

Prosjektbeskrivelse

TDT4102 Prosedyre- og objektorientert programmering

Institutt for datateknologi og informatikk

1 Innledning

Som en del av øvingsopplegget i faget TDT4102 - Prosedyre- og objektorientert programmering inngår det et prosjekt. Dette dokumentet beskriver prosjektet, lister opp krav, frister og generell informasjon. Prosjektet er en frivillig oppgave som gir studentene muligheten til å anvende programmeringsferdigheter i praksis. Målet er å utvikle et program som demonstrerer forståelse for prosedyre- og objektorienterte prinsipper samt bruk av grafikkbiblioteket som benyttes i faget.

Faget har totalt 12 øvinger, og prosjektet kan gi opptil 3 godkjente øvinger. Antall godkjente øvinger avhenger av størrelsen på prosjektet, og det er mulig å oppnå 0, 1, 2 eller 3 godkjente øvinger. Jo høyere kvalitet på prosjektet, desto flere øvinger kan godkjennes. I Tabell 1 kan en se viktige datoer og nøkkel informasjon om prosjektet.

Nøkkelpunkter	Detaljer
Prosjektstart	uke 11
Innleveringsfrist	fredag uke 15, klokken 23:59
Demonstrasjonsfrist	onsdag uke 18
Gruppestørrelse	1 eller 2 personer

Tabell 1: Nøkkelpunkter for prosjektet i TDT4102

2 Mål med prosjektet

Prosjektet skal bidra til at studentene:

- Blir kjent med prosedyre- og objektorientert programmering i praksis.
- Anvender fagets grafikkbibliotek aktivt i programutviklingen.
- Øver på strukturering av kode og dokumentasjon av programvare.
- Lærer å utvikle et program fra bunnen av, inkludert design, implementering og testing.
- Blir kjent med hvordan arv og unntakshåndtering fungerer i objektorientert programmering.

3 Generell informasjon

Prosjektet kan gjennomføres individuelt eller i grupper på opptil to personer. Uansett om en velger å jobbe alene eller sammen med en annen, **må en melde seg inn i en gruppe i Blackboard**. Merk at innleveringsmappen til prosjektet ikke vil være synlig før en har meldt seg inn i en gruppe. Det oppfordres til å bruke spørreforumet på Piazza dersom en ønsker å finne en partner å samarbeide med på prosjektet. Dersom prosjektet gjennomføres i par, forventes det at begge medlemmene er aktivt involvert i alle deler av prosjektet. I tillegg vil det forventes at et prosjekt utført i par vil være mer omfattende enn et individuelt prosjekt, både når det gjelder kompleksitet og omfang. Læringsassistenter vil vurdere at alle medlemmene har forståelse for alle aspekter ved prosjektet under demonstrasjonen.

En står fritt til å velge hva en ønsker å programmere. Prosjektet kan være basert på egne interesser, så lenge det oppfyller faglige krav beskrevet i seksjon 5 og demonstrerer forståelse for de grunnleggende prinsippene i faget. Dette gir mulighet for kreativitet og utforskning av ulike konsepter og ideer. Prosjektet skal utvikles fra bunnen av, og dersom det bygger videre på et eksempel, vil den eksisterende koden ikke bli vurdert som en del av prosjektet. Dersom prosjektet er basert på eksisterende kode må det presiseres eksplisitt i dokumentasjonen til prosjektet.

Overdreven bruk av kunstig intelligens (KI) er ikke tillatt. Det vil derfor legges stor vekt på forståelse av både koden og logikken bak den under demonstrasjonen. Det er viktig å kunne vise at prosjektet er gjennomført på en måte som demonstrerer forståelse for de grunnleggende prinsippene i faget. Merk at vurderingen vil baseres på egen innsats, og at en kun vil vurderes for kode en selv har skrevet.

Oppfordringer:

1. Dersom du jobber sammen i par, anbefales det å bruke **Git** for versjonskontroll og samarbeid. Selv om det ikke er et krav, vil bruk av Git gjøre samarbeidet lettere, da det gir mulighet for effektiv fildeling, versjonshåndtering og sporing av endringer. Det er også en viktig ferdighet for fremtidige kodeprosjekter og i arbeidslivet.
2. Det er sterkt anbefalt å starte tidlig med prosjektet. Å begynne tidlig gir deg tid til å utforske ideer, håndtere utfordringer som måtte oppstå, og sikre at du har god tid til å dokumentere og teste prosjektet ditt. Tidlig start gir også mulighet for å få tilbakemeldinger fra medstudenter eller læringsassistenter før prosjektet nærmer seg innlevering, noe som kan bidra til et mer vellykket sluttresultat.

4 Retningslinjer for innhold

Denne seksjonen beskriver hva som er tillatt og ikke tillatt når det gjelder innholdet i prosjektet ditt. Følg retningslinjene nøye for å sikre at arbeidet ditt er i tråd med kravene og unngå eventuelle problemer med plagiering.

- **Ikke tillatt:**

- Prosjekter basert på øvinger i faget eller øvinger fra tidligere år.

- Prosjekter basert på tidligere eksamensoppgaver.
- Prosjekter basert på eksempelprosjektene vist til i seksjon 6.

