



HÖGSKOLAN  
DALARNA

## **Examensarbete 2 för ämneslärarexamen inriktning 7–9**

Avancerad nivå

### **Programmering för matematisk problemlösning i årskurs 7-9**

---

#### **Hur hänger det ihop?**

#### **Computer programming for mathematical problem solving for 13-16 year olds**

Författare: Charis Snell

Handledare: Eva-Lena Erixon

Examinator: Anna Teledahl

Ämne/huvudområde: Matematikdidaktik

Kurskod: MD3007

Poäng: 15 hp

Examinationsdatum: 4 november 2017

Vid Högskolan Dalarna finns möjlighet att publicera examensarbetet i fulltext i DiVA. Publiceringen sker open access, vilket innebär att arbetet blir fritt tillgängligt att läsa och ladda ned på nätet. Därmed ökar spridningen och synligheten av examensarbetet.

Open access är på väg att bli norm för att sprida vetenskaplig information på nätet. Högskolan Dalarna rekommenderar såväl forskare som studenter att publicera sina arbeten open access.

Jag/vi medger publicering i fulltext (fritt tillgänglig på nätet, open access):

Ja

Nej

**Abstract:**

Syftet med studien är att undersöka hur programmering för matematisk problemlösning kan integreras i matematikundervisning i årskurs 7-9. Skolverket har nyligen ändrat kursplanen för matematik så att programmering för matematisk problemlösning inkluderas i det centrala innehållet. För att undersöka syftet genomfördes studien i två delar. Lärare med programmeringskompetens intervjuades om hur de anser att programmering för matematisk problemlösning kan integreras i matematikundervisning. Två programmeringsverktyg undersöktes för att få svar på vilka möjligheter till arbete med matematisk problemlösning programmeringsverktyg erbjuder. Resultaten analyserades utifrån Bernsteins teori om klassifikation och inramning och visar att det finns många olika sätt att integrera programmering i matematikundervisning. Stora möjligheter finns för integrering av programmering för matematisk problemlösning med det centrala innehållet i kursplanen för matematik. Olika verktyg, uppgifter och arbetssätt erbjuder olika pedagogiska möjligheter.

**Nyckelord:**

Programmering i skolan, matematikundervisning, problemlösning, IKT, Bernstein  
Programming in school, mathematics education, problem solving, ICT, Bernstein

## Innehållsförteckning

1 Inledning .....	1
2 Syfte och frågeställningar .....	2
3 Bakgrund .....	2
3.1 Digital kompetens .....	2
3.2 Digitala verktyg.....	3
3.3 Programmering.....	3
3.4 Matematik i läroplanen .....	7
3.5 Problemlösning och programmering.....	8
3.6 Bernsteins teori om kategorisering och inramning .....	9
4 Metod .....	11
4.1 Metodval .....	11
4.2 Etiska frågor .....	16
4.3 Validitet och reliabilitet.....	16
5 Resultat.....	16
5.1 Hur anser lärare att programmering för matematisk problemlösning kan integreras i matematikundervisning? .....	16
5.2 Vilka möjligheter till arbete med matematisk problemlösning erbjuder programmeringsverktyg? .....	20
6 Diskussion .....	27
6.1 Metoddiskussion .....	27
6.2 Resultatdiskussion.....	27
7 Förslag på framtida forskning .....	31
Litteratur.....	33

# 1 Inledning

Det finns flera syften med att läsa ämnet matematik i skolan. Matematik beskrivs som ”en kreativ, reflekterande och problemlösande aktivitet som är nära kopplad till den samhälleliga, sociala, tekniska och digitala utvecklingen” (Skolverket 2017a, s. 56). Att eleverna behärskar matematik leder till att de i framtiden har förutsättningarna för att ”fatta välgrundade beslut i vardagslivets många valsituationer” och ökar deras möjligheter att ”delta i samhällets beslutsprocesser” (Skolverket 2017a, s. 56). Grundläggande matematiska kunskaper ses som en viktig del i upprätthållandet av de demokratiska värderingarna som skolan ska förmedla. Problemlösningsförmåga och förmågan att använda digitala verktyg för att lösa problem ingår i de övergripande kunskaperna som skolan ska se till att eleverna utvecklar under grundskoletiden.

I den reviderade läroplanen för 2017 inkluderas programmering i kursplanen för matematik. Den nya läroplanen kan användas redan nu, höstterminen 2017, och från och med höstterminen 2018 är den obligatorisk. Den 2 september 2017 rapporterade Svenska Dagbladet att åtta av tio högstadielärare i matematik känner sig osäkra på undervisning i programmering (Thurfjell 2017) och Skolverket bekräftar att det finns ett stort behov av kompetensutveckling (Skolverket 2017b). Att kritiskt analysera och välja programmeringsverktyg och att välja lämpligt ämnesinnehåll för att kunna integrera programmering för problemlösning i den dagliga matematikundervisningen kan vara en stor utmaning för den som saknar kompetens i programmering.

