Skolverket

Algebra, åk 4–6

Den här modulen fokuserar på reflektionens betydelse för hur ni i lärargruppen kan utveckla ett professionellt förhållningssätt till undervisning av algebra i årskurserna 4–6.

Modulen är uppdelad i åtta delar som behandlar ett antal didaktiska perspektiv inom matematikundervisning. Rubrikerna på delarna ger en kort sammanfattning av det didaktiska perspektiv som är huvudfokus i respektive del. Modulen har kompletterats med två texter som handlar om programmering, dessa finns i del 5 och del 7. Dessa texter och aktiviteter kan läsas och genomföras fristående från delens övriga innehåll.

Modulen består av följande delar:

1. Reflektion som lärprocess
2. Resonemangsförmåga
3. Bedömning för utveckling av undervisning i algebra
4. Interaktion i algebraklassrummet
5. Algebra och programmering som språk
6. Sociomatematiska normer
7. Kommunikation och programmering i algebraklassrummet
8. Algebraiskt tänkande

De didaktiska perspektiven som genomsyrar modulen kommer att användas tillsammans med två områden som är av betydelse för elevernas lärande i algebra, nämligen: mönster och talföljder samt likheter och olikheter. I modulen ges förslag på uppgifter och aktiviteter baserade på det algebraiska innehållet. Under arbetet med algebramodulen kan ni använda ”Mall – Planering av lektion” för att planera era lektioner” och ”Lektionsobservation – Protokoll” för att dokumentera genomförandet av era lektioner. I filmen ”Variationsteori” presenteras två viktiga begrepp som ligger till grund för innehållet i modulen. Dessa begrepp är kritiska aspekter och variationsmönster.

Ansvariga för modulen

Linnéuniversitetet, i samarbete med Göteborgs universitet, Blekinge Tekniska Högskola och Högskolan i Jönköping.

Del 5. Algebra och programmering som språk

Målet med den här delen är att visa det algebraiska språket som en viktig komponent för att skapa en bra interaktion i klassrummet. Genom att studera på vilket sätt algebra som språk uttrycks i klassrummet kommer ni att utveckla en förståelse för språkets olika ansikten i algebra. Det algebraiska innehållet i delen handlar om ekvationer.

Den här delen har kompletterats med en text och aktiviteter som har fokus på programmering. Ni får veta vad programmering innebär och bli bekanta med centrala begrepp inom programmering. Texten och aktiviteterna kan läsas och genomföras fristående från delens övriga innehåll.

Del 5: Moment A – individuell förberedelse

Denna del har kompletterats med material om programmering. Ni kan därför välja att antingen (I) läsa texten ”Algebra som språk” och genomföra tillhörande aktiviteter eller att (II) läsa texten ”Programmering – centrala begrepp” och genomföra tillhörande aktiviteter.

I. Läs

I texten ”Algebra som språk” behandlas tre komponenter som är viktiga att beakta vid analysen av elevernas kunskaper om det algebraiska symbolspråket.

Det algebraiska innehållet presenteras i texten ”Olika sätt att lösa ekvationer”. I bildspelet ”Ekvationsspelet” ges förslag på aktiviteter som kan vara ett stöd när ni utformar de frågor som behövs för att på ett systematiskt sätt få reda på elevernas kunskaper.

I. Lös ekvationer

Lös ekvationerna som finns i texten ”Olika sätt att lösa ekvationer” på sidan 1 och reflektera över hur du löser dessa ekvationer.

II. Läs

I texten ”Programmering – centrala begrepp” presenteras vad programmering kan innebära samt centrala begrepp i programmering. Ni får även förslag på olika aktiviteter som kan vara ett stöd när ni ska genomföra en aktivitet som har fokus på entydiga instruktioner.

Skolverket

Material

Algebra som språk
H. Lennerstad och C. Kilhamn

Olika sätt att lösa ekvationer
C. Kilhamn och L. Olteanu

Ekvationsspelet
Filformatet kan inte skrivas ut.
C. Kilhamn

Programmering – centrala begrepp
C. Olteanu och L. Olteanu

Matematik – Grundskola åk 4–6
Modul: Algebra
Del 5: Algebra och programmering som språk

Algebra som språk

Håkan Lennerstad, Blekinge tekniska högskola och Cecilia Kilhamn, Göteborgs universitet

Att ha ett språkligt perspektiv på algebra innebär att ta vara på de många likheter som faktiskt finns mellan algebrans språk och naturliga språk (språk som är modersmål för någon, t.ex. svenska). En förhoppning är att elever märker att de kan använda kunnandet i sitt modersmål för att ta till sig algebran.

I likhet med alla andra språk finns det i det algebraiska symbolspråket både syntax (språkregler) och semantik (betydelser). Algebrans syntax består av de räkneregler och konventioner som finns för tecknen, och man kan säga att algebrans semantik är som en bro mellan språket och världen. Liksom lärandet av ord i naturliga språk sker lärandet av algebrasymboler genom att man lär sig en definition eller ett synonym samt genom att se och använda dem många gånger. Vi kommer också senare att ta upp översättningar mellan algebrans språk och svenska för att synliggöra och lättare kunna diskutera algebrans betydelser.

Syntaxen avgör vilka algebraiska uttryck eller påståenden som är tolkningsbara och meningsfulla. Jämför följande symbolsammansättningar:

$b4c{=}{=}{+}/8$ (en meningslös sammansättning symboler)
$4b + 3$ (ett algebraiskt uttryck innehållande två operationer, två tal och en variabel)
$4b + 3 = x$ (ett påstående som är sant för vissa värden på b och x, men falskt för andra)
$a + b = b + a$ (kommutativa lagen för addition, som är sann för alla värden på a och b)

Alla språk är verktyg att tänka med, varav modersmålet är det kraftfullaste. Just för matematik är det algebraiska symbolspråket särskilt effektivt, under förutsättning att man behärskar det. Elevers svårigheter med det algebraiska symbolspråket kan dels handla om semantik, det vill säga vad de olika symbolerna betyder, och dels om syntax, dvs. vad man kan göra med symbolerna.

Från aritmetik till algebra

I aritmetik har elever blivit vana vid att **lösa** matematiska problem. I algebra ägnar man sig mer åt att **representera**, det vill säga att uttrycka och beskriva. Här är ett exempel:

- På en gren satt 13 kråkor. 9 kom till och 6 flög iväg. Hur många kråkor fanns det kvar på grenen? Frågan fokuserar på antalet, det vill säga svaret (som är 16).
- På en gren satt 13 kråkor. 9 kom till och 6 flög iväg. Representera situationen med matematiskt språk för att ta reda på hur många som fanns kvar. Uppgiften fokuserar på representation av den matematiska situationen (13 + 9 - 6).
- På en gren satt 13 kråkor. x kråkor kom till och y kråkor flög iväg. Skriv ett uttryck för hur många fåglar som fanns kvar. Fundera på vilka värden x och y kan ha. Uppgiften fokuserar på den algebraiska representationen och handlar om att använda algebra för att skriva ett generellt uttryck (13 + x – y).

För att komma till denna generalisering behöver ni fokusera på att först representera (dvs. uttrycka problemet med matematiskt symbolspråk) och sen räkna ut och svara. Elever behöver förstå symbolspråket på ett mer medvetet sätt. Ett steg på vägen är att uppfatta likhetstecknet som ”samma värde på båda sidor”, snarare än ”blir” eller ”därefter får jag”. Det betyder till exempel att ”3 + 8 = 11” och ”11 = 3 + 8” är lika giltiga påståenden.

