Programmering for Lærarar i VGS

Hans Georg Schaathun

12. juni 2019

Velkommen til etterutdanningskurs i programmering for lærarar i vidaregåande skule. Læringsmål:

1. Kjenna grunnleggjande prinsipp for imperativ programmering.
2. Kunna bruka Jupyter og Python som hjelpemiddel for å studera matematiske, statistiske og numeriske problem.
3. Meistra nokre døme på programmeringsoppgåver som kan vera eigna i realfagsundervisinga i VGS.
4. Ha eit visst oversyn over andre tilnærmingar til programmering.

1 Normaltimeplan

Kl.	Aktivitet
8.30	Førelesing
9.30	Øving med vegleiing
11.30	Lønsj
12.00	Førelesing
13.00	Øving med vegleiing
15.00	Førelesing
16.00	Slutt

Klokkesletta er omtrentlege, med pausar 10–15 min. mellom kvar økt. Opplegget er basert på dialog og vert tilpassa etter kvart. Me skal do halda avsluttinga kl. 16.

2 Kurs 27. mai 2019

Fyrste utgåve av kurset vart halde 27. mai 2019 med 40 plassar. Kurset var vellukka, men vart tilpassa deltakarane på staden, og fekk difor ein annan form enn planlagd.

2.1 Jupyter Notebook

1. *Jupyter Notebook som skriveprogram for matematisk tekst*

   Me arbeider med matematiske problem, og drøftar løysingar i prosa. Me programmerer utvalde steg som egnar seg for maskinrekning. Jupyter lèt oss fletta saman prosa og programmering.

2. Matematisk funksjon og plotting: `Jupyter/plotting.ipynb`
   - Funksjonsdefinisjonen
   - `print`
   - Variablar, tilordning
   - Kommandosatsar
   - Plotting
   - Dokumentasjon, kjelder til fleire moglegheiter

3. Øving `Jupyter/xplotting.ipynb`

2.2 Imperative programmering

I den fyrste delen såg me berre på korte seriar med instruksjonar, og stort sett kunne me ta éin instruksjon åt gongen. Programmering handlar om å setja saman instruksjonar, og lata fortsetjinga avhengja av kva som skjer undervegs.

1. Halveringsmetoden: `Jupyter/bisection.ipynb`
   - Vilkårssats
   - Løkker
   - Algoritmisk funksjon
   - Rekursjon

2. Sekantmetoden: `Jupyter/secant.ipynb`

2.3 Andre paradigme og idéar

2.3.1 Føredrag

1. Kva er programmering?

2. Tilstandsmaskina

2.3.2 Objektorientert

Det dominerande paradigmet for å utvikla større programsystem er objektorienter programmering, med Java som eitt av dei mest populære språka. Java er òg det føretrukne språket for app-utvikling på Android, som er spesielt tilpassa Java.

Greenfoot og BlueJ er pedagogiske hjelpemiddel for å læra objektorientert programmering med Java. BlueJ er tilpassa generell Java-programmering, og er mest interessant ved systematisk opplæring i objektorienter programmering.

Greenfoot er interessant fordi det legg godt til rette for agentbasert simulering. Her går det an å eksperimentera med simulering av

1. bytte/rovdyr (biologi/økologi)
2. demografi (samfunnsfag)
3. mekaniske system (fysikk)

2.3.3 Programmering for vevlesaren (Javascript)

Javascript er eit interessant språk fordi ein skriv kode som køyrer direkte i vevlesaren. Dette gjer det til det mest platformuavhengige språket i dag. Det er det einaste som brukaren kan køyra direkte frå ei vevside utan å installera lokalt.

Javascript vert brukt både til små snuttar og komplekse system. Ein kan til flotte presentasjonar ved å tenkja mest i HTML, og fylla inn med små javascriptsnuttar for å få spesielle effektar.

Spesielt er terskelen for visualisering lågare i Javascript enn i dei fleste andre språk. Dette gjer det velegna til ein del relevante elevoppgåver i simulering og dataanalyse, t.d.

1. simulering av mekaniske system (fysikk)
2. dataanalyse (matematikk og statistikk)

3 Disposisjon

Disposisjonen er mest å sjå som ei idémyldring til vidare kurs.

