Kravik

Programmering og geometri

Denne artikkelen ser på hvordan programmering kan implementeres i klasserommet i grunnskolen og hvordan programmering kan brukes som et verktøy for utforskning av geometri i matematikkfaget. Artikkelen tar for seg en måte å implementere programmering og geometri i et undervisningsopplegg i en 9. klasse.

Læreplanene vi har i dag, inneholder bruk av digitale verktøy for elevene i matematikkundervisningen (Utdanningsdirektoratet, 2006). De nye læreplanene fører videre bruken av digitale verktøy, samt også programmering og algoritmisk tenkning (Utdanningsdirektoratet, 2019). Hvordan dette skal implementeres, er det ikke gitt føringer for, men spesifikke mål når det gjelder programmering og algoritmisk tenkning, er gitt for ulike årstrinn. For eksempel innføres begrepene variabler, vilkår og løkker fra 4. trinn (Utdanningsdirektoratet, 2019). Ulik kunnskap blant lærere i programmering og hvordan man tilrettelegger for matematikk og programmering i samspill, kan skape utfordringer og ulikheter mellom skoler og klasserom.

Grunnleggende ferdigheter i læreplanen trekker frem bruken av programmering i utforskning og problemløsning (Utdanningsdirektoratet, 2019). Undersøkende matematikkundervisning trekkes frem som ett av fem viktige temaer i matematikkdidaktikken. Motivasjonen til elevene for å lære matematikk kan påvirkes positivt ved en undersøkende matematikkundervisning (Wæge, 2007).

Reiar Kravik
Universitetet i Sørøst-Norge
reiar.kravik@usn.no

Bakgrunn og undervisningsopplegg

Et undervisningsopplegg ble utviklet for å implementere programmering i klasserommet. Dette ble testet i en 9. klasse med 21 elever. Klassen hadde begrenset erfaring med programmering og hadde ikke tidligere brukt tekstbasert programmering.

Undervisningsøkten tok for seg geometriske figurer og innføring av tekstbasert programmering i Python, med datapakken «Turtle graphics». Med denne datapakken flytter man en figur på skjermen og lager figurer, på lik linje med at man bruker penn og papir. Enkle kommandoer som *forward*, *left* og *right* kan brukes til å lage visuelle figurer av ulik karakter. Et eksempel er gitt i figur 1, der man først beveger seg fremover 100 piksler, snur seg 90 grader mot venstre og deretter beveger seg fremover 100 piksler. Resultatet blir to like lange streker som står vinkelrett på hverandre.

I klasserommet ble elevene gjort kjent med de mest brukte kommandoene forward(), backward(), left() og right() i en kort introduksjonsøkt på 5 minutter. Det ble også vist for elevene

Kravik, R. (2020). Programmering og geometri. *Tangenten – tidsskrift for matematikkundervisning, 31*(2). 7–11.

www.caspar.no/tangenten

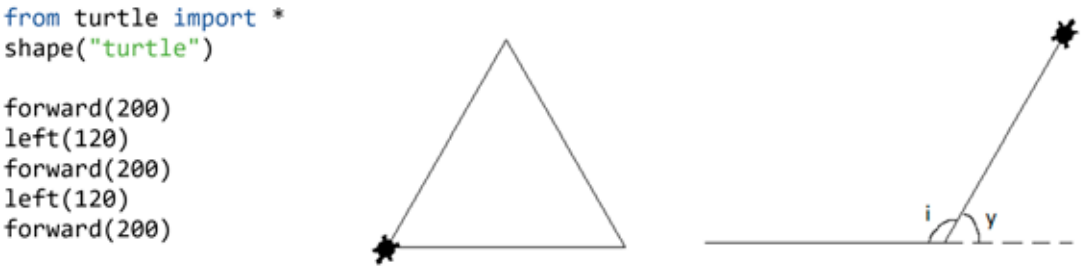
Figur 1: Eksempel på kode (venstre), visuell figur (høyre)

hvordan man programmerer et kvadrat. Elevene satt deretter sammen to og to og fikk et oppgaveark med oppgaver knyttet til geometriske figurer og programmering (Kravik, 2019). Noen av oppgavene besto i å lage figurer som rektangel, ulike trekanter og andre geometriske figurer som elevene ble utfordret til å finne selv. Tanken med at elevene skulle finne ulike figurer selv, var å få dem til å reflektere over hva vi har av ulike geometriske figurer, og hvordan disse ser ut. Det ble også gitt oppgaver med å lage figurer som skulle speiles, i tillegg til tredimensjonale figurer. Da elevene hadde kommet gjennom oppgaven, fikk de utdelt et stort utvalg kommandoer tilknyttet «Turtle graphics»-pakken, som de kunne prøve ut på egen hånd.