Flera språk: naturligt språk, symbolspråk och fackspråk

Matematiska symboler och formler förekommer aldrig ensamma, utan alltid tillsammans med naturligt språk. Varje text på svenska med matematiska tecken kan därför sägas vara tvåspråkig, svenska blandat med matematikens symbolspråk. Magnus Österholm har visat att universitetsstudenter läser löpande text på ett annat sätt om det finns symboler i texten, vilket kan förväntas gälla minst lika mycket för elever. De tenderar att inrikta sig på symbolerna, och kan missa vad själva texten säger (Österholm, 2006). På samma sätt fokuserar elever ofta på siffrorna (som är de mest grundläggande av matematikens symboler) i en problemuppgift och kan bortse från problemets formulering. Matematiken har också ett eget fackspråk (matematikens terminologi) som delar ett stort antal ord med vardagsspråket, ibland med helt olika betydelser. Några omedelbara exempel på det är volym (ljudstyrka), udda (ovanlig), tal (föredrag, prat), bråk (handgemäng), dividera (diskutera), lösning (vattenblandning). Vi har alltså i matematiken ett **symbolspråk** med en egen uppsättning symboler (ett särskilt alfabet), och även **naturligt språk** och **fackspråk**, där de två sistnämnda ofta delar ord fast med olika betydelser.

Naturligt språk har både synonymer (lukt och doft betyder samma sak) och dubbelbetydelser (lag betyder både en samling personer som samarbetar och en förordning vi måste följa). Båda typerna finns också i symbolspråket och i matematikens fackspråk, och kan så klart vålla svårigheter. Exempelvis kan multiplikation skrivas med flera olika symboler: $\cdot$, $\times$, eller $*$ (vanligt i datorprogrammeringsspråk), eller med ingen symbol alls, som i uttrycket $2x$. I vissa länder skriver man $2\cdot3$ som $2(3)$. Minustecknet är en symbol med en dubbelbetydelse som kan orsaka svårigheter. Dels betyder det subtraktion (ex: 5 – 3: operator mellan två tal) och del används det för att ange att ett tal är negativt (ex: –3: egenskap hos ett tal). Vid ett uttryck som –3 – 4 bör det första minustecknet tolkas som en del av talet –3, och det andra som en subtraktion mellan talen –3 och 4.

Vad som kan göras med ett uttryck som –3 – 4 kan diskuteras på ett teckenmässigt sätt (Vad får man göra med minustecknen?) och på ett mer innehållsmässigt sätt (Är detta en subtraktion, är talen negativa eller positiva?).

Elever som klippte ut trianglar i papper lär ha sagt: ”En triangel har alltid två sidor, framsidan och baksidan!” Det är en bra illustration till att fackspråket inte alltid är konsistent. Man säger att en kub har åtta hörn, tolv kanter och sex sidor, där ”sida” ju betecknar en yta (som mäts med area). Samtidigt säger man att en kvadrat har fyra sidor. I två dimensioner (kvadrat, triangel) är sidan en sträcka, men i tre dimensioner (kub) används ”kant” för detta, och här är en ”sida” en yta.

Bokstäver har i vanligt språk endast ett ljudvärde, och får betydelse som en del av ett ord där ordet har betydelse och inte bokstaven. Detta är helt annorlunda i algebra där alla bokstäver kan användas, och där de då alltid har en numerisk betydelse. Bokstäver används som förkortningar av enheter (m, g, s, …) där de representerar ett visst bestämt mått, en kvantitet (längden på en meter, vikten av ett gram, …). Bokstäver förekommer också för speciella tal, till exempel (den grekiska bokstaven pi som är ett irrationellt tal) eller som prefix där bokstaven anger ett tals storlek (k som i kilo betyder tusen, m som i milli betyder tusendel …). Men den vanligaste användningen av bokstäver i algebra och matematik i allmänhet, är för att beteckna ett obekant tal eller storhet. Det är viktigt att eleverna förstår att det alltid finns ett tal bakom varje bokstav som förekommer i en matematisk beräkning. Bokstaven står för ett värde, den har en numerisk betydelse.

En annan stor skillnad mellan naturligt språk och symbolspråk är hur parenteser används. I det naturliga språket betecknar de något som kan utelämnas, medan de i algebran markerar vad som ska prioriteras, det vill säga beräknas först.

Att översätta mellan språken

Vid problemlösning formulerar man ofta ett matematiskt problem från praktiska exempel, och återvänder till det igen när problemet är löst. Detta brukar kallas modellering och tolkning.

Om Lasse är tre år äldre än Anna, kan detta översättas till formeln $L = A + 3$. Det är då viktigt att L och A är tydliga numeriska storheter. Lasse och Anna är inga numeriska storheter, det är däremot Lasses ålder och Annas ålder. När vi skapat formeln $L = A + 3$

har vi nått in i matematiken och har tillgång till alla dess verktyg och resonemang. Men L = A + 3 kan också översättas till naturligt språk som inte har med åldrar att göra, till exempel som ”ett tal som vi kallar L är alltid tre enheter större än ett annat tal, som vi kallar A”.

Man kan alltså tala om översättning mellan en praktisk situation och algebra, men också om översättning mellan algebra och naturligt språk, där någon praktisk situation inte finns med. Syftet med den första är att lösa ett praktiskt problem med hjälp av matematik. Syftet med den senare är att förstå matematiken lättare.

Matematiska symboler har aldrig särskilt långt till svenska (eller andra språk), och detta beror på att de alltid kan översättas. Det finns dessutom olika matematiska översättningar, som lyfter fram olika sidor av ett matematiskt samband. Här är exempel på olika översättningar av operationen 4 + 5 = 9:

- Fyra plus fem är lika med nio (läser ut vad symbolerna heter)
- Addera fyra och fem så får du nio (beskriver beräkningsproceduren)
- Summan av de två talen fyra och fem är lika mycket som nio (beskriver strukturen)

Översättningarna är exempel på retorisk matematik – matematik som skrivs utan symboler. Fram till 1600-talet var all skriven matematik retorisk. Det symboliska språket är alltså mindre än 400 år gammalt. Att förkorta naturligt språk med hjälp av symboler var en stor drivkraft bakom denna utveckling. Det innebar också andra vinster för matematiken: den blev mer internationell, mer oberoende av skilda tolkningar och speciella tillämpningar, och mer effektiv och lämplig för sitt särskilda syfte. Men ur elevers och studenters synvinkel tog den även längre avstånd från vardagsspråket, och fick större behov av förklaring och översättning. Men retoriska motsvarigheter finns alltid kvar i bakgrunden – det går alltid att översätta.

Ibland kan det vara bra att göra olika översättningar och att diskutera dessa. En generell beskrivning av den distributiva lagen, $(x + y)z = xz + yz$ *gäller alltid*, kan exempelvis översättas på följande något skilda sätt.

- Produkten av en summa av två tal och ett tredje tal har alltid samma värde som summan av produkterna av de två talen med det tredje.
- Om man har ett tal gånger en parentes som innehåller två tal, så kan man ta bort parentesen om man multiplicerar det första talet med de båda andra talen var för sig.

Den första översättningen är mer algebraisk genom att den beskriver produkter och summor, matematikens semantiska innehåll, och därmed handlar mer om förståelse av dessa begrepp. Den andra är mer teckenorienterad (och därmed mer syntaktisk), genom att den mer beskriver hur tecknen används och fungerar.

Genom att diskutera olika översättningar kan man få ett samtal både om de matematiska betydelserna, (operationerna, strukturen), och om hur tecknen används. En elev kan mycket väl förstå det ena utan att ha fått grepp om det andra. Elever bygger upp förståelse kring både semantik och syntax om de vänjer sig vid att diskutera sina egna översättningar.