Utbildningen ska vila på vetenskaplig grund, men det är svårt att hitta forskning som stöder införandet av programmering i ämnet matematik. Kring hur programmering praktiskt ska integreras finns lite tidigare forskning. Det bekräftas av rapporten *Översikt avseende forskning och erfarenheter kring programmering i förskola och grundskola* som skrevs på uppdrag av Skolverket (Källander, Åkerfeldt och Petersen 2016, s. 27). Skolverket lyfter fram i sina förslag till förändringar i styrdokumentet att ”det finns synpunkter, framför allt på hur programmering skrivs fram i förslagen” (Skolverket 2016, s. 1). Här antyds att dessa synpunkter är av negativ karaktär. Under arbetsprocessen i att ta fram förslagen efterlystes mer forskning om programmering i skolan, framför allt ”praktiknära forskning” (Skolverket 2016, s. 11). Förslagen för förändringarna remitterades under våren 2016. Bland annat svarade Nationellt centrum för matematikutbildning (NCM) remissinstansen med att det saknades stöd i forskning för införande av förändringarna i läroplanen och ifrågasatte ”om det ger goda resultat att lägga ansvaret för programmering i kursplanen för matematik” (Skolverket 2016, s. 21). Även Göteborgs universitet ifrågasatte den vetenskapliga grunden för införandet av programmering i undervisning. Ett stort behov av kompetens-utveckling och även förändringar i lärarutbildningarna nämns i remissvaren (Skolverket 2016, s. 21). I min egen pågående lärarutbildning har inte undervisning i programmering ingått. Mishra och Koehler (2006, s. 1018) menar att det finns en bristande teoretisk grund för forskning kring integration av digital teknik i undervisning. De menar att mer fokus behövs på hur tekniken används i klassrummet. Det finns således ett stort behov av forskning kring programmering i matematikundervisning och dess praktiska införande.

Under författarens tidigare arbete påvisades delvis motsägelsefulla resultat gällande kopplingar mellan programmering och problemlösningsförmågan hos elever (Snell 2017, s. 39). Det finns dock studier som visar att eleverna kan lära sig matematiskt innehåll genom programmering för matematisk problemlösning samtidigt som de lär sig programmering (Martínez Ortiz 2015; Li et al. 2014; Grubbs 2013; Ardito et al. 2014; Calder 2010).

Eftersom förändringarna i kursplanen nu är verklighet och det finns en konstaterad brist i programmeringskunskaper bland lärare i matematik är det högst intressant att utforska vilka möjligheter som finns till att skapa en undervisning som integrerar programmering som ett verktyg för matematisk problemlösning och hur detta praktiskt kan genomföras.

## 2 Syfte och frågeställningar

Syftet med studien är att undersöka hur programmering för matematisk problemlösning kan integreras i matematikundervisning i årskurs 7-9.

Utifrån detta syfte formuleras följande frågor:

- Hur anser lärare att programmering för matematisk problemlösning kan integreras i matematikundervisning?
- Vilka möjligheter till arbete med matematisk problemlösning erbjuder programmeringsverktyg?

## 3 Bakgrund

Det finns många faktorer som samverkar och ska beaktas i integreringen av programmering för problemlösning i matematikundervisning. I detta avsnitt behandlas digital kompetens i läroplanen och begreppet *datalogiskt tänkande*, som är starkt kopplat till programmering, definieras. Programmering och dess olika former beskrivs, för att få en bakgrund till vad programmering är och de olika sorters verktyg som finns tillgängliga för lärare och elever. Det centrala innehållet i kursplanen för matematik beskrivs för att ge en bakgrund till det matematiska innehållet som potentiellt kan undersökas i arbete med programmering i matematikundervisning. Matematiska förmågor och kopplingar mellan programmering, problemlösning och andra matematiska förmågor beskrivs utifrån tidigare forskning. Översikten ska skapa en grund för att kunna förstå hur programmering skulle kunna kopplas ihop med det matematiska innehållet och därigenom bidra till en ökad digital kompetens hos eleverna.

### 3.1 Digital kompetens

I de övergripande kunskapsmålen för grundskolan krävs det att eleverna ska kunna "använda sig av matematiskt tänkande", att kunna "lösa problem" och att kunna använda digitala verktyg för "problemlösning, skapande... och lärande" (Skolverket 2017a, s. 13). Kursplanen har reviderats för att ge ett tydligare fokus på elevernas digitala kompetens. Eleverna måste utveckla en förståelse för hur digitalisering påverkar samhället vi lever i och kunna "förhålla sig till... information på ett kritiskt och ansvarfullt sätt" (Skolverket 2017b). Eleverna ska även utveckla en förmåga att kreativt kunna använda digitala verktyg för att lösa problem (Skolverket 2017b).

Arbete med programmering för matematisk problemlösning ska därför även ha ett bredare syfte och eleverna ska utveckla en förståelse för hur människor kan påverka samhället och samhällsdebatt genom användning av digital teknik och programmering (Skolverket 2017d, s. 8).

Begreppet *digital kompetens* grundas i EU:s beskrivningar av nyckelkompetenser som togs fram 2006 och innefattar bland annat informationssökning och bearbetning, kritiskt tänkande, kreativitet och innovation (Skolverket 2017d, s. 7). Regeringens *Digitaliseringskommission* har utgått ifrån EU:s beskrivning och kommit fram till en beskrivning av digital kompetens som innefattar:

- kunskaper att söka information, kommunicera, interagera och producera digitalt
  - färdigheter att använda digitala verktyg och tjänster
  - förståelse för den transformering som digitaliseringen innebär i samhället med dess möjligheter och risker
  - motivation att delta i utvecklingen
- (SOU 2015:28, s. 103)

Ett begrepp som kan ses som en del av en digital kompetens och som är relevant för problemlösning är *innovativ förmåga*. Eleverna måste förstå att digitala lösningar har skapats av människor och att vi även har möjlighet utveckla och förändra dem (Skolverket 2017d, s. 9).

### 3.1.1 Datalogiskt tänkande

*Datalogiskt tänkande (computational thinking)* är ett brett begrepp. Det innefattar processerna som måste genomföras för att arbeta med programmering eller design för programmering, och handlar bland annat om ”problemlösning, logiskt tänkande, att se mönster och att skapa algoritmer som kan användas vid programmering” (Skolverket 2017d, s. 9). Wing (2006, s. 35), i en flitigt citerad artikel, menar att datalogiskt tänkande är en grundläggande förmåga som alla barn har rätt att lära sig och betonar människans kreativitet som faktorn som möjliggör användningen av digitala verktyg för att lösa komplexa problem.

