

IN1000 - Høst 2019

Siri Moe Jensen
Geir Kjetil Sandve
Henrik Hillestad

Velkommen!

I dag

- Første innføring i Python
- <pause med registrering utenfor auditoriet>
- Hva fikk dere med dere – og hvem er dere? (mentimeter)
- Informatikk og noen nyttige begreper
- Praktisk informasjon og tips for IN1000

Viktige beskjeder

- Studieadministrasjonen registrerer oppmøte i pausen – utenfor auditoriet.
- Mikronfonlyd og skjerm tas opp og legges ut
- Se semestersiden m/ lenker for informasjon!
<http://www.uio.no/studier/emner/matnat/ifi/IN1000/h19/>
- Les eller videresend studentkonto-mail!!

Studieadministrasjonen

- Opptak
- Fravær
- Studieveiledning
- Alt administrativt/ formelt
- studieinfo@ifi.uio.no
- 4. etasje i Ole-Johan Dahls hus, oppgang B

Gruppelærere og rettere

Gruppeundervisning

Retting av obliger

Faglige spørsmål



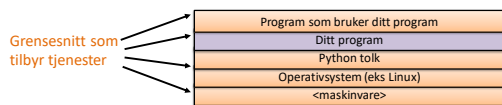
Og mye annet - prøv 😊

Informatikk

Informatikk er læren om hvordan datasystemer konstrueres og brukes.

Et datasystem består av en eller flere **datamaskiner** som kjører **programvare** og kan være knyttet til et eller flere **nettverk** for overføring av **data**

Din maskin: Lagdeling og grensesnitt



- Hvert **lag** har sine arbeidsoppgaver, og tilbyr disse gjennom et **grensesnitt** til laget over
- Grensesnittet gjør det mulig å **abstrahere** bort (ikke kjenne til) *hvordan* laget under løser oppgavene sine

Abstrahere/ abstraksjon

Wikipedia:

- "...tenkemåter der en skiller ut enkelte elementer, for å betrakte resten.."

Store Norske Leksikon:

- "...å sammenfatte enkeltfenomener under ett begrep idet man ser bort fra avvikende detaljer"
- "...Det er en måte å komme fram til fellestrekk eller felles egenskaper og klassifiseringer ..."

Modell

SNL:

<...> brukes ordet om en skjematisering eller en matematisk modell

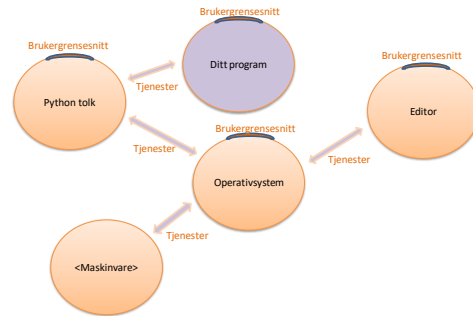
Slik brukt innebærer en modell alltid en **forenkling** basert på et utvalg av de viktigste trekk.

Wikipedia:

Teoretiske modeller gir et **sterkt forenklet** bilde av virkeligheten.

Forenklingen dreier seg om å fokusere på de **trekk som anses å være vesentlig** for det som skal forklares, samtidig utelukke alt som er uvesentlig

En annen modell



Objektorientert programmering

- Når programmet ditt utføres, foregår det (bak kulissene) vha **objekter** som samarbeider.
- Et objekt holder rede på **data** som hører sammen, og kan utføre **handlinger** knyttet til disse
- *Hvilke data og handlinger er definert i objektets **klassedefinisjon***
- Objektene tilbyr tjenester til andre deler av programmet gjennom sine **grensesnitt**
- Objektorientering hjelper oss å **organisere** programmene våre bedre, ved å **modellere** og **simulere** komplekse systemer

Vi kommer tilbake til hvordan vi kan opprette og bruke objekter, og hvordan vi kan skrive egne klasser som mønstre for objekter!

Oppsummering

- Informatikk krever at vi abstraherer bort forskjeller og kompleksitet vi ikke trenger i det vi jobber med
- For eksempel modeller og metaforer som beskriver hva som skjer "bak kulissene" når programmene våre utføres
- Et fenomen kan modelleres på mange ulike måter avhengig av hvilke egenskaper/ aspekter man er interessert i/ trenger å si noe om
- **Objektorienterte språk** har mekanismer som hjelper oss med dette når vi skriver programmer

Informatikk-studiet og IN1000

- Mål for emnet
 - Solid grunnlag for videre studier
 - Vekt på grunnleggende konsepter og begreper
 - Trening i programmering ved hjelp av disse i Python
 - Ikke et hurtigkurs i praktisk programmering
- Ingen krav til forkunnskaper, men helt nødvendig med jevnt arbeid!
 - Undervisningen forutsetter Forkurs
 - Bruk nettkurset ved behov/ for repetisjon (lenke fra semestersiden)

Programmert før?

- Noen fallgruver
- Følg undervisning og løs obliger, ikke «mist toget»
- Møt opp på Fredags-Python!
- Mye bakgrunn i Java? Infomøte/ opptaksprøve
 - IN2030 - Prosjektoppgave i programmering
 - onsdag 21.8 kl 14.15 på Shell
- Overskudd til ekstra studiepoeng?
 - IN1150 – Logiske metoder

Semesteret i IN1000

- Uke 1-6: Grunnleggende elementer i programmering
- Uke 7-11: Objektorientert programmering, større programmer, kompleksitet
- Uke 12-15: Prøveeksamen, eksamenstips, repetisjon
- Uke 16: Eksamen

Arbeidet i [IN1000](#)

- Undervisningstilbud, 6 timer (+4)



- Selvstendig arbeid, 7+ timer

Arbeid med teori, Trix-oppgaver og obligatoriske oppgaver

Ressurser: Se [semestersiden](#)
Bøker/ notater/lysark/ opptak, kollokvier, Piazza, gruppelærere, ..

Tips for programmeringsstudier

- Programmering handler om å definere og løse problemer/ finne fremgangsmåter, dvs
 - Læringen skjer mens du klør deg i hodet
 - Lær å like det 😊
- Start koding selv om du ikke forstår alt, eksperimenter
- Men finn alltid ut hvorfor en feil "forsvant"!
- Jobb sammen med andre, diskuter løsninger og "hva skjer her"
- Jobb alene, svett over problemer 😊
- Du kan ikke hoppe over pensum i programmering