En översättning är ofta betydligt klumpigare och otydligare än motsvarande formulering på symbolspråk. Detta är ett indirekt sätt att visa symbolspråkets effektivitet – när det väl behärskas. Fördelen med en översättning är att en översättning är ett sätt att *undersöka* ett matematiskt uttryck bit för bit, dess funktion granskas, genom att varje del översätts. På det sättet kan förståelse och intuition flytta över från den associationsrika svenskan till det vanligen associationsfattiga symbolspråket. Slutligen kan gemensamma översättningar öka möjligheterna till meningsfull dialog om matematik mellan lärare och elever (se mer i Ljungblad & Lennerstad, 2012).

Att formulera en algebraisk regel

Algebran är mer än ett symbolspråk, det är också ett speciellt sätt att resonera och argumentera, och ett verktyg för att uttrycka generaliseringar. För att det algebraiska symbolspråket ska bli meningsfullt för eleverna behöver de först ägna tid åt att formulera generella påståenden med hjälp av naturligt språk, och även argumentera för att dessa generella påståenden gäller.

Med naturligt språk kan det vara lättare för elever att formulera en algebraisk regel eftersom modersmålet generellt är det kraftfullaste tankeverktyget. Ett exempel på det kan vara att läraren ställer frågan: Kan man förklara varför 16+16=15+17 utan att behöva räkna ut vad varje sida blir? I ett exempel svarar Laura, åk 5 (Cai & Knuth, 2011, s. 54–55):

Laura: ”Ena talet bara skickar över ett till det andra talet. Ena 16 gav bort 1 och blev 15 och den andra tog den och blev 17”. Laura visar med ett nytt exempel: 8+8=7+9. Efter en stunds samtal formulerar hon följande generalisering: ”Det funkar inte bara för 1. Man kan ta bort vilket tal som helst bara jag också lägger till samma tal till det andra talet. Då ändras inte summan”. Här uttrycker Laura i naturligt språk och matematiskt fackspråk ett generellt algebraiskt samband som är användbart i många situationer. Så småningom kommer Laura förhoppningsvis att kunna uttrycka följande generella påstående: $a + a = (a–x) + (a+x)$ för alla a och x som är naturliga tal.

Hur lärs språk?

Modersmålet lär sig ett barn på ett spontant sätt genom lyssnande, imitation och anpassning, vi kan kalla det för att barnet tillägnar sig språket. I denna process lär sig barn använda grammatiken korrekt oftast utan att vara medvetna om de språkliga reglerna. Detta är en helt annan lärprocess än hur man senare kan lära sig ett språk med hjälp av en språkkurs. Det är mycket ovanligt att man lär sig ett nytt språk lika väl som ett som man tillägnat sig.

Vid ett naturligt samtal är det oftast inte språket som samtalet handlar om. Vi kan tala om vår frukost, en ishockeymatch, hur arbetsplatsen fungerar, en bok vi läst – språket är vårt medel till allt och används till stor del på ett oreflekterat sätt eftersom vi i sådana sammanhang då tänker på frukosten, ishockeymatchen, och så vidare. Ändå tränar vi samtidigt språket och utökar vår språkliga förmåga. Men språket är, utanför språkkurserna, naturligt oreflekterat. På liknande sätt är symbolspråket ofta oreflekterat, endast ett medel för matematiska resonemang, vilket kan vara ett problem vid elevers lärande eftersom det fungerar på ett annat sätt än naturliga språk. Detta är ännu ett skäl för att regelbundet göra översättningar. Som lärare kan man naturligtvis fokusera på de vanligaste svårigheterna i det ”nya” språket.

Den process barn använder när de lär sig sitt modersmål, genom att babbla, prata strunt, pröva, bli förstådda och missförstådda, har också sin plats för att finna säkerhet i det matematiska symbolspråket. Det gäller även innehållet: för att lära sig logik behöver man *pröva* och *testa* vilka resonemang som kan gälla, för att tydligt se var och varför de eventuellt spårar ur.

Användning av matematisk terminologi – viktigt men varsamt

Att använda en korrekt matematisk terminologi (fackspråk) är viktigt. Lika viktigt är det att symbolerna används på ett korrekt sätt. Både för att det är en del av den kunskap skolan syftar på, men också för att det framtida lärandet blir lättare. Dock kan det bli alltför mycket fokus på terminologin. Om det är svårt för elever att både förstå matematiken och terminologin, så kan stort fokus på terminologin leda till att den tar all uppmärksamhet, och att det inte blir något utrymme kvar till kreativitet och förståelse. Det matematiska innehållet kan då försvinna. Elever kan också bli mycket tysta och passiva om de är rädda att använda terminologin felaktigt. Därför är det viktigt att vara försiktig med hur man korrigerar terminologin. Det kan vara bra att lägga fokus på den matematiska tanken, och möjligtvis indirekt kommentera terminologianvändningen.

Som lärare kan man föregå med gott exempel genom att använda rätt terminologi, så eleverna ofta får höra och se den. Att sätta ord på saker är en viktig del av att förstå dem. Ska eleverna kunna föra matematiska resonemang behöver de ha matematiska fackord såsom dividera, subtrahera, udda, jämna, summa, differens, produkt, term, faktor, likhet, variabel, ekvation och så vidare. som en del av sitt aktiva ordförråd. Vid en lektion som handlar just om terminologi, snarare än om matematiska problem, kan man i högre grad kommentera terminologianvändningen. En elev som precis har lyckats lösa ett eller flera problem som denne finner svåra är ofta mer mottaglig för information om rätt terminologi.

Även en matematiker använder ofta den matematiska terminologin inkorrekt när ett forskningsproblem diskuteras, även om man förstås vet vad som är korrekt och är mycket noga med att skriva korrekt i sina rapporter. De flesta av oss bortser från stavfel och grammatiska fel när vi tar anteckningar på en föreläsning eller skriver en minneslapp, men är noga med det när vi skriver formella texter. Eleverna behöver få uppleva båda sorters språkanvändning i algebraklassrummet. Dels situationer där matematiken, de algebraiska resonemangen, är i fokus och terminologi hamnar i bakgrunden, och dels situationer då matematiska resonemang och slutsatser skrivs upp och redovisas på ett formellt korrekt sätt.

Referenser

Cai, J., & Knuth, E. (Red.). (2011). *Early algebraization. A global dialogue from multiple perspectives*. Springer.

Ljungblad, A.-L., & Lennerstad, H. (2012). *Matematik och respekt – matematikens mångfald och lyssnandets konst.* Liber Förlag.

Österholm, M. (2006). *Kognitiva och metakognitiva perspektiv på läsförståelse inom matematik*. [Doktorsavhandling, Linköpings universitet].

Olika sätt att lösa ekvationer

Cecilia Kilhamn, Göteborgs universitet och Lucian Olteanu, Linnéuniversitetet

Att lösa ekvationer är en central del av algebran, det är dess ”hantverk”. En ekvation kan ofta lösas på olika sätt och det är med hjälp av fantasi och kreativitet som du och dina elever kan hitta enkla och effektiva sätt att lösa ekvationer. En dator utför alltid en procedur på ett förprogrammerat sätt även om det är onödigt komplicerat. Det gör ingenting eftersom datorn är så snabb, och varken tappar lusten eller orken. Med människan är det tvärtom. Vi utnyttjar vår erfarenhet, intuition och kreativitet för att komma på smarta, enkla sätt som besparar både tid och arbete. Dessutom kan vi också komma på nya idéer och nya procedurer som utvecklar matematiken. Det gör aldrig datorn åt oss. Det här innehållet handlar om att fostra just denna insikt hos eleverna; insikten att en ekvation kan lösas på olika sätt, och att man kan utnyttja sin taluppfattning och sin förståelse av operationerna, sin förmåga att se mönster och sin fantasi som resurser. Målet är att elever utvecklar sin förmåga att välja lämpliga sätt att lösa en ekvation utifrån hur ekvationen ser ut, samt att de upplever ett behov av en formell och generell metod för ekvationer de finner svåra att lösa.