### 3.2 Digitala verktyg

Digitala verktyg kan innefatta många olika saker, från miniräknare och grafräknare till smarttelefoner, surfplattor och datorer (Skolverket 2017c, s. 9). Dessa är exempel på hårdvaror. Begreppet digitala verktyg kan även innefatta mjukvaror som appar och programmeringsverktyg. Det som beskrivs som digitala verktyg kan förändras i framtiden och det är därför viktigt att förstå begreppet på ett brett och flexibelt sätt (Skolverket 2017d, s. 8). När en lärare ska välja vilket programmeringsspråk de vill använda i undervisning är det viktigt att beakta vilka digitala verktyg eleverna har tillgång till eftersom inte alla programmeringsverktyg fungerar på alla typer av hårdvara (Berry 2017, s. 45). Till exempel så fungerar inte blockprogrammeringsverktyget *Scratch* på surfplattor eller I-pads, ett digitalt verktyg som många skolor har valt till sina elever.

### 3.3 Programmering

Programmering är ett sätt att skriva programmeringskod för att få en dator eller annat digitalt verktyg att utföra olika operationer. Datorer måste ha tydliga stegvisa

instruktioner för att kunna genomföra en operation. I studien handlar programmering om ett sätt att skapa programmeringskod i ett digitalt verktyg för att undersöka matematiska problem. Det finns många olika programmeringsspråk som kan användas för programmering, men för att få datorn att göra någonting måste programmeringskoden på något sätt översättas till ettor och nollor (binärkod) som kan förstås av datorn (Kjällander, Åkerfeldt och Petersen 2016, s. 7). Som Skolverket påpekar utvecklas programmeringsspråk kontinuerligt och ett av syftena med programmering i läroplanen är att ”eleverna ska utveckla en generell förståelse för programmering och hur den kan påverka omgivningen” (Skolverket 2017d, s. 8). Det är således mindre viktigt vilket programmeringsspråk lärare väljer att jobba med, förutsatt att det ger eleverna möjlighet att utveckla en generell förståelse av programmeringsprinciper och dess användning (Skolverket 2017d, s. 9). Hubweiser et al. (2015, s.76) har sammanställt en lista över vilka programmeringsspråk som används i skolor (i alla ämnen). Det finns även flera språk som har utvecklats sedan denna lista skrevs som nu är i bruk, till exempel *Snap!* (<http://snap.berkeley.edu/>) och *Swift* (<https://developer.apple.com/swift/>).

### 3.3.1 Algoritmer

En algoritm kan enligt Skolverket beskrivas som ”en begränsad uppsättning väldefinierade instruktioner för att lösa en uppgift. Den kan också beskrivas som en systematisk procedur för hur man genom ett begränsat antal steg utför en beräkning eller löser ett problem” (2017c, s. 7). Det beskrivs även i programmeringskontext som ”entydiga stegvisa instruktioner som styr en dator att göra det man vill” (Skolverket 2017c, s. 17). I arbete med programmering för matematisk problemlösning måste tydliga stegvisa instruktioner skapas för att formulera problemet matematiskt och utnyttja det digitala verktyget för att undersöka möjliga lösningar. Skolverket menar att ”en generell modell kan uttryckas som en algoritm som är skapad utifrån en matematisk eller vardaglig funktion och kan lösa olika typer av problem, till exempel sortera stora mängder data” (2017d, s. 25).

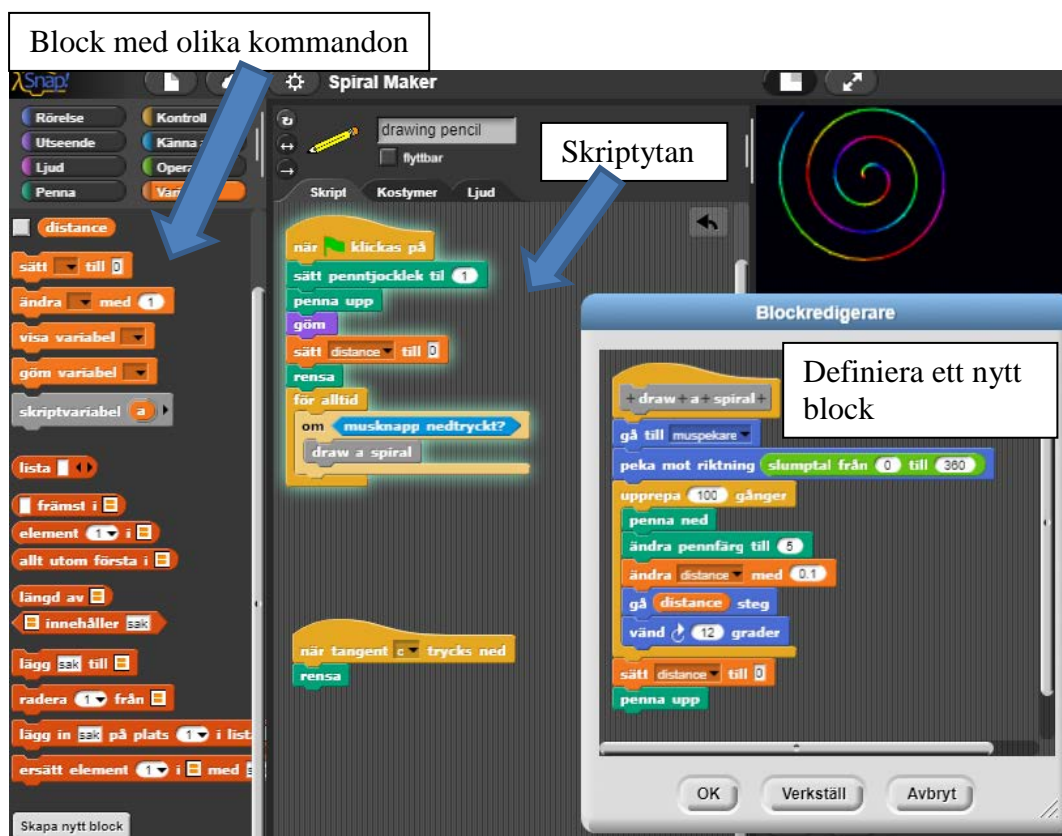
### 3.3.2 Olika programmeringsmiljöer

I detta avsnitt beskrivs några olika programmeringsmiljöer som finns och deras utmärkande drag. Dessa drag skulle kunna påverka hur eleverna kan arbeta med programmering för matematisk problemlösning och därför hur programmering kan integreras i matematikundervisning.

I läroplanen används begreppen *visuella programmeringsmiljöer* och *olika programmeringsmiljöer* (Skolverket 2017a, s. 59-60). Visuella programmeringsmiljöer är digitala verktyg där programmeringskod byggs upp genom att dra in olika element eller block av programmeringskod till skriptytan, där de ordnas för att bygga ihop flera rader av programmeringskod. Ett exempel på en visuell programmeringsmiljö är *Snap!* (<http://snap.berkeley.edu/index.html#bjc>).

Exemplet i Figur 1 är från ett program i *Snap!* som ritar en spiralform utifrån punkten där muspekaren befinner sig på skärmen. Den som skriver koden väljer kommandon de vill använda från den vänstra menyn och drar in dem till mitten (skriptytan) där de sätts ihop till ett program. Detta är ett exempel på en algoritm. När programmet körs ses resultatet till höger. Här har ett nytt kommando som heter ”draw a spiral” skapats. Ett exempel på ett kommando är att pennan ska sättas ner (*penna ner*) så att ett streck visas på skärmen. Matematiska begrepp som grad och avstånd används i

programmet och eleverna kan variera dessa variabler för att undersöka konsekvenserna. En fördel vid visuella programmeringsmiljöer är att programmeringsblocken bara kan sättas ihop på vissa sätt och därför minskar risken för syntaktiska fel (Berry 2017, s. 47). Blockprogrammering i undervisning kopplad till matematisk problemlösning beskrivs i studier av Li, Vandermeiden, Lemieux och Nahoo (2014), Calder (2010) och Parker (2012). I dessa studier användes blockprogrammering till att bygga spel som innehöll matematik.



Figur 1: Exempel från *Snap!*/blockprogrammeringsverktyg

Ett annat sätt att använda programmering i undervisning är att programmera robotar som rör sig runt en bana eller utför uppdrag. Detta sker vanligtvis genom blockprogrammering men det finns även möjlighet att programmera med textbaserad kod. Eleverna behöver använda matematik för att beräkna hur de ska programmera roboten så att den lyckas köra runt banan. De behöver även tänka logiskt och programmera roboten så att den utför olika rörelser i rätt ordning. Programmering av robotar i undervisning med kopplingar till matematisk problemlösning beskrivs av Grubbs (2013), Ardito, Mosley och Scollings (2014), Castledine och Chalmer (2011), Hussain, Lindh och Shukur (2006), Lindh och Holgersson (2007) och Martínez Ortiz (2015).

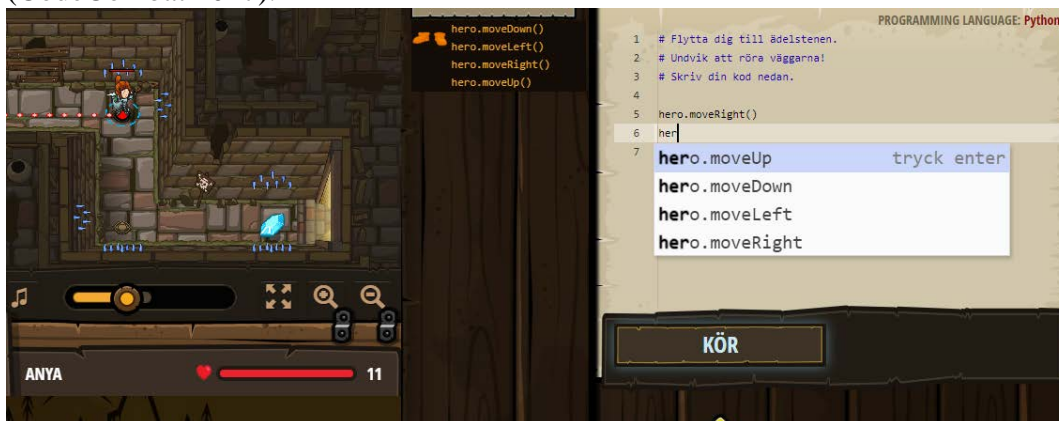
Skolverket menar att en av de olika programmeringsmiljöerna som eleverna ska arbeta med i de högre årskurserna kan vara textbaserad programmering, där eleverna skriver koden med hjälp av ett vanligt tangentbord (2017d, s. 22). Det finns ett stort antal olika programmeringsspråk att välja emellan. Textbaserade språk erbjuder stora möjligheter eftersom nästan allting som är möjligt att utföra på en dator kan teoretiskt sätt programmeras. Det kan dock vara tidskrävande och svårt för nybörjare och Rolandsson menar att ”många elever [har] svårt att förstå ämnet”



(programmering) (2015, s. 60). Barry (2017, s. 49) förklarar att textbaserad programmering har historiskt sätt varit svårt för barn att bemästra när de lär sig programmering. Okita (2014, s. 858) fann däremot att elever i åldern 9-11 år som först hade lärt sig programmera med ett textbaserat verktyg kunde lättare överföra sina programmeringskunskaper till nya problem än de elever som hade börjat med ett blockprogrammeringsverktyg.

Ett textbaserat språk som länge har funnits i skolor är LOGO som utvecklades under 1960-talet. Syftet med arbete i LOGO var att hjälpa elever att utveckla matematiskt och logiskt tänkande. Språket skulle varken ha tröskel eller tak och därigenom låta elever själva utforska problem (Feurzeig, Papert & Lowler 2011, s. 487-490). Programmeringsundervisning som använder LOGO och kopplas till matematisk problemlösning beskrivs även av Aydin (2005) och Ratcliff och Anderson (2011).

Programmeringsspel är en speciell programmeringsmiljö som kan användas för att lära sig textbaserad programmering. Användaren guidas igenom olika nivåer där de lär sig att skriva programmeringskod för att utföra olika steg i ett uppdrag. Ett exempel är *CodeCombat* (<https://codecombat.com/home>) som visas i Figur 2. Det erbjuder ett val av textbaserat språk mellan JavaScript, Python och Lua. När en användare börjar skriva kod visar sig en ruta med möjliga val av kod för att stödja skrivandet. Detta minskar risken att användaren skriver på ett felaktigt sätt. Spelare lär sig gradvis ett större antal kommandon genom att spela igenom olika nivåer (CodeCombat 2017).



Figur 2: *CodeCombat* programmering med Python textbaserat programmeringsspråk

Berry beskriver även kalkylblad som Microsoft Excel som ett textbaserat programmeringsspråk och menar att datalogiskt tänkande krävs på samma sätt i utvecklande av en modell i ett kalkylblad som i kodskrivande (Berry 2017, s. 51). Skolverket menar att kalkylblad är ett bra verktyg för att ”hantera stora mängder data eller ta fram prognoser med hjälp av matematiska modeller och programmering” (Skolverket 2017d, s. 22). Genom att skapa referenser mellan celler i kalkylbladen fungerar cellerna som variabler och kan användas för att skapa matematiska modeller.

### 3.3.3 Val av digitala programvaror för matematikundervisning

Attard och Northcote (2011, s. 1) menar att det är viktigt att det är de pedagogiska målen som driver valet av verktyg. Läraren måste kunna analysera verktyget utifrån syftet med undervisningen. Mishra och Koehler (2006) presenterar begreppet *technological pedagogical content knowledge* (TPCK) (teknologiskt pedagogiska

ämneskunskaper) som bygger på Skulmans ideér om pedagogiska ämneskunskaper (*pedagogical content knowledge* - PCK). Lärare behöver utöver ämneskunskaper och pedagogiska kunskaper ha kunskaper om teknik, och även förstå hur de tre faktorerna interagerar i undervisningsprocessen (s. 1026). De menar att eftersom digital teknik förändras så fort måste lärare kunna lära sig nya verktyg kontinuerligt allt eftersom de förändras och ersätts med nya (s. 1023). Lärare måste även förstå hur ämnesinnehåll kan förändras genom användning av teknik (s. 1028). De flesta digitala verktyg är inte utvecklade med pedagogik som drivande kraft. Lärare måste granska möjligheter och begränsningar för att kunna använda verktygen för att nå specifika pedagogiska mål. Mishra och Koehler varnar för att om läraren inte gör detta kan det finnas en risk att eleverna inte lär sig ämnesinnehållet utan endast lär sig tekniken (s. 1032).

Berry fokuserar på faktorer som lärare måste beakta i valet av programmeringsverktyg för undervisning och menar att tillgång till undervisningsmaterial och stödgrupper online är viktig. Läraren måste även fundera på vilket programmeringsspråk som är mest lämpligt för eleverna med tanke på svårighetsnivå (2017, s. 45).

Palmér och Helenius (2016) presenterar tre verktyg som kan användas för analys av digitala programvaror och de möjligheter de erbjuder för matematikundervisningen. De påpekar att analysen måste utgå ifrån den unika kontexten där verktyget ska användas. Analysverktygen fokuserar på olika aspekter av hur programvarorna kan användas. Ett av verktygen benämns *Klassifikation och inramning* och utgår ifrån Bernsteins pedagogiska teorier. Det används av Palmér (2015) för att klassificera appar som används i förskolan och för att analysera kontexten kring användningen av apparna.

### 3.4 Matematik i läroplanen

Att kunna använda sig av ett matematiska tankesätt i livet utanför skolan beskrivs som ett av skolans övergripande mål (Skolverket 2017a, s. 13). Kursplanen i matematik delas upp i en syftesdel med fem övergripande förmågor:

- formulera och lösa problem med hjälp av matematik samt värdera valda strategier och metoder,
- använda och analysera matematiska begrepp och samband mellan begrepp,
- välja och använda lämpliga matematiska metoder för att göra beräkningar och lösa rutinuppgifter,
- föra och följa matematiska resonemang, och
- använda matematikens uttrycksformer för att samtala om, argumentera och redogöra för frågeställningar, beräkningar och slutsatser (Skolverket 2017a, s. 57)

Undervisningen ska ge eleverna förutsättningar att utveckla dessa förmågor.

#### 3.4.1 Det centrala innehållet i kursplanen för matematik

Det centrala innehållet i kursplanen för matematik delas upp i sex huvudområden. Dessa är taluppfattning och tals användning, algebra, geometri, sannolikhet och statistik, samband och förändring samt problemlösning (Skolverket 2017a, 60-61).

### **3.4.2 Problemlösning i matematik**

Problemlösning inkluderas i kursplanen både som en av de övergripande förmågorna som eleverna ska utveckla och som ett område i det centrala innehållet (Skolverket 2017a, s.56-61). Ett matematiskt problem beskrivs som en situation eller uppgift ”där eleverna inte på förhand känner till hur problemet ska lösas” (Skolverket 2017c, s. 25). Ett matematiskt problem måste ha något matematiskt innehåll och problemlösning ska inkludera problem inom de olika ämnesområdena (Skolverket 2017a, s. 61).

Det tar lång tid för elever att utveckla strategier för problemlösning och utveckla sin problemlösningförmåga (Garofalo och Lester, 1987, s. 166). Det finns många faktorer som påverkar elevers problemlösningförmåga. Möllehed (2001, s. 158) utvecklade en modell över påverkansfaktorer. Faktorena kan delas upp i tre kategorier: matematiska faktorer, kognitiva faktorer och övriga faktorer. Det är inte enbart matematiska faktorer som påverkar hur elever lyckas med problemlösninguppgifter.

## **3.5 Problemlösning och programmering**

Detta avsnitt behandlar kopplingar mellan problemlösning i matematik och programmering.

### **3.5.1 Kopplingar mellan matematisk problemlösning och programmering i styrdokument**

Skolverket menar att programmering erbjuder möjligheter att undersöka matematik genom utforskning, och kan användas för att genomföra avancerade beräkningar, utforska geometri, sannolikhet och statistik och att skapa visualiseringar av abstrakta fenomen (2017d, s. 9). I den reviderade kursplanen för matematik i årskurs 7-9 inkluderas det centrala innehållet ”hur algoritmer kan skapas, testas och förbättras vid programmering för matematisk problemlösning” (Skolverket 2017a, s. 61). Eleverna ska även kunna formulera en frågeställning eller ett problem matematiskt och skapa ”enkla matematiska modeller” (Skolverket 2017a, s. 61). Eleverna ska ”kunna använda programmering som ett verktyg i matematik” (Skolverket 2017d, s. 17). Eleverna måste även kunna värdera modellen och tolka resultaten (Skolverket 2017c, s. 26).

### **3.5.2 Kopplingar mellan matematisk problemlösning och programmering i forskning**

Logiskt tänkande är en av faktorerna som påverkar elevernas problemlösningförmåga som identifieras av Möllehed (2001, s. 158) och som även nämns i diskussioner om programmering och datalogisk tänkande. Garofalo och Lester (1985, s. 163) beskriver metakognitiva funktioner som används vid problemlösning för att styra och värdera handlingar och tänkande. Många av de metakognitiva funktioner som används vid problemlösning liknar de som används vid programmering och inkluderar analys av information, planering och utförande av strategi och reflektion över resultat och val av strategi.

*Transfer* eller överföring beskrivs som en process när tidigare lärande eller kunskaper påverkar lärande i en ny situation (Mayer & Wittrock 1996, s. 48). Att

kunna skapa en analogi mellan två problem är en strategi som skulle kunna underlätta för eleverna vid lösning av nya problem. Det kan dock vara svårt för elever att faktiskt upptäcka likheterna mellan problem (s. 55). Det ska därför inte förväntas att bara för att matematisk problemlösning och programmering kräver liknande kognitiva förmågor att eleverna kommer kunna överföra strategier från ett område till ett annat (Snell 2017, s. 10).

Vissa forskare som har undersökt hur undervisning i programmering påverkar elevernas förmågor att lösa matematiska problem har funnit en positiv påverkan på matematiska faktorer, som till exempel begreppsförståelse (Ardito et al., 2014; Aydin, 2005; Calder, 2010; Feurzeig et al., 2011; Grubbs, 2013; Li et al., 2014; Martínez Ortiz, 2015). En positiv påverkan på kognitiva faktorer, som generaliseringsförmåga, rumsuppfattning, resonemangsförmåga och förmåga att reflektera kan skapas genom arbete med programmering (Ardito et al., 2014; Calder, 2010; Castledine & Chalmer, 2011; Li et al., 2014; Parker, 2012; Ratcliff & Anderson, 2011). Några studier fann att även övriga faktorer som uthållighet och motivation påverkades positivt genom undervisning i programmering (Feurzeig et al., 2011; Li et al., 2014; Parker, 2012; Ratcliff & Anderson, 2011). Trots detta fanns det inte något övertygande bevis på en generell förbättring av elevers matematiska problemlösningsförmåga efter arbete med programmering (Snell 2017, s. 29-33).

### **3.5.3 Programmeringsaktiviteter för problemlösning**

Forskning beskriver ett antal olika aktiviteter som elever har genomfört i undervisning i programmering med kopplingar till matematisk problemlösning. Dessa är styrning av robotar, skapande av spel med visuella programmeringsspråk, lösning av strukturerade uppgifter, självstyrd utforskning av programmeringsverktyg och kollaborativt arbete. Eleverna har även skapat spel med syfte att lära ut ett visst matematisk innehåll till andra elever (Snell 2017, s. 34).

## **3.6 Bernsteins teori om kategorisering och inramning**

Detta avsnitt handlar om teorin som används i studien och presenterar och förklarar några begrepp som är viktiga för förståelsen av teorin.

Bernsteins (2000, s. 3) teori om kategorisering och inramning utvecklades för att beskriva den sociala kontexten för skapandet och återskapandet av kultur och makt i samhället. Teorin om klassifikation och inramning kan användas som ett verktyg för att beskriva det som sker i sociala kontexter, som till exempel i skolan eller klassrummet. Bernstein använder begreppet *klassifikation* för att beskriva gränserna mellan olika kategorier och *inramning* för att beskriva hur mycket frihet en individ har att agera i en specifik situation (s. 5). Bernsteins teori har utvecklats och använts i olika forskningssyften för att beskriva undervisningspraktiker.

### **3.6.1 Klassifikation**

Bernstein använder begreppet klassifikation för att beskriva hur starka avgränsningar det finns mellan olika kategorier i en situation. Kategorier kan analyseras på olika nivåer, till exempel mellan olika skolämnen eller på organisationsnivå, mellan de olika grupperingarna (ledning, personal, elever) i en skola (Bernstein 2000, s. 9). Med stark klassifikation är gränserna mellan de olika kategorierna tydliga. Vid en svag klassifikation är gränserna mellan kategorierna

suddiga (s. 6). Till exempel, i en analys av läroplaner skulle en läroplan med stark klassifikation ha tydliga gränser mellan ämnen. En läroplan med svag klassifikation skulle ha mer integration mellan de olika ämnena. Inom matematikämnet skulle till exempel en stark klassifikation betyda att det fanns tydliga gränser mellan de olika ämnesområden som nämns i det centrala innehållet, som algebra, geometri och samband och förändring. En svag klassifikation skulle däremot beskriva en mer integrerad syn på matematikämnet. Gränserna mellan skolan och samhället utanför skolan kan också analyseras med klassifikationsbegreppet (s. 21).

### 3.6.2 Inramning

Inramning handlar om *vem* som kontrollerar *vad* (Bernstein 2000, s. 12). I undervisningskontexten beskriver graden av inramning lärarens och elevernas möjligheter att påverka och styra det som sker. Inramning beskriver kontroll över flera faktorer. Dessa faktorer, adapterade för undervisningssituationer, är:

- att kunna välja innehållet i undervisning
  - att kunna välja i vilken ordning innehållet presenteras och arbetas igenom
  - tempot av det som presenteras (till exempel hur snabbt elever ska arbeta genom ett material)
  - att kunna påverka kriterium för vad som behandlas som acceptabelt som elevarbete, till exempel om lärandemål och förväntningar presenteras tydligt för elever eller är osynliga
  - att kunna kontrollera och påverka relevanta sociala grupper, till exempel eleverna
- (Bernstein 2000, s. 12-13).

Bernstein (2000, s. 13) menar att det alltid finns både en pedagogisk diskurs för ett ämne, som påverkar val av innehållet, ordningen och tempot av arbetet, och en regulativ diskurs som kontrollerar den sociala ordningen i en situation. Den regulativa sociala diskursen inkluderar värderingar och ordningsregler för ordning och beteende i skolan, men påverkar även urvalet av det som ingår i ämnens pedagogiska diskurser (s. 34). Den regulativa diskursen kan begränsa möjliga värden för den pedagogiska diskursen. En analys av den regelgivande sociala diskursen ligger utanför ramen för denna studie, som har en matematikdidaktisk fokus.

### 3.6.3 Bernsteins teori i forskning

Bernsteins idéer har tolkats och utvecklats av forskare som analysverktyg. Palmér (2015, s. 369) använder teorin för att kategorisera appar som används av förskolebarn inom matematik, och även för att analysera kontexten och aktiviteterna kring användningen av apparna. Syftet med studien var att utveckla en förståelse av hur man kan välja appar för att planera lärande (s. 370). Figur 3 visar Palmérs kategorisering av appar utifrån Bernsteins teori. En stark klassifikation exemplifieras av appar med ett tydligt matematiskt innehåll. Appar med svag klassifikation innehåller tillämpad matematik. Stark inramning (*framing*) har appar som inte är interaktiva och i dessa appar begränsas eleverna i hur de kan agera. Appar med svag inramning är interaktiva och eleverna mindre begränsade i hur de kan agera.

	<b>Strong framing</b>	<b>Weak framing</b>
<b>Strong classification</b>	<i>Non-interactive applications containing strictly mathematics.</i>	<i>Interactive applications containing strictly mathematics.</i>
<b>Weak classification</b>	<i>Non-interactive applications containing integrated mathematics.</i>	<i>Interactive applications containing integrated mathematics.</i>

Figur 3: Palmér (2015, s. 371) kategorisering av appar utifrån Bernstein

Palmér visade att pedagoger i förskolan interagerade på olika sätt med barnen beroende på vilka matematikappar barnen arbetade med (s. 374). Resultaten i studien visade att barnens deltagande, kommunikation kring matematik och därför möjligheter att lära matematik påverkades av klassifikation och inramning av apparna (s. 377). Appar med stark inramning verkade minska barnens deltagande och barnen fick få möjligheter till reflektion. Appar med svag inramning ökade barnens deltagande och kommunikation men krävde en större insats av pedagogerna. Med svag klassifikation och svag inramning var kommunikation mellan eleverna och pedagogerna inte alltid fokuserat på matematik, medan med stark klassifikation och svag inramning var matematik alltid i fokus (s. 377). Palmér påpekar att denna kombination, med stark klassifikation och svag inramning i appar, var den som var svårast att hitta (s. 378). Palmér påpekar att resultaten stöds av andra forskningsresultat men att tillgång till en kompetent lärare som planerar effektiv undervisning är avgörande för möjligheter till matematikinläring (s. 378). Resultaten handlar om förskolebarn och inte elever i årskurs 7-9 som denna studie fokuserar på men är ändå intressanta.

## 4 Metod

I detta avsnitt presenteras metoden som valdes för studien. Urvalet av studiedeltagare och programmeringsverktyg beskrivs tillsammans med etiska frågor. Genomförandet, validitet och reliabilitet diskuteras.

### 4.1 Metodval

Som konstaterats i introduktionen är programmering i matematikundervisning ett nytt och relativt outforskat område. Därför behövs ett flexibelt och fantasifullt tillvägagångsätt. Larsen (2009, s. 20) menar att en bred frågeställning är vanlig när det inte finns så mycket tidigare forskning om ämnet, vilket är fallet med programmering i matematikundervisning. Utifrån syftet, att undersöka hur programmering för matematisk problemlösning kan integreras i matematikundervisning i årskurs 7-9, bestämdes undersökningsmetoder för studien. Kvalitativa metoder valdes som mest lämpliga. Larsen menar att kvalitativa metoder passar väl in med målet i undersökningen av att få en ”helhetsförståelse av enskilda

fenomen” (Larsen 2009, s. 23), i detta fall integreringen av programmering för matematisk problemlösning i matematikundervisning.

För att undersöka hur lärare anser att programmering för matematisk problemlösning kan integreras i matematikundervisning valdes intervjuer med lärare som metod. Kvale och Brinkmann (2009, s. 185) argumenterar för kvalitativa intervjuer som metod inom explorativ forskning. För att undersöka vilka möjligheter till arbete med matematisk problemlösning programmerings-verktyg erbjuder valdes en analys av potentiella programmeringsverktyg som metod. Efter sökningar i litteraturen för att hitta forskning som behandlar användning av appar eller digitala verktyg i matematikundervisning hittades en teoretisk utgångspunkt som bedömdes lämplig och som kunde användas för att analysera både intervjuvaren och programmeringsverktygen.

#### **4.1.1 Urval av intervjudeltagare**

Rektorer eller förstelärare med ansvar för matematik på alla högstadieskolor i två kommuner kontaktades angående möjligheter för lärare på skolorna att delta i studien. En skola svarade inte. Övriga skolor svarade och alla matematiklärare som hade programmeringserfarenhet på dessa skolor kontaktades. Lämplighet i form av erfarenhet var en faktor i urvalet eftersom det bedömdes att lärarna borde ha någon form av erfarenhet kring användning av programmering för att kunna delta. Urval har skett delvis utifrån självval, att lärarna kände att de hade tid och vilja att delta i studien. En lärare kunde inte delta på grund av tidsbrist under den tilltänkta intervjuperioden. Även två lärare som jobbade på andra skolor men som ansågs lämpliga för att bidra till studien kontaktades. En av dessa kunde delta i studien. Totalt genomfördes fem intervjuer. De intervjuade lärarna arbetar på tre olika skolor.

Intervjupersonerna fick först en förfrågan via email för att få reda på om de kunde tänka sig delta i studien. De fick sedan ett informantbrev samtidigt som en lämplig tid för intervjun bestämdes. Slutligen fick de ett dokument med urval från den reviderade läroplanen, för att informera dem om förändringarna om de inte redan var medvetna om dem, och för att de bättre skulle kunna förstå syftet med studien. Även intervjuguiden skickades ut så att lärarna fick möjlighet att fundera på vilka av sina erfarenheter som skulle kunna vara av intresse för studien.

#### **4.1.2 Intervjumetod**

En semistrukturerad intervjudesign skapades för intervjuerna. Kvale och Brinkmann (2009, s. 121) menar att en explorativ intervju kan vara öppen och strukturen kan vara mindre strikt än i en hypotesprövande intervju. De föreslår att en intervju skall vara både tematisk och dynamisk för att både belysa de ämnen som undersöks i studien och för att underlätta för intervjupersoner att kunna berätta om sina erfarenheter och tankar (s. 147). Widerberg (2002, s. 190) menar att analysen och metoden bör hänga ihop. Bernsteins teori skulle användas för att analysera klassifikation och inramning. Kvale och Brinkman påpekar att syftet med intervjuer är att få svar som kan analyseras utifrån den valda teorin för att få svar på forskningsfrågan (2009, s. 148). Därför har intervjuguiden skapats utifrån forskningsfrågan, att få syn på hur lärare anser att programmering för matematisk

problemlösning kan integreras i matematikundervisning, och även med tanke på det teoretiska perspektivet för analys (*Bilaga 2 – Intervjuguide*).

Intervjuguiden testades i en första intervju med en lärare. Efter detta tillfälle justerades intervjuguiden till viss del genom att ta bort en fråga, formulera om en fråga, och ändra i intervjustrukturen för att få en mer sammanhängande struktur. Denna provintervju gav intervjuaren möjlighet att träna färdigheter i att genomföra intervjuer (Widerberg 2002, s. 102). Intervjuerna tog mellan 20 och 40 minuter att genomföra.

#### **4.1.3 Transkriberingsprocess**

För att det som informanterna sa i intervjuerna skulle kunna transkriberas spelades intervjuerna in. Transkriberingsprocessen är en tolkning av talat språk till skriftligt språk och innebär en förlust av information som tonfall och kroppsspråk (Kvale & Brinkmann 2009, s. 194). Kvale och Brinkmann påpekar att intervjuutskriften inte är det som analyseras utan ett verktyg för att kunna tolka det som har sagts (s. 208). Det som lärarna har sagts i intervjuer har analyserats men samtidigt fanns minnen om hur det har sagt och lärarnas kroppsspråk kvar vilket bidrar till ett rikare material än det som kan transkriberas med text. Korta pauser och medhållande från intervjuaren har inte inkluderats i transkriberingen om inte de ansetts ha betydelse för svaret (Kvale & Brinkman 2009, s. 197). Normerna för skriftspråk har följts så långt som möjligt vid transkribering för att underlätta för läsning.

#### **4.1.4 Val av programmeringsverktyg**

Hubweiser et al. (2015, s.76) har sammanställt en lista över alla programmeringsmiljöer som används i skolor. Denna lista utökades med andra programmeringsverktyg genom sökningar på Google och lades samman med verktyg som hade hittats i forskningsartiklar men inte fanns med på Hubweisers et al. lista. Berry (2017, s. 45) beskriver följande faktorer som viktiga för lärare att beakta vid val av programmeringsverktyg för undervisning (se 3.3.2 *Olika programmeringsmiljöer*). Faktorerna är:

- att verktyget fungerar på det digitala verktyget som eleverna har tillgång till
- att det finns undervisningsmaterial
- att det finns stödgrupper online
- att svårighetsnivån är lämplig för eleverna (Berry 2017, s. 45)

Utifrån dessa kriterier ansågs det viktigaste för studiens syfte att det fanns undervisningsmaterial tillgängligt som innehöll matematik för att kunna analysera vilka möjligheter till arbete med matematisk problemlösning programmeringsverktyg erbjuder. Därefter granskades alla kända potentiella verktyg utifrån tillgång till undervisningsmaterial som behandlar matematik på en lämplig nivå.

Två verktyg som klassificeras som olika programmeringsmiljöer (textbaserad och blockprogrammering) valdes för att få en insyn i de möjligheter och begränsningar som finns med programmering i de olika miljöerna. De verktyg som valdes för vidare analys är *RoboBlockly* och *Kojo*. Dessa verktyg valdes eftersom de erbjuder



undervisningsmaterