Utvikle et eget program

2IT

Utvikling

November 2025

Beskrivelse

Du har den siste tiden jobbet med Python og sett en rekke videoer som introduserer biblioteket Tkinter.

Du skal nå utvikle et eget program. Du velger selv hva programmet skal utføre, men det er viktig at du følger kravene (se vedlegg 1).

Du har tilgang til flere instruksjonsvideoer som viser enda flere muligheter med Tkinter, som du kan bruke som inspirasjon til å lage ditt eget program. Det er viktig at din løsning er av eget arbeide, og ikke bærer preg av andres kode.

Bruk gjerne T-kinter-videoene eller andre eksempler som inspirasjon, og hvis du ønsker å benytte andre biblioteker eller programmeringsspråk, kan du gjøre det.

I denne oppgaven skal du prioritere planleggingsfasen også, ikke bare sluttproduktet. Ha derfor som mål å ikke lage for avansert program, da du bare har tre uker på hele oppdaget.

Før du setter i gang med å programmere, skal du lage en ryddig og strukturert arbeidsplan. Denne skal leveres som en Word-rapport med egen forside med en passende tittel, ditt navn, klasse og årstall. Rapporten skal også ha automatisk innholdsfortegnelse og sidetall.

Du får tre uker til dette arbeidet (utviklingstimene), og ukene fordeles slik:

- **Uke 46**: Planlegging + grunnleggende struktur og funksjoner. Arbeidsplan leveres i løpet av uken.
- **Uke 47**: Utvikling av hovedfunksjonalitet + testing
- **Uke 48**: Finpuss, feilretting, og innlevering av endelig løsning (kodefiler + readme-fil)

Se vedlegg 2 for mal til arbeidsplan.

Se vedlegg 3 for eksempler på konsepter i Python.

Se vedlegg 4 for vurderingskriterier.

Lykke til!

Vedlegg 1 – Krav til kode

1. Modularitet

- Koden skal i størst mulig grad være delt opp i funksjoner, metoder eller moduler.
- Hver funksjon bør ha ett klart ansvar.

2. Kommentarer og dokumentasjon

- Viktige deler av koden skal være kommentert.
- Lag en egen readme-fil med:
    - Hva programmet gjør
    - Hvordan det brukes
    - Eventuelle installasjonskrav
    - Oversikt over eventuelle endringer i sluttprodukt sammenliknet med hva som ble planlagt
    - Forslag til forbedringer eller videreutvikling

3. Lesbarhet

- Variabelnavn og funksjonsnavn skal være beskrivende.
- Koden skal følge god innrykk og struktur.

4. Feilhåndtering

- Programmet skal håndtere feil på en kontrollert måte (f.eks. try/except, validering av input).
- Det skal ikke krasje ved vanlig brukerfeil.

5. Brukerinteraksjon

- Programmet skal ha en GUI-form for input/output.
- Programmet skal være intuitivt (brukeren skal forstå hvordan programmet brukes).

6. Testing

- Programmet skal være testet og fungere som forventet.

7. Filstruktur og navngivning

- Filene skal ha logiske navn, og være ryddig strukturert

8. Bruk av relevante konsepter

- Du skal vise at du har forstått og brukt sentrale konsepter i det valgte programmeringsspråket (f.eks. løkker, kontrollstrukturer, variabler og datatyper, operatorer, funksjoner, lister, strenger osv.). Se vedlegg 3.

9. Originalitet og refleksjon

- Programmet skal være egenutviklet.
- Du skal ha såpass eieforhold til din løsning slik at du klarer å forklare hvordan og hvorfor du har valgt sin løsning.

Vedlegg 2 – Mal til arbeidsplan

Arbeidsplanen skal leveres som en Word-rapport med egen forside, automatisk innholdsfortegnelse og sidetall. Innholdet skal organiseres slik:

1. Definer målet med programmet

- Hva skal programmet gjøre?
- Hvem er det laget for (deg selv, andre brukere, et spesifikt miljø)?

Tips: Skriv en kort problemformulering eller målsetning.

---

2. Lag en kravspesifikasjon

Del opp i:

- **Funksjonelle krav**: Hva skal programmet kunne gjøre? (f.eks. lagre data, vise grafikk, sende meldinger)
- **Ikke-funksjonelle krav**: Ytelse, brukervennlighet, sikkerhet, kompatibilitet osv.

---

3. Skissér programstruktur og funksjoner

- Lag en oversikt over hvilke moduler eller deler programmet skal ha.
- Tegn et flytskjema eller pseudokode for hvordan programmet skal fungere.
- Skriv ned hvilke biblioteker eller verktøy du vil bruke.

---

4. Lag en tidsplan for de tre ukene

Ta utgangspunkt i din timeplan, og fordel tiden slik at du har rom for:

- Planlegging + grunnleggende struktur og funksjoner
- Utvikling av hovedfunksjonalitet + testing
- Finpuss, feilretting, dokumentasjon og eventuell presentasjon

---

5. Planlegg testing og evaluering

- Hvordan skal du teste at programmet fungerer?
- Skal noen andre prøve det?
- Hvordan vil du dokumentere og evaluere resultatet?

Vedlegg 3 – Konsepter i Python

Grunnleggende konsepter

1. Variabler og datatyper

- int, float, str, bool
- Typekonvertering (int("5"), str(3.14))

2. Operatorer

- Aritmetiske (+, -, *, /, //, %, **)
- Sammenligningsoperatorer (==, !=, <, >, <=, >=)
- Logiske operatorer (and, or, not)

3. Kontrollstrukturer

- if, elif, else
- for-løkker og while-løkker
- break, continue, pass

4. Funksjoner

- Definere med def
- Parametere og returverdier
- Lokale og globale variabler

5. Lister

- Listeforståelse, som f.eks iterasjon (gå gjennom alle verdiene)

6. Strenger

- Formatering (f"Hei {navn}")
- Metoder (.lower(), .split(), .replace())

Mer avanserte konsepter

7. Feilhåndtering

- try, except, finally
- Egne feilmeldinger

8. Filbehandling

- Åpne, lese og skrive filer med open()
- with open(...) as f:-struktur

9. Moduler og biblioteker

- Importere standardbibliotek (math, random, datetime)
- Installere og bruke eksterne pakker (pip install)

10. Objektorientert programmering (OOP)

- Klasser og objekter (class, __init__)
- Metoder og attributter
- Arv og polymorfi

11. Databehandling og visualisering *(hvis relevant)*

- Bruk av pandas, matplotlib, numpy osv.

Vedlegg 4 – Vurderingskriterier

Kriterium	1	2	3	4	5	6
Modularitet	Koden er lite strukturert, få eller ingen funksjoner.	Koden har noen funksjoner, men ansvar og struktur er delvis uklar.	Koden har flere funksjoner med noe struktur.	Koden er strukturert med tydelig ansvar i funksjoner.	Koden er godt modulert med god struktur og ansvar.	Koden er svært godt modulert med optimal struktur og ansvar.
Kommentarer og dokumentasjon	Ingen kommentarer. Readme mangler.	Få kommentarer. Readme er ufullstendig.	Noen kommentarer. Readme inneholder noe info.	Kommentarer på viktige deler. Readme er delvis komplett.	Godt kommentert kode. Readme er informativ.	Meget godt kommentert. Readme er komplett og oversiktlig.
Lesbarhet	Dårlige navn og rotete struktur.	Noe beskrivende navn, men ujevn struktur.	Beskrivende navn og delvis ryddig struktur.	Klare navn og god struktur.	Meget gode navn og ryddig struktur.	Eksepsjonell lesbarhet og struktur.
Feilhåndtering	Ingen feilhåndtering. Programmet krasjer.	Enkel feilhåndtering, ikke konsekvent.	Delvis feilhåndtering. Noen brukerfeil håndteres.	God feilhåndtering. De fleste feil håndteres.	Gjennomtenkt feilhåndtering. Robust program.	Svært god feilhåndtering. Programmet tåler alle brukerfeil.
Brukerinteraksjon (GUI)	GUI mangler eller er ubrukelig.	GUI fungerer, men er lite intuitiv.	GUI er delvis funksjonell og forståelig.	GUI er funksjonell og brukervennlig.	GUI er intuitiv og godt designet.	GUI er svært brukervennlig og profesjonell.
Testing	Ingen testing. Mange feil.	Lite testing. Programmet fungerer delvis.	Noe testing. Programmet fungerer stort sett.	God testing. Fungerer som forventet.	Meget god testing. Få feil.	Svært god testing. Ingen feil.
Filstruktur og navngivning	Rotete filer og dårlige navn.	Delvis logisk struktur og navn.	Logisk struktur og navn.	God struktur og beskrivende navn.	Meget god struktur og navn.	Eksepsjonell struktur og navngivning.
Bruk av relevante konsepter	Få konsepter brukt.	Noen konsepter brukt med lav kvalitet.	Flere konsepter brukt med varierende kvalitet.	Relevante konsepter brukt korrekt.	Mange konsepter brukt effektivt.	Svært god bruk av relevante konsepter.
Originalitet og refleksjon	Lite originalt. Ingen refleksjon.	Noe originalitet. Lite refleksjon.	Originalt arbeid. Enkel refleksjon.	God originalitet og forklaring av valg.	Meget god refleksjon og eierskap.	Svært originalt og godt begrunnet arbeid.
Arbeidsplan og rapport	Mangler eller svært ufullstendig.	Delvis strukturert og relevant.	Strukturert med noe refleksjon.	Godt strukturert med tydelig plan.	Meget god rapport og refleksjon.	Svært god rapport med høy kvalitet.