- **Tillatt:**

- Helt egne konsepter og ideer utviklet fra bunnen av.
- Videreutvikling av kjente konsepter, inkludert enkle spill (f.eks. versjoner av eksisterende spill).
- Andre applikasjoner og programmer som bygger på kjente ideer (f.eks. enkle verktøy, kalkulatorer, eller simuleringer).
- Det er tillatt å bygge videre på eksempler fra forelesningene, men dette må presiseres tydelig i prosjektets dokumentasjon, og vurderingen vil kun baseres på det du selv har utviklet, ikke på den opprinnelige koden.

Merk at det kjøres streng plagiatkontroll i faget. All form for kopiering og plagiering av andres kode kan føre til at prosjektet blir underkjent.

Dersom dere har spørsmål, spør på Piazza eller send mail til **tdt4102-undass@idi.ntnu.no**.

5 Krav til prosjektet

For at prosjektet skal kunne bidra til godkjente øvinger, bør det oppfylle så mange av følgende krav som mulig. Dersom man ønsker at prosjektet skal gi maksimalt antall godkjente øvinger, må alle krav være oppfylt.

Merk at godkjenning av alle krav alene ikke garanterer maksimal uttelling på antall godkjente øvinger. Prosjektets størrelse, kompleksitet og kvalitet vil også inngå i den helhetlige vurderingen. Eksempler på prosjekter er gitt i seksjon 6.

Listen med krav er som følger:

1. **Bruk av fagets grafiske bibliotek for GUI:** Programmet skal bruke fagets grafiske bibliotek til å implementere et grafisk brukergrensesnitt.
2. **Arv:** Minst en egendefinert klasse skal arve fra `AnimationWindow` og ha funksjonalitet som viser forståelse av arv.
3. **Unntakshåndtering:** Minst to forskjellige typer unntak skal håndteres i programmet med try-catch eller tilsvarende mekanismer.
4. **Kompilering og kjørbarhet:** Programmet skal kunne kompileres og kjøres uten feil.
5. **Dokumentasjon:** Prosjektet skal inkludere tilstrekkelig dokumentasjon. Gjerne en beskrivelse av hva programmet gjør, logikken bak implementasjonen og hva de viktigste funksjonene gjør. Dokumentasjonen skal også inneholde en liste over ressurser

og verktøy som er brukt, som for eksempel biblioteker, AI-verktøy (ChatGPT, Copilot, Grammarly, etc.) eller andre eksterne verktøy. Dersom KI-verktøy som ChatGPT eller Copilot er brukt, skal det tydelig dokumenteres hvilke deler av koden som er KI-generert. Dokumentet skal være på minst **250 ord** og kan leveres i et valgfritt format, som en README-fil, Word-dokument, PDF eller generert med Doxygen.

6. **Filhåndtering:** Programmet skal kunne **lese fra og skrive til minst en fil**, men kan også håndtere flere filer. Filene kan for eksempel brukes til å lagre spilltilstand, lagerbeholdning i en butikk eller innlegg i en dagbok.
7. **Refleksjonsnotat:** Hver gruppe skal levere et refleksjonsnotat som beskriver utviklingsprosessen og de valgene som ble tatt. Notatet skal ha **minst ett avsnitt per gruppemedlem**.

6 Eksempler

Det er laget noen eksempelprosjekter dere kan bruke som inspirasjon i arbeidet deres. For å finne eksemplene i VSCode må en holde inne Ctrl+Shift+P / Cmd+Shift+P og deretter skrive *TDT4102: Create Project From TDT4102 Template*. Her vil eksemplene ligge under “Prosjekt”-mappen. Prosjektene vil også inneholde eksempler på hvordan et prosjekt kan dokumenteres, men de inkluderer ikke refleksjonsnotater.

De samme kodeeksemplene finnes også i følgende GitHub-repositorier:

- Tic Tac Toe (tre på rad): <https://git.ntnu.no/TDT4102/tic-tac-toe>
- Sudoku: <https://git.ntnu.no/TDT4102/sudoku>
- Text-editor: <https://git.ntnu.no/TDT4102/text-editor>

7 Innlevering

- Prosjektet leveres som en **ZIP-fil** til Blackboard innen fristen i **fredag uke 15, klokken 23:59**, laget på samme måte som tidligere øvinger i faget. Dobbeltsjekk at ZIP-filen inneholder all relevant kildekode, nødvendige filer og dokumentasjon før innlevering. Merk at innleveringsmappen til prosjektet ikke vil være synlig før en har meldt seg inn i en gruppe i Blackboard.
 - Dersom **Git** brukes, kan du legge ved en lenke til et offentlig eller NTNU-hostet Git-repositorium som kommentar til innleveringen på Blackboard. **Merk at det ikke er nok å kun levere et GitHub-repositorium – du må fortsatt levere prosjektet som en ZIP-fil i Blackboard.**
- Prosjektet skal i tillegg demonstreres for en læringsassistent, og dette kan gjøres så snart prosjektet er levert, inkludert før påske. Siste frist for demonstrasjon er **onsdag i uke 18**, men det oppfordres til å gjennomføre fremvisningen så tidlig som mulig og ikke vente til de siste dagene før fristen. Mer detaljert informasjon om selve fremvisningen vil bli gitt senere.

8 Annet

De beste prosjektene får premie!