Först en kort definition

En ekvation är en likhet, så det är ett påstående. Ett påstående är sant eller falskt. En ekvation innehåller ofta en eller flera obekanta. Viss litteratur definierar ekvation som en likhet med obekanta, och betraktar därför inte en numerisk likhet såsom $4 + 5 = 9$ som en ekvation. Här har vi emellertid valt att hålla oss till definitionen i boken Matematiktermer för skolan där ekvation definieras som en matematisk utsaga som innehåller en likhet. Lösningen till en ekvation är de värden på de obekanta som gör ekvationen sann.

Om du har en ekvation med ett x ställer du alltså frågan: För vilka värden på x är denna likhet sann? Svaret på frågan är lösningen på ekvationen. Betydelsen av ordet lösning här skiljer sig alltså från betydelsen i problemlösning.

Några ekvationer

Innan du läser vidare ska du lösa följande ekvationer och reflektera över hur du löser dem. Löser du dem på olika sätt? Varför gör du inte på samma sätt? Vad är lika och vad är olika i ditt sätt att lösa dem?

a) $x + 3 = 10$
b) $2x + 3 = 13$

c) $5x + 3 = 1543$
d) $3{,}4x + 3 = 20$
e) $17 – 2x = 5x + 3$

Undervisningsdilemma

Undervisning om ekvationslösning medför ett dilemma som kan leda till att eleverna tappar både lusten och orken och finner det hela meningslöst. Dilemmat gäller vilka exempel du ska välja. Ska du välja så aritmetiskt enkla exempel att enbart ekvationslösningsmetoden träder fram, eller ska du välja exempel där aritmetiken är svår och metoden blir en nödvändig hjälp? Ekvation a) är sådan att eleven troligen direkt kan se att x måste vara 7. Ekvation b) kan lösas på flera olika sätt. Till exempel genom att **Gissa och pröva**, eller genom **Övertäckning**, det vill säga att eleven först tänker ”något plus 3 är lika med 13, då måste detta något vara lika med 10” och sedan ”2 gånger 5 är lika med 10 alltså är x lika med 5”. Om eleven enbart möter exempel av typ a) och b) men ändå tvingas skriva ut alla led i metoden Att göra lika på båda sidorna blir det väldigt omständligt och eleven får ingen möjlighet att uppleva metodens fördelar. Elevens tilltro till sin förmåga att se samband, använda sin taluppfattning och vara ”smart” nedvärderas till förmån för en träning i en procedur som är menad att underlätta för eleven längre fram i mötet med mer komplicerade ekvationer. Många forskare och elever vittnar om att lärare ofta väljer så lätta uppgifter att eleven inte ser poängen med att lära sig en specifik metod. Man löser ekvationen med en metod som läraren/läroboken bestämt istället för att använda en metod som är effektiv.

Arbeta från aritmetiken

Ett sätt att inleda arbetet med ekvationslösning beskrivet av Herscovics och Kieran (1980) är att starta med tal och ersätta talen med bokstäver så att ekvationer skapas. Eleverna i deras undersökning fick börja med en aritmetisk identitet: $3 \cdot 7 + 3 = 25 - 1$. Sedan ”gömdes” ett av talen i likheten bakom en bokstav och på så vis skapades olika ekvationer:

$a \cdot 7 + 3 = 25 - 1$
$3 \cdot b + 3 = 25 - 1$
$3 \cdot 7 + 3 = c - 1$
$x \cdot 7 + x = 25 - 1$
$p \cdot 7 + 3 = m - 1$

Genom att skapa variationer av samma ekvation kunde läraren diskutera med eleverna att bokstaven stod för ett tal, att man kunde använda vilken bokstav man ville, och att om samma bokstav förekom flera gånger i en ekvation gömde den samma tal.

Helt spontant väcktes elevernas medvetenhet om lösningen som den inverterade processen till skapandet av ekvationen: att finna vilket tal som gömmer sig bakom bokstaven. Även ekvationer med mer än en lösning dök upp i elevernas arbete. En elev valde att gömma 3:orna i likheten 2 + 3 = 3 + 2 och fick ekvationen 2 + a = a + 2. I diskussionen som följde upptäckte eleverna att likheten blev sann oavsett vilket tal man ersatte a med.

I nästa steg fick eleverna skapa ekvationer direkt och sedan försöka lösa dem. De upptäckte då att alla ekvationer inte har en lösning, till exempel 2x + 3 = 2x + 4. Olika ekvationer har olika egenskaper och kan upplevas som lättare eller svårare av olika skäl. Här är några exempel på olika typer av ekvationer:

$x + 2 = 12$	en enkel ekvation med en lösning
$30 – x = 24+1$	en ekvation med uttryck i båda leden och med en lösning
$2x – 8 = 30$	en ekvation med flera räknesätt i ena ledet och med en lösning
$x + 2=10 – x$	en ekvation med samma obekanta tal på två ställen och med en lösning
$25 = a \cdot b$	en ekvation med två obekanta tal och med många lösningar
$2 + s = s + 2$	en ekvation med oändligt många lösningar

Informella metoder för ekvationslösning

När man inleder undervisning om ekvationslösning är det vanligt att man först använder informella lösningsmetoder. De kallas informella eftersom de inte är generella. Många av de informella metoderna bygger på en god taluppfattning och en förståelse för inversa operationer (omvända räknesätt). Detta gäller till exempel **Baklängesräkning** eller **Övertäckning**. Om eleverna i de tidigare skolåren arbetat med hur räkneoperationerna (räknesätten) hänger samman, så är det enkelt att utnyttja den kunskapen även i algebran. Till exempel kan eleverna se samma struktur i dessa ekvivalenser. Ekvivalenspilen ⇔ visar att två ekvationer är sanna samtidigt. Om den ena gäller så gäller den andra och vice versa.

$8 + 3 = 11 \Leftrightarrow 8 = 11 – 3$ $\qquad$ $4 \cdot 3 = 12 \Leftrightarrow 3 = 12/4$

$x + 3 = 11 \Leftrightarrow x = 11 – 3$ $\qquad$ $4 \cdot x = 12 \Leftrightarrow x = 12/4$

Metoden **Gissa och Pröva** är en metod du inte bör förakta. Ju bättre taluppfattning och ju mer erfarenhet man har desto bättre gissningar kan man göra. En matematiker arbetar hela tiden med att gissa, men man gissar ju inte blint. Att gissa är att tänka ”det skulle kunna vara”, och då använder man sin fantasi. Undervisningen borde på alla sätt stärka eleverna att använda sin fantasi och komma med välgrundade gissningar. En viktig följdfråga till en gissning är: varför tror du det? En sådan fråga uppmanar eleven att föra ett resonemang och hitta motiveringar för sin gissning. Genom gissningar kan eleverna snabbt få en bra uppfattning om ungefär vad svaret ska bli, och kan på slutet använda det för att bedöma rimligheten i svaret.

Att göra lika i båda leden är en formell metod och det är bra om man får eleverna att se den som en metod bland andra. Fördelen med den metoden är att den är generell och alltid går att använda. Det är en stor vinst om man lyckas få eleverna att efterfråga en sådan metod och se fördelarna med den även om den ibland kan tyckas omständlig. Det finns inte heller en speciell uppsättning informella metoder som eleverna ska lära sig, det viktiga är att de lär sig föra resonemang som hjälper dem att lösa ekvationer, att de fokuserar algebraiskt tänkande snarare än bestämda procedurer.