Utprøving i klasserommet

Elevene tok tak i kommandoene umiddelbart. Et eksempel på kode og figur til en likesidet trekant er gitt til venstre og midten av figur 2. Figuren til høyre viser noen av utfordringene, vinklene, elevene måtte ta stilling til når de jobbet med programmeringen av denne typen figurer. Elevene har fra tidligere arbeidet med likesidede trekanter i matematikkundervisningen og har en tanke om at vinklene er like i en likesidet trekant, 60 grader. Man kan tenke seg at elever derfor vil starte med å prøve en vinkel på 60 grader når de skal lage en likesidet trekant. Skilpadden går fremover 100 piksler og snur seg 60 grader mot venstre basert på ytre vinkel, beskrevet med y på figur 2, og ikke indre vinkel (i). Elevene får dermed en direkte (visuell) respons på at noe er feil i koden, ved at de ikke oppnår det de hadde sett for seg ved likebeinte trekanter. Elevene må reflektere rundt vinkelen og hvilke endringer de må gjøre i koden. Elevene ser da at utgangspunktet til «skilpadden» er viktig, og at programmet ser på ytre vinkler når de lager figurer som denne trekanten.

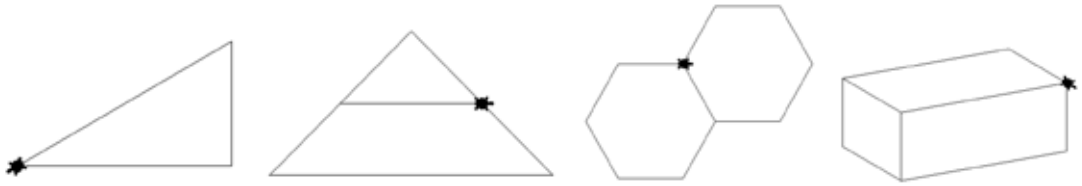
Etter som elevene ble ferdige med de spesifikke figurene i oppgavene, skulle de tenke på andre geometriske figurer og få frem disse. Noen figurer som elevene programmerte er gitt i figur 3. Vi ser en rettvinklet trekant, formlike trekanter, to sekskanter, disse sekskantene er speilet om en side av sekskanten, og et rett prisme. Det kan være ulike utfordringer knyttet til de ulike figurene. Når det gjelder den rettvinklede trekanten, kan for eksempel utfordringene ligge i å velge øvre vinkel og lengde på hypotenusen i trekanten, for å komme tilbake til utgangspunktet. Dette kan elevene gjøre ved prøving og feiling, og da få en direkte respons på om de fullfører trekanten eller ikke. Alter-



Figur 2: Elevbesvarelse med kode og figur (midten) av likesidet trekant. Utfordringer til indre(i)/ytre(y) vinkel vist til høyre.

Kravik, R. (2020). Programmering og geometri. *Tangenten – tidsskrift for matematikkundervisning, 31*(2). 7–11.

www.caspar.no/tangenten



Figur 3: Ulike resultater fra elevene, rettvinklet trekant, formlikhet, sekskanter, speiling og tredimensjonale figurer, her rett prisme.

nativt kan de bruke det de kan om rettvinklede trekanter og Pytagoras, og deretter regne ut lengden av hypotenusen. Utfordringer for elevene knyttet til rett prisme kan være: Hvordan man starter, og hvordan man vil at rett prisme skal ligge i planet.