3.1 Jupyter Notebook

1. *Jupyter Notebook som skriveprogram for matematisk tekst*

   Me arbeider med matematiske problem, og drøftar løysingar i prosa. Me programmerer utvalde steg som egnar seg for maskinrekning. Jupyter lèt oss fletta saman prosa og programmering.

2. Matematisk funksjon og plotting: `Jupyter/plotting.ipynb`
   - Funksjonsdefinisjonen
   - `print`

- Variablar, tilordning
- Kommandosatsar
- Plotting
- Dokumentasjon, kjelder til fleire moglegheiter

3. Øving `Jupyter/xplotting.ipynb`

3.2 Imperative programmering

I den fyrste delen såg me berre på korte seriar med instruksjonar, og stort sett kunne me ta éin instruksjon åt gongen. Programmering handlar om å setja saman instruksjonar, og lata fortsetjinga avhengja av kva som skjer undervegs.

1. Halveringsmetoden: `Jupyter/bisection.ipynb`
   - Vilkårssats
   - Løkker
   - Algoritmisk funksjon
   - Rekursjon
2. Sekantmetoden: `Jupyter/secant.ipynb`

3.3 Andre døme

3.3.1 Oppsummering

Foilar

3.3.2 Trapesmetoden

3.3.3 Gausseliminasjon

3.4 Slumptal og terningkast

- Terningkast: `Jupyter/random.ipynb`

3.5 Finansmatematikk

- Finansmatematikk: `Jupyter/finance.ipynb`

3.6 Gruppeprosjekt: ei oppgåve til klassa

Oppgåve 1. *Arbeid i grupper med felles undervisingsfag. Drøft interessant elevoppgåver i faget. Finn éi eller to oppgåver som (kanskje) kan løysast programmatisk og presenter dei for heile kursgruppa.*

3.7 Andre paradigme og idéar

Foilar

3.7.1 Objektorientert

Det dominerande paradigmet for å utvikla større programsystem er objektorienter programmering, med Java som eitt av dei mest populære språka. Java er òg det føretrukne språket for app-utvikling på Android, som er spesielt tilpassa Java.

Greenfoot og BlueJ er pedagogiske hjelpemiddel for å læra objektorientert programmering med Java. BlueJ er tilpassa generell Java-programmering, og er mest interessant ved systematisk opplæring i objektorienter programmering.

Greenfoot er interessant fordi det legg godt til rette for agentbasert simulering. Her går det an å eksperimentera med simulering av

1. bytte/rovdyr (biologi/økologi)
2. demografi (samfunnsfag)
3. mekaniske system (fysikk)

3.7.2 Programmering for vevlesaren (Javascript)

Javascript er eit interessant språk fordi ein skriv kode som køyrer direkte i vevlesaren. Dette gjer det til det mest platformuavhengige språket i dag. Det er det einaste som brukaren kan køyra direkte frå ei vevside utan å installera lokalt.

Javascript vert brukt både til små snuttar og komplekse system. Ein kan til flotte presentasjonar ved å tenkja mest i HTML, og fylla inn med små javascriptsnuttar for å få spesielle effektar.

Spesielt er terskelen for visualisering lågare i Javascript enn i dei fleste andre språk. Dette gjer det velegna til ein del relevante elevoppgåver i simulering og dataanalyse, t.d.

1. simulering av mekaniske system (fysikk)
2. dataanalyse (matematikk og statistikk)

4 Idéar til vidaregåande kurs

4.1 Python utan Jupyter

4.1.1 iPython

4.1.2 Køyrbare scripts

4.2 Grafikk og animasjon

4.3 Programmering i fugleperspektiv

Kva er programmering?

- Kode - maskinsyntaks
- Deklarativ programmering
  - Funksjonell programmering
  - Logisk programmering
- Imperativ programmering
  - Objektorientert programmering
- Hybridspråk

Kva er ein algoriritme?
Ulike språk

1. Java
   a) Greenfoot
   b) Java
   c) Java EE
2. JavaScript
   a) Angular
   b) D3
3. Haskell

Språka kan alt, men er ikkje best til alt.