Tänkbara kritiska aspekter

För att elever ska lära sig att urskilja olika sätt att lösa ekvationer finns det vissa aspekter som är kritiska att de urskiljer. Vilka dessa är beror självfallet på eleverna, hur de tänker och vad de kan sedan förut. Här finns några delar av innehållet beskrivna som tänkbara kritiska aspekter.

1. Att förstå att en ekvation är en likhet som innehåller obekanta.
2. Att förstå ett en bokstav i en ekvation alltid representerar ett tal (ett värde). I arbetet med ekvationslösning kan du göra detta extra tydligt genom ditt ordval, att tala om bokstaven som ett obekant tal.
3. Att förstå att lösningen till en ekvation med en eller flera obekanta är de värden som gör ekvationen (likheten) sann, och att det finns ekvationer som kan ha inga, en eller flera lösningar.

Innehållet som tas upp i del två och tre kan bli synliga om du sätter upp en rad ekvationer och sedan diskuterar om de är sanna eller inte för olika värden på den obekanta. Om delarna blir synliga eller ej beror på vilka frågor du ställer. Bra frågor kan vara:

> Är det här värdet på den obekanta en lösning till ekvationen? Hur vet du det?
>
> Kan det finnas fler lösningar till ekvationen?
>
> Hur många lösningar kan det finnas?

4. Att förstå att de kunskaper eleven har med sig från aritmetiken kan utnyttjas vid ekvationslösning, exempelvis talfakta och inversa operationer. Med begreppet talfakta menas här sådan kunskap om talen som eleverna lärt sig som automatiserad fakta, exempelvis tiokamraterna, multiplikationstabellerna och så vidare. Inversa operationer avser motsatta räkneoperationer eller omvända räknesätt. Addition och subtraktion är inversa operationer, liksom multiplikation och division. I denna del

ingår också att se värdet av välgrundade gissningar och behovet av att pröva och revidera sina gissningar. Ju mer eleven kan utnyttja sina aritmetiska kunskaper desto bättre blir eleven på att gissa. Att gissa och pröva kan också föra in ett moment av systematik i resonemanget. Jag gissar, prövar och gissar igen, allteftersom kommer mina gissningar allt närmare lösningen.

Vid undervisning om ekvationslösning vill vi få fram så många olika sätt att lösa ekvationen på som möjligt för att sedan kunna jämföra strategier och diskutera likheter och olikheter samt vilken kunskap eleverna utnyttjar. Vissa elever kanske inte kan lösa ekvationen eller fastnar på halva vägen. För dessa elever är det speciellt viktigt att prata om hur de kan utnyttja den kunskap de har. Det är inte ekvationens lösning i sig som ska stå i centrum för diskussionen utan strategier för att ta reda på lösningen. Därför är det viktigt att du väljer ekvationerna med omsorg för att få fram den variation av strategier du önskar. Du har också ett ansvar att själv bidra med strategier om de inte kommer från eleverna. Om du tittar på de fem ekvationerna du löste inledningsvis kan du se att de initierar olika strategier.

a) $x + 3 = 10$

Här är det troligt att eleverna använder talfakta (tiokamraterna) eller kunskap om inversa operationer (7 + 3 = 10 eftersom 10 – 3 = 7)

b) $2x + 3 = 13$

Här kanske några elever tänker i två steg med så kallad **Övertäckning**: först med inversa operationen för att veta att $2x = 10$, och sedan med hjälp av talfakta. Andra elever kanske gissar och prövar.

c) $5x + 3 = 1543$

Här har eleverna inte längre hjälp av sina talfakta utan tvingas söka nya strategier. Kanske ser de likheten med föregående ekvation och inser att 5x = 1540. I så fall använder de en strategi som brukar kallas för att **Lösa en enklare ekvation**. Ett samtal om att man subtraherat 3 i båda leden kan här bli grogrund för den formella metoden att **Göra lika i båda leden**. Vad är sedan en välgrundad gissning om man vill veta vilket tal som ska multipliceras med 5 för att vara lika med 1540? Här är taluppfattning i det större talområdet en viktig resurs, eller kunskapen att division är inversen till multiplikation.

d) $3{,}4x + 3 = 20$

Här utökas talområdet ytterligare och elevens talfakta förslår inte. När man kommit så långt som till att $3{,}4x = 17$ är det ändå inte säkert att eleven kan se lösningen. En välgrundad gissning kan vara ett sätt. Någon kanske kommer på att man kan använda inversa operationen och beräknar 17/3,4 på miniräknaren.

e) $17 - 2x = 5x + 3$

Här blir svårigheterna mycket större och den vanligaste strategin blir nog att **Gissa och pröva,** och eleven får ett argument för att han eller hon behöver en mer pålitlig metod såsom den formella metoden att **Göra lika i båda leden**.

5. Att se fördelar och nackdelar med olika metoder. Om eleven urskiljer denna del fullt ut inser eleven också fördelarna med en generell metod som han eller hon alltid kan lita på. Den här delen är på metanivå eftersom det är först när eleverna fått syn på en variation av olika metoder som de kan börja diskutera metodernas fördelar och nackdelar.

Referenser

Kiselman, C., & Mouwitz, L. (2008). *Matematiktermer för skolan*. Nationellt centrum för matematikutbildning.

Herscovics, N., & Kieran, C. (1980). Constructing meaning for the concept of equation. *The Mathematics Teacher*, *73*, 572–589.

Matematik – Grundskola åk 4–6
Modul: Algebra
Del 5: Algebra och programmering som språk

Programmering – centrala begrepp

Constanta Olteanu och Lucian Olteanu, Linnéuniversitetet

Att arbeta med programmering i matematik gör det möjligt för eleverna att pröva sig fram hur de kan använda entydiga stegvisa instruktioner för att lösa ett problem snarare än att hitta ”det rätta svaret”. På grundskolenivå handlar det om ett sätt att tänka hos eleverna och att arbeta så att eleverna till exempel får arbeta med algoritmer, tänka logiskt, bryta ned problem i mindre delar, söka och korrigera fel samt tolka resultat.

Programmeringsprocessen

Programmering kan beskrivas som en process för att utforma och skriva en uppsättning instruktioner (ett program) för en dator på ett språk som den kan förstå. Programmeringsprocessen kan exempelvis inkludera följande steg:

- analysera och förstå problemet
- dela upp problemet i mindre delar (bryta ned problemet)
- skissa en lösning eller flera lösningar
- välja en lösning
- hitta återkommande mönster och utnyttja dessa
- skapa en algoritm
- förbättra en algoritm
- skriva instruktioner i naturligt språk, en programmeringsmiljö eller i ett programspråk (”koda”)
- göra felsökningar
- tolka resultat

Programmeringsprocessen är inte linjär, utan man ofta får gå tillbaka till tidigare steg flera gånger. Det första steget i programmeringsprocessen är att utveckla elevernas förmåga att konstruera, beskriva och följa entydiga stegvisa instruktioner. Detta kan göras utan att använda en dator. Det är viktigt att höja svårighetsgraden och komplexiteten gradvis.

Centrala begrepp inom programmering

Instruktioner

Instruktioner utgör grunden i ett program och används för att manipulera data, utföra beräkningar, skriva ut data på skärmen, och så vidare. Det finns tre huvudsakliga typer av instruktioner.

Sekvens – en följd av instruktioner som utförs en i taget i den ordning de skrivits

Alternativ eller villkorssats – används för att ge datorn möjlighet att välja mellan olika instruktioner beroende på den situation som gäller för tillfället

Villkorssatserna skrivs på datorn som, *om-så* eller *om-så-annars* (på engelska if och if-else)

Exempel:

> *Om* du har svarta byxor *så* ska du gå två steg framåt, *annars* ska du snurra runt ett varv.