Det er store muligheter for å kunne differensiere oppover når man underviser i eller bruker programmering som et verktøy i skolen. Det viktigste er at man legger til rette for grunnnivået blant elevene. Dette med differensiering ble også synlig i klasserommet, der enkelte elever programmerte trekanter, mens andre programmerte tredimensjonale figurer (figur 3). Det ble gitt små hint til elevene for å starte på nye utfordringer, for eksempel: «Kan dere lage en tredimensjonal boks?», med tilhørende skisse på tavlen. Det ble spesielt engasjement rundt dette med tredimensjonal rett prisme fra to av gruppene. Elevene løste oppgaven med prøving og feiling. En av gruppene fikk vise hvordan figuren deres ble i plenum, og hva de hadde oppnådd. Det er mange ulike programmeringsspråk og mye tilgjengelig materiale både i form av læringsvideoer og nettsider, noe som gir store muligheter for utvikling av kompetanse rundt programmering for elevene. Elevene kan få motivasjon av den direkte visuelle tilbakemeldingen ved en figur. Wæge (2007) trekker frem at undersøkende matematikkundervisning påvirker motivasjonen til elevene positivt. Når det gjelder programmering, er det gode muligheter for å kunne utfordre elevene med fleksible oppgaver og differensiere til ulike nivåer i klasserommet. Med tanke på noen av resultatene fra klassen (figur 3) kan man tenke seg at elever i neste steg for eksempel kan lage programmer som gir mangekanter ved innføring av løkker.

Noen elever så bort fra oppgavearket og lagde egne figurer, som en bil og et «opp-ned»-hus. Disse elevene var innom ulike geometriske figurer, for eksempel en sirkel til hjul til bilen, og trekant og kvadrat til et hus. De fikk dermed brukt tankene sine om geometriske figurer i ulike settinger til å innføre programmeringen. Disse elevene fikk jobbe videre med sine små prosjekter siden de implementerte programmering i disse prosjektene.

Tilbakemeldinger

I slutten av undervisningsøkten ble det sendt ut et spørreskjema til elevene med ulike påstander som de skulle gradere i fire kategorier fra enig til uenig. Spørreskjemaet inneholdt også to åpne spørsmål der elevene kunne svare med tekst for et mer utfyllende svar. Motivasjonen for å bruke et spørreskjema var å få en tilbakemelding på hvordan elevene oppfattet innføringen av tekstbasert programmering, og hvordan dette var relatert til matematikk.

Tabell 1 viser to av spørsmålene og svarene til elevene etter undervisningsøkten. Selv om introduksjonen til tekstbasert programmering for klassen var kort og bare inneholdt et eksempel på hvordan man skriver kode til et kvadrat, viser flertallet at de synes oppgavene ikke var for vanskelige å løse på egen hånd. Noen elever kommenterte til og med på spørreskjemaet at de ønsket vanskeligere og mer kompliserte oppgaver.

Kravik, R. (2020). Programmering og geometri. *Tangenten – tidsskrift for matematikkundervisning, 31*(2). 7–11.

www.caspar.no/tangenten

Spørsmål	Enig	Litt enig	Litt uenig	Uenig
Jeg synes oppgavene vi skulle løse på egen hånd, var for vanskelige	5	19	19	57
Det var lett å skrive feil i kodene når jeg programmerte	38	48	24	0

Tabell 1: Svar på spørreskjema fra elevene. Tallene er i prosent.

Vi ser også av tabellen at det er en overvekt i andelen elever som gir uttrykk for at det er fort å skrive feil kode. Elevene har erfaringer med digitale verktøy, som regneark og digital graftegner, noe som er implementert i undervisningen i grunnskolen. Disse digitale verktøyene har noen forskjeller og likhetstrekk med tekstbasert programmering. Elevene må være nøyaktige og skrive av riktig når de arbeider med tall, formler, områder, ligninger, funksjoner osv. i disse digitale verktøyene. Et annet likhetstrekk kan være at de får en umiddelbar (visuell) respons, som en figur i GeoGebra eller svar på oppgaver ved bruk av formler i regneark. En vesentlig forskjell mellom programmering og andre slike digitale verktøy er at programmering har en algoritmisk tankegang. Elevene må skaffe seg en oversikt over hva de skal gjøre, hvordan de vil gå frem, og hva de må ha med av koden sin. Kombinasjonen av svarene fra elevene er også interessant, ved at de synes oppgavene ikke er for vanskelige, men at det er lett å skrive feil.

Lignende undervisningsopplegg kan man gjøre ved å bruke programmering som et verktøy for å skape forståelse for matematikk i andre temaer. I sannsynlighet kan man bruke programmering som et verktøy til for eksempel å kaste terning. Dette gjør at man som lærer kan utvide antall kast av en fysisk terning, sammenlignet med å kaste terning i et klasserom. Ved å se på fordelinger av antall øyne på terningen ved terningkast kan det være lettere for elevene å se for seg hva vi mener med sannsynlighet. Kaste terning kan man også gjøre i regneark, men man får større muligheter til dette i programmering. I for eksempel programmeringsspråket Python kan man lage korte koder der man enkelt kan variere antallet terningkast. Det vil for eksempel være enkelt å endre antall kast til en million eller flere, mens dette vil være mer utfordrende i for eksempel regneark. Andre områder i matematikk for grunnskolen der man kan trekke inn programmering, kan for eksempel være øvelser i geometri, plotting av grafer til ulike funksjoner, ulike diagrammer i statistikk, løse ligninger, variabler i algebra, optimeringsoppgaver, for å nevne noe.

I undervisningsopplegget presentert her legger man opp til en balanse mellom geometri og programmering. Elevene bruker sine kunnskaper eller skaffer seg kunnskap om ulike geometriske figurer for å lage en avbildning av figuren ved hjelp av programmering. Det at programmering kan øke forståelsen for matematikk, gir elevene uttrykk for i etterkant av undervisningsøkten, tabell 2.

Utdanningsdirektoratet (2019) trekker frem ferdigheter knyttet til bruken av programmering for å utforske og løse matematiske problem. To elever har svart følgende i spørreunder-

Spørsmål	**Enig**	**Litt enig**	**Litt uenig**	**Uenig**
Jeg tror programmering kan gjøre at jeg forstår matematikk bedre	38	57	5	0
Jeg vil lære mer om programmering	67	29	5	5

Tabell 2: Svar på spørreskjema fra elevene. Tallene er i prosent.

Kravik, R. (2020). Programmering og geometri. *Tangenten – tidsskrift for matematikkundervisning, 31*(2). 7–11.

www.caspar.no/tangenten

søkelsen: «Du kan lære mer av matematikk» og «Morsomt å programmere, å gjøre noe nytt». I tillegg er majoriteten positiv til å lære mer om programmering basert på denne undervisningsøkten i tekstbasert programmering og geometri.

Sammendrag

Elevene gir også uttrykk for at de tror at programmering kan gjøre at de forstår matematikk bedre. Dette understreker at programmering kan være et verktøy for læring i matematikk. Elevene ga uttrykk for at oppgavene var overkommelige selv med en kort introduksjon, men samtidig ga de uttrykk for at det er lett å skrive feil når de skriver kode, og at de var positive til å lære mer om programmering.

Tanker i etterkant

Undervisningsopplegget ble gjennomført ved å bruke Python på nettstedet Trinket.io. Her kan man kjøre Python versjon 2.7 med datapakken «Turtle graphics» uten å installere programvaren. Ungdomsskolen er en IPAD-skole, og dette i kombinasjon med den nettbaserte Python-versjonen ga noen utfordringer knyttet til oppstart av undervisningsøkten. Bakgrunnen for å bruke denne «nettversjonen» av Python var som sagt at det var hensiktsmessig med hensyn til installasjon, og at det ikke var optimale gratisapper med aktuelle datapakker som kunne brukes til IPAD. Gjennomføring og spesielt oppstart av undervisningsopplegg om programmering i et klasserom vil kunne være enklere med datamaskin dersom programvaren er installert, eller dersom elevene har kjennskap til Trinket.io fra tidligere.

Undervisningsopplegget er gjennomført i en klasse på 21 elever, og det kan forventes at andre klasser kan ha ulike utfordringer. Det er jobbet med gjennomføring av et utvidet undervisningsopplegg i flere klasser på ulike ungdomsskoler og i grunnskolelærerutdanningen på Notodden.

Referanser

Kravik, R. (2019). *Undervisningsopplegg – Programmering og geometri.* Hentet 15.12.2019 fra https://web01.usn.no/~rek/annetmateriell/index.html

Nosrati, M. & Wæge, K. (2015). *Sentrale kjennetegn på god læring og undervisning i matematikk*. Hentet 15.11.2019 fra www.matematikksenteret.no

Wæge, K. (2007). *Elevenes motivasjon for å lære matematikk og undersøkende matematikkundervisning* (Doktoravhandling). NTNU, Trondheim.

Utdanningsdirektoratet (2006). *Læreplan i matematikk (LK06)*. Hentet fra https://www.udir.no/kl06/MAT1-04

Utdanningsdirektoratet (2019). *Læreplanverket i matematikk 1–10*. Hentet fra https://www.udir.no/lk20/mat01-05

Kravik, R. (2020). Programmering og geometri. *Tangenten – tidsskrift for matematikkundervisning, 31*(2). 7–11.

www.caspar.no/tangenten