]	Versjonskontrollsysten som gjør at man kan lagre kode underveis, og gå tilbake og se og endre koden i historikken. Tett knyttet med GitHub.
GitHub	-	Brukt for å dele koden med andre over nettet og støtter utviklere å samarbeide på prosjekter og gir mer funksjonalitet til utviklingen som issues og pipelines.
Maven	Apache Maven 3.9.9	Maven er et verktøy for å sikre at filstrukturen er oversiktlig, og at java-applikasjonen blir riktig kompilert, testet og utdelt.
Word	-	Tekstbehandlingsverktøy brukt til å utforme rapporten.
Figma	-	Verktøy for å lage wireframes, som brukes for å vise det grunnleggende produktet visuelt med noe funksjonalitet.

3.3 Bruk av KI verktøy

[I starten av rapporten skal dere fylle ut en **KI-deklarasjon**. Du/dere kan selv velge om dere vil beskrive **hva** og **hordan** dere har benyttet KI-verktøy enten i skjemaet, eller her i dette kapittelet. Dersom dere velger å beskrive det her, så **henviser du/dere til dette kapittelet i KI-deklarasjons-skjemaet**. Beskriv også her **hvorfor** du/dere valgte å benytte KI-verktøy. Hva ønsket dere å oppnå?]

Som skrevet i KI-deklarasjonen, så har vi benyttet oss mye av KI-verktøy som tekstfullfører og kodeforslag.

Kode vi har fått utdelt har vi stilt oss kritiske til, og gått gjennom før en eventuell implementasjon.

Siden språkmodeller også samler inn kode, har mye av funksjonaliteten vi leter etter blitt implementert før, og modeller som ChatGPT kan lett finne frem passende kode. Det gjør at vi slipper å finne opp hjulet på nytt, derav blir utviklingsprosessen kortere enn om vi ikke hadde brukt KI.

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]



Edit Players

		Choose piece:
1	Elias	    
2	Harald	    
3		    
4		    
5		    

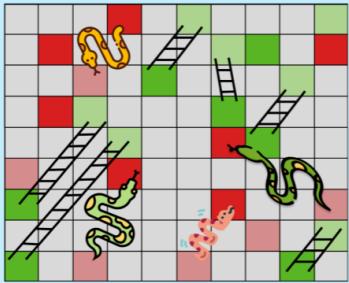
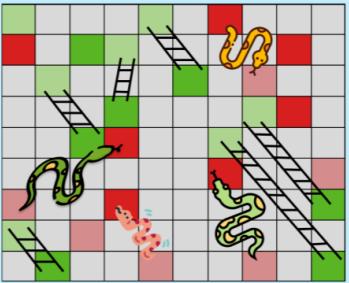
Download  Upload 

Done

Save or upload player list as CSV

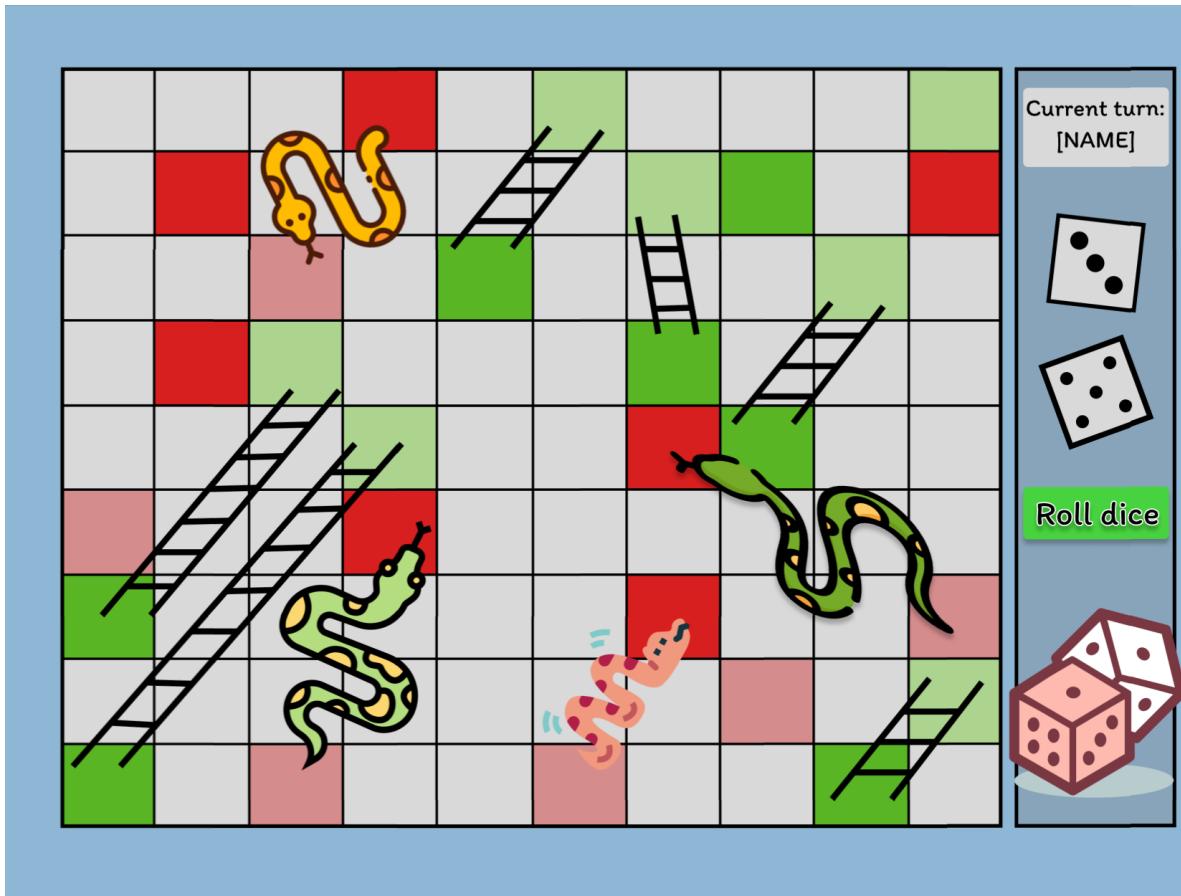
Choose Board



Snake Pit

Slippery Slope



After a thrilling and suspenseful game ...

The winner is ...

[NAME]

[Play again!](#)

[Back to menu](#)

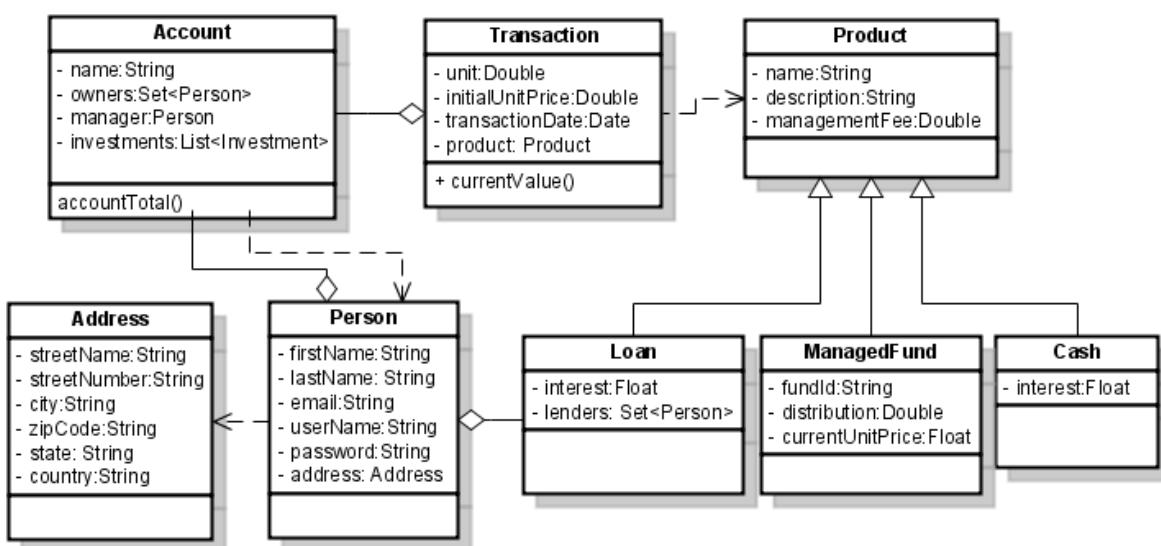
4.2 Implementasjon

[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 **aktivitetsdiagrammer** 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 **brukertestning**. Hvor mange brukere har testet løsningen? Hvordan ble brukertestning gjennomført? Vis også eventuelle resultatene fra brukertestingen 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, kodestil, 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 trå 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.

[4] Ikoner brukt for Wireframe:

[5] https://www.flaticon.com/free-icon/snake_616487

[6] <https://www.iconarchive.com/show/fluentui-emoji-flat-icons-by-microsoft/Ladder-Flat-icon.html>

[7] <https://www.creativefabrica.com/product/stick-man-doddle-graphic-icon-symbol/>

[8] https://www.flaticon.com/free-icon/snake_1303574

[9] https://www.flaticon.com/free-icon/snake_1934147

[10] https://www.flaticon.com/free-icon/snake_5374318

[11] https://www.flaticon.com/free-icon/snake_3873178

[12] https://www.flaticon.com/free-icon/dice_2325974

[13] https://www.flaticon.com/free-icon/dice-cube-from-top-view-on-the-face-with-three-dots_43723

[14] https://www.flaticon.com/free-icon/dice_142305

[15] <https://www.vecteezy.com/png/10252107-black-snooker-number-eight-png-file>

[16] <https://iconduck.com/icons/261713/puzzle-piece>

[17] https://iconscout.com/free-icon/coin-781_1093492

[18] https://www.flaticon.com/free-icon/dice_7469372

[19] https://www.iconfinder.com/icons/445793/board_chess_game_pawn_piece_white_icon

[20] https://www.flaticon.com/free-icon/upload_3580382

[21] https://www.flaticon.com/free-icon/download_3580085

[22] Alle ikoner for wireframe hentet 27.03.2025

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 kapittelet.]