Repetition eller slingor/loopar – upprepar en del i programmet ett givet antal gånger eller tills ett givet villkor har uppfyllts

Exempel:

> *Upprepa* fyra gånger. Gå två steg framåt och hoppa en gång.
>
> *Repetera* tills du har nått fram till målet: Gå två steg framåt.

Algoritmer

En algoritm är en uppsättning instruktioner som skapas för att till exempel lösa ett problem eller förverkliga en idé och beskriver exakt hur man ska gå tillväga för att lösa problemet. Beskrivningen görs steg-för-steg.

Algoritmer finns överallt i vardagslivet, exempelvis ett matlagningsrecept eller en monteringsanvisning, men används även i vardagliga situationer som när man borstar tänderna eller klär på sig. De kan förekomma i klassrummet, till exempel en lektionsplan eller ett schema. I klassrummet kan man öva på stegvisa entydiga instruktioner genom att en elev ger stegvisa instruktioner till en annan elev för att förflytta sig från en punkt till en annan punkt. Dessa stegvisa instruktioner formar tillsammans en algoritm och bygger på ett systematiskt tillvägagångssätt. Det finns algoritmer som upprepas gång på gång, för alltid, eller tills något annat händer.

Det finns likheter och skillnader mellan användningen av algoritmer i algebra och i programmering. En likhet är att en algoritm består av en följd av instruktioner, operationer eller regler, som beskriver hur en viss uppgift ska lösas (t.ex. en additionsalgoritm, hur vi ställer upp räkneoperationer). En annan likhet är att algoritmer är abstrakta och behöver visualiseras eller konkretiseras för att kunna diskuteras. I algebra kan visualiseringen göras genom att använda naturligt språk, grafer eller tabeller och i programmering genom att använda blockdiagram eller ett programspråk i till exempel visuella programmeringsmiljöer.

Inom matematiken är en algoritm en procedur som utförs i bestämda steg i en viss ordning och används för att lösa ett givet problem eller utföra en beräkning. Algoritmer inom matematiken kännetecknas av att de består av en beräkningsmetod som alltid utförs på samma sätt oavsett vilka tal som ingår i beräkningen. Skillnaden är att i programmering består en algoritm av stegvisa instruktioner som kan formuleras på olika sätt och till ett visst problem kan finnas flera algoritmer med olika egenskaper. Ett exempel är om man ska gå till ett bestämt mål så kan man gå olika vägar med olika antal steg.

För att kunna stödja eleverna att förstå strukturen i en algoritm är det av betydelse att du som lärare vet att en algoritm måste uppfylla följande villkor:

- ska ha indata – en algoritm har en startpunkt och instruktioner som talar om vad som ska utföras
- ska ge utdata – en algoritm ger ett resultat, men olika algoritmer kan ge samma resultat
- ska vara ändlig – varje algoritm måste avslutas efter ett ändligt antal steg
- ska vara definierad – varje steg i algoritmen ska vara precist definierat

Variabel

Du känner redan till begreppet variabler, som bland annat används i algebra. Då talar man ofta om variabler som x, y, z och så vidare, som används för att representera en kvantitet i ett matematiskt uttryck. Variabler i programmering används lite annorlunda än i algebra. I programmering har variabler ett symboliskt namn som är associerat med ett värde och vars associerade värde kan ändras. I programmering kan vi exempelvis se variabler som olika burkar, där vi kan stoppa in olika värden. Varje burk har ett namn, som är variabelns namn. I vårt exempel är variabelnamnen ”bilar”, ”frukter” och ”spelare”. Lägg märke till att variabelnamnen är valda så att man omedelbart kan förstå vad det står för.

Figur 1

Variabler i programmering

Det är viktigt att förstå att innehållet i varje burk varierar när man kör ett program, vilket inte är fallet med variabler i matematik, där variabeln x i en funktion inte varierar när man genomför olika beräkningar. Beroende på variabelns typ kan man lagra olika värden, såsom till exempel heltal, decimaltal eller text. Låt oss anta att våra variabler bilar, frukter och spelare ska representera antal och därmed lagrar heltal. Man säger att man tilldelar en variabel ett värde. Hur detta ser ut i praktiken beror på vilken programmeringsmiljö man använder eller på vilket programspråk man jobbar i.

Likhetstecknets användning

Till skillnad från användningen av likhetstecknet i algebra används detta tecken i programmering med en annan innebörd, nämligen för att föra över innehållet från högersidan till variabeln på vänstersidan, det vill säga att högerledets uttryck beräknas först och därefter får vänsterledets variabel detta värde. Till exempel skriver man *frukter* = 5 för att lagra värdet 5 i variabeln frukter, medan *frukter* = *frukter* + 4 innebär att man lägger till fyra i burken frukter, det vill säga man ersätter det ursprungliga värdet med det nya värdet.

Felsökning

Programmeringsprocessen avslutas med att eleverna tolkar resultatet och säkerställer att indata ger den utdata som var tänkt, det vill säga lösning på problemet. Om så inte är fallet måste en felsökning ske. Felsökning innebär att identifiera och åtgärda eventuella fel. På engelska kallas det *debugging*. För att kunna göra felsökning behöver eleverna uppmärksamma helheten, detaljer i helheten och relationer mellan detaljer, men även egenskaper hos begrepp som avser både algebra och programmering. Felsökning är en viktig del i programmeringsprocessen och eleverna bör därför bli medvetna om detta. Det behövs uthållighet för att hitta fel i programmet och åtgärda dessa.

För att kunna felsöka en instruktion behöver eleverna i första hand uppmärksamma olika fel. Det finns flera olika typer av fel, men vi koncentrerar oss på de slags fel som kan uppträda när eleverna ger instruktioner till varandra utan att använda dator. Några möjliga fel kan vara:

- i vilken ordning olika instruktioner ges eller genomförs
- instruktioner är inte tillräckligt tydliga

- instruktioner saknas i vissa steg
- instruktioner används inte stegvis, det vill säga en för stor mängd information på en gång

Dessutom det kan finnas modellfel när en modell av verkligheten inte stämmer överens med verkligheten. Modellfelet kan bero på att man har gjort orimliga instruktioner eller att man missförstått själva problemet.

Om man använder en dator, så kan det uppträda även andra fel. De vanligaste felen kan vara syntaxfel (ett grammatiskt fel som bryter mot de lagar som ett programmeringsspråk har), exekveringsfel (uppkommer under körning och medför att programmet inte kan utföras), logiskt fel (genererar fel svar eller svar som blir fel bara för en viss sorts indata).

Ett problem och därmed även den algoritm som löser det kan brytas ned i olika delar. I nedbrytningsprocessen kan eleverna förstå, lösa, utveckla och felsöka delarna separat. Beroende på vilket slags fel det är fråga om kan du använda olika metoder för att leta efter och upptäcka fel. Om eleverna ger eller genomför instruktioner utan att använda en dator kan du exempelvis:

- förklara instruktionerna för någon annan
- kontrollera instruktionerna kritiskt
- göra en lista med möjliga fel
- anteckna vad du redan har testat

Om eleverna använder en dator kan du eller eleverna testa programmet systematiskt med olika indata för att se om resultatet blir rätt. Felsökning är ett utmärkt tillfälle för eleverna att lära sig av sina misstag och att bli bättre på programmering.

Andra begrepp

Andra begrepp som används inom programmering är exempelvis

- kod – en sekvens av instruktioner i ett programspråk som datorn kan tolka och utföra
- programspråk – de språk som man använder för att skriva koden (ex. JavaScript, Python)
- satser – instruktioner till datorn för att utföra något (ibland används ordet kommando)
- script – en sekvens av satser
- syntax – språkets grammatik, anger hur instruktioner i språket får bildas

- köra/exekvera – att utföra instruktionerna i ett program

Tinkering

I programmeringsprocessen är det viktigt att förklara och klargöra idéer med exempel och visualiseringar samt att skapa en lärandemiljö som gör det möjligt för eleverna att lära sig programmering (Thota, 2014). Ett flertal forskare hävdar att en lärandemiljö som tillåter eleverna att testa saker och idéer i en ostrukturerad process, *tinkering,* gynnar lärandet av programmering.

Inom didaktisk forskning presenteras två olika sidor av tinkeringprocessen, dels den fria aktiviteten där elever skapar eller ändrar ett befintligt datorprogram och dels aktiviteter där eleverna får testa, göra fel, samt ge och få feedback från andra elever, lärare eller dator. Det finns forskare som gör en distinktion mellan tinkerers och planerare (Blikstein, m.fl., 2014). Tinkerers testar och gör en serie stegvisa förändringar i sina program för att skapa en färdig lösning. Planerare däremot är mer systematiska, de tar fram och genomför en handlingsplan för att kunna göra slutliga ändringar i skapandet av en algoritm eller ett program. Att låta eleverna utvecklas som både tinkerers och planerare skapar möjligheter för dem att förstå olika faser i programmeringsprocessen. Användningen av visuella programmeringsmiljöer kan stödja både tinkerers och planerare eftersom miljön erbjuder elever möjligheter att både testa sig fram och gå systematiskt tillväga. Detta spelar en central roll i vad eleverna har möjlighet att förstå när programmering används i matematikundervisningen.

Tinkering tillåter att idéer kolliderar (dvs. att en eller flera kritiska aspekter medverkar samtidigt) och det är just i dessa kollisionspunkter som kreativitet uppstår. Lärare ger eleverna självtillit genom att de tillåts experimentera, ta risker och utforska egna idéer (Martinez & Stager, 2013). Detta i sin tur leder till att de börjar se sig själva som elever som har bra idéer och kan förvandla sina idéer till verklighet. Detta kan bidra till att eleverna utvecklar färdigheter som kan användas senare i programmering. Dessa färdigheter är centrala dimensioner för att lära sig programmera och inkluderar:

- Vana vid att hantera komplexitet.
- Uthållighet vid arbete med svåra eller stora problem.
- Tolerans för tvetydighet eller osäkerhet.
- Förmåga att hantera öppna problemuppgifter.
- Förmåga att kommunicera och samarbeta med andra för att komma fram till en gemensam lösning.

Referenser

Blikstein, P., Worsley, M., Piech, C., Sahami, M., Cooper, S., & Koller, D. (2014). Programming pluralism: Using learning analytics to detect patterns in the learning of computer programming. *Journal of the Learning Sciences*, *23*(4), 561–599.

Martinez, S. L., & G. Stager. (2013). *Invent to learn: Making, tinkering, and engineering in the classroom*. Constructing Modern Knowledge Press.

Thota, N. (2014). Programming course design: Phenomenographic approach to learning and teaching. I *Proc. 2nd International Conference on Learning and Teaching in Computing and Engineering* (s. 125–132). IEEE Computer Society.

Förslag på aktiviteter

Nedan finns förslag på aktiviteter som hjälper dig komma igång med grundläggande programmering med eleverna. Inled med att berätta om att en del av programmering är att lära sig läsa och felsöka sin egen och andras kod samt att skapa entydiga instruktioner. Precis som i olika matematikaktiviteter kan eleverna, även i programmering, behöva möta flera liknande övningar för att de ska ha möjlighet att förstå själva huvuduppgiften.

Inled en lektion med en gemensam övning. Lägg en penna och ett papper på ett bord. Eleverna ska nu ge dig instruktioner att ta upp pennan från bordet och lägga tillbaka den på bordet igen. Med denna övning kan du göra eleverna medvetna om vikten av tydlighet och vad eleverna bör tänka på när de ger instruktioner för att skapa en algoritm. När du har skapat en gemensam förståelse för instruktionernas betydelse vid programmering kan du genomföra en eller flera av nedanstående aktiviteter.

Aktivitet 1 – Algoritm: felsökning

Du kan välja att genomföra variant 1 eller 2. Skillnaden är att i variant 1 ger eleverna instruktioner medan i variant 2 följer eleverna de instruktioner som ges.

Variant 1

Syfte: Att utveckla elevernas förståelse för ordningen i vilken olika instruktioner ges, att instruktionerna behöver vara tydliga, att man inte kan ge en för stor mängd information på en gång samt behovet av att korrigera instruktionerna om vissa steg saknas eller är felaktiga.

Utrustning: Valfritt föremål, papper och penna

Instruktioner till elever: Skriv ner en algoritm som ska vara så tydlig att du ska kunna följa den för att ta upp ett föremål från golvet och ställa det på ett bord.

Genomförande: Ställ ett föremål på golvet. Dela in eleverna i par och låt dem skriva ner en algoritm som ska vara så tydlig att du ska kunna följa den för att ta upp föremålet från golvet och ställa det på ett bord. Be eleverna ange hur säkra de tror de är på att deras algoritm fungerar och välj några algoritmer som:

a) gör det möjligt för dig att ta upp föremålet från golvet och ställa det på ett bord;

b) inte gör det möjligt att ta upp föremålet från golvet och ställa det på ett bord.

Utför instruktionerna och felsök därefter tillsammans med eleverna de instruktioner som inte fungerade. Diskutera även vad som ska ändras och varför. I detta sammanhang kan du även lyfta fram att en given algoritm ger ett givet resultat, men att olika algoritmer kan ge samma resultat.

Variant 2

Syfte: Att utveckla elevernas förståelse för ordningen i vilken olika instruktioner ges, att instruktionerna behöver vara tydliga, att man inte kan ge en för stor mängd information på en gång samt behovet av att korrigera instruktionerna om vissa steg saknas eller är felaktiga.

Utrustning: Valfritt föremål, kopierade instruktioner

Instruktioner till elever: Ställ ett föremål på golvet. Undersök vad det är som händer när man använder följande algoritm för att ta upp föremålet från golvet.

- Greppa tag om föremålet.
- Lyft föremålet.
- Ställ föremålet på bordet.

Diskutera: Vilka delar av dessa instruktioner kan en dator inte förstå? Hur kan en fungerande algoritm se ut istället?

Genomförande: Ställ ett föremål på golvet. Låt eleverna arbeta i par och undersöka vad det är som händer när man använder en algoritm för att ta upp ett föremål från

Låt eleverna diskutera vilka delar av dessa instruktioner som en dator inte skulle förstå. Exempel: vilket föremål, vad det innebär att greppa tag om, hur man lyfter, hur högt, vilket bord. Låt eleverna diskutera:

- Kommer algoritmen att fungera? Om inte, vad var det som blev fel?
- Hur kan en fungerande algoritm se ut istället?

Gå slutligen igenom ett eller flera av felen tillsammans. Du kan ställa följande frågor:

- Vad var problemet?
- Hur hittade ni det?
- Hur löste ni det?
- Har någon annan löst problemet på ett annat sätt?
- Var det något som var förvirrande?
- Vilka fel hittade ni och vilka åtgärder har ni tagit?

Alternativ: Utifrån elevernas förkunskaper och erfarenheter kan du variera instruktionerna så att det ska finnas flera steg eller innehålla flera "fel" (t.ex. att instruktionerna kommer i fel ordning, någon instruktion är otydlig, något moment saknas).

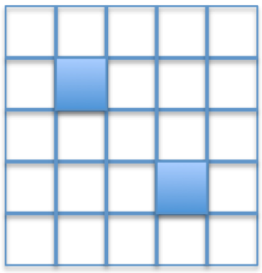
Aktivitet 2 – Programmering med papper och penna

Syfte: Att utveckla elevens förmåga att ge stegvisa instruktioner.

Utrustning: Penna, kopior på kvadrater som består av 5 x 5 rutor

Instruktioner till elever: En av er är programmerare och den andre utför programmet. Programmeraren markerar två rutor i sin kvadrat utan att visa ”utföraren”. Nu ska programmaren beskriva var rutorna är placerade i kvadraten genom att ge stegvisa instruktioner. När ni är klara ska ni jämföra resultatet och byta roller.

Genomförande: Dela in eleverna i par där elev A ska vara programmerare och ge sin kamrat instruktioner. Elev B ska genomföra instruktionerna som programmeraren ger. Gör uppgiften två gånger så att båda elever får prova på att vara programmerare. Elev A markerar två rutor utan att visa elev B. Varje ruta betraktas som en pixel. Dessa rutor kan exempelvis placeras på följande sätt:

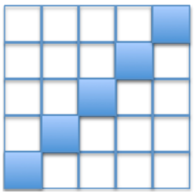
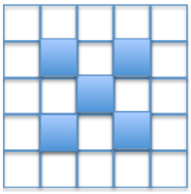
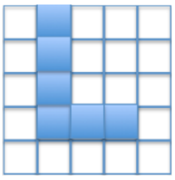


Elev B ska markera samma rutor utifrån A:s instruktioner. Uppmuntra eleverna att beskriva rutornas positioner med egna ord även om de kanske inte har det exakta vokabuläret. När båda är överens jämför eleverna sina inlagda rutor. Byt sedan roller.

Den här övningen fungerar mycket väl för alla åldrar och färdighetsnivåer och kan varieras i det oändliga. Man kan även göra den tredimensionellt med hjälp av exempelvis lego eller multilink. Det blir en riktig utmaning när eleverna får abstrakta bilder att jobba med. Då sätts förmågan att beskriva och ge stegvisa instruktioner verkligen på prov. Övningen kan enkelt varieras och ges i stigande svårighetsgrad. Nedan presenteras tre alternativ.

Alternativ 1

Elev A markerar flera rutor som bildar ett mönster. Exempelvis:

 eller  eller 

A ger instruktioner om var rutorna ligger utan att se på vilka rutor B markerar. B får ställa frågor och fråga fler gånger. A repeterar informationen tills B är nöjd.

Alternativ 2

Elev A lägger in flera rutor som bildar ett mönster. A ger information men endast då B ställt en fråga. B måste ta initiativ och vara aktiv annars får han eller hon ingen information.

Alternativ 3

Elev A markerar flera rutor som bildar ett mönster. A agerar sändare och ger instruktioner utan att interagera med B. I det här alternativet har B ingen möjlighet att ställa frågor eller reagera med kroppsspråk eller ljud för att visa att han eller hon inte förstått.

Oavsett vilket alternativ du väljer eller om du skapar nya mönster än vad som föreslås i denna text är det viktigt att samtala med eleverna:

- På vilka olika sätt kan de stegvisa instruktionerna beskrivas?
- Vilka instruktioner var svåra att ge?
- Vilka instruktioner fungerade inte?

Del 5: Moment B – kollegialt arbete

I. Diskutera

Utgå från era reflektioner från de texter som ni läste samt era erfarenheter från intervjuerna som ni gjorde i del 3. Fokusera på följande:

- Hur kan ni utveckla den syntaktiska medvetenheten hos eleverna?
- Är det den symboliska, den syntaktiska eller den tvetydiga medvetenheten som eleverna behöver utveckla i arbetet med ekvationer?
- Vilken aktivitet väljer ni att genomföra i era klasser för att på bästa sätt få syn på vilket sätt eleverna använder det matematiska språket och matematisk terminologi?

I. Förbered en aktivitet

Välj en aktivitet att genomföra utifrån de förslag som ges i bildspelet ” Ekvationsspelet”. Utgå från den valda aktiviteten och utforma frågor som underlag för att kunna identifiera vad som är kritiskt i elevens lärande. Om ni undervisar elever i olika årskurser, kan ni behålla samma innehåll men variera talområden utifrån elevgruppen. Ni kan använda punkt 9 i ”Lektionsobservation – Protokoll” för att dokumentera genomförandet av aktiviteten.

II. Diskutera

Utgå från era reflektioner i texten ”Programmering – centrala begrepp” samt era erfarenheter av programmering.

- Vilka skillnader och likheter finns mellan användningen av algoritmer i matematik och i programmering?
- Vilka felsökningsstrategier kan ni använda för att ge eleverna möjlighet att lära sig av sina misstag och att bli bättre på programmering?
- Hur kan ni främja elevernas utveckling till att bli både tinkers och planerare?

II. Förbered en aktivitet

Välj en aktivitet att genomföra utifrån de förslag som ges i texten ”Programmering – centrala begrepp”. Utgå från den valda aktiviteten och utforma felsökningsstrategier som underlag för att kunna identifiera vad som är kritiskt i elevens lärande. Om ni undervisar elever i olika årskurser, kan ni behålla samma innehåll men variera antal instruktioner utifrån elevgruppen. Ni kan också variera innehållet både genom att låta eleverna följa givna instruktioner och att låta dem skapa egna instruktioner. Ni kan använda punkt 7, 8 och 9 i ”Lektionsobservation – Protokoll” för att dokumentera genomförandet av aktiviteten.

Del 5: Moment C – aktivitet

I. Genomför aktivitet

Genomför aktiviteten med fokus på att få syn på elevernas kunskaper om ekvationer. Dokumentera vad eleverna ger uttryck för och vilka aspekter som är kritiska i deras lärande.

II. Genomför aktivitet

Genomför aktiviteten med fokus på att få syn på elevernas förmåga att felsöka en instruktion. Dokumentera vad eleverna ger uttryck för och vilka aspekter som är kritiska i deras lärande.

Del 5: Moment D – gemensam uppföljning

I. Diskutera

Utgå från elevernas språkliga medvetenhet i algebra i era diskussioner. Analysera elevernas lösningar samt era anteckningar om de observationer som ni gjorde i ” Lektionsobservation – Protokoll” punkt 9.

- Vilken språklig medvetenhet uppvisar eleverna i genomförandet av aktiviteten? Hur förklarar ni detta?
- Vad verkar eleverna redan ha förståelse för?
- Vilka aspekter av innehållet är kritiska för era elever?

I. Reflektera

Avsätt de sista tio minuterna för att reflektera över:

- Vad gjorde vi?
- Vad lärde jag mig?

Sammanfatta tillsammans arbetet med denna del i några punkter.

II. Diskutera

Utgå från elevernas språkliga medvetenhet i programmering i era diskussioner. Analysera era anteckningar och diskutera:

- Vilka felsökningsstrategier använder eleverna vid genomförandet av aktiviteten? Hur förklarar ni detta?
- Vad verkar eleverna redan ha förståelse för?
- Vilka aspekter av innehållet är kritiska för era elever?
- Vilka möjligheter erbjöds eleverna i att utvecklas som både tinkers och planerare i den aktivitet ni genomförde?

II. Reflektera

Avsätt de sista tio minuterna för att reflektera över:

- Vad gjorde vi?
- Vad lärde jag mig?

Sammanfatta tillsammans arbetet med denna del i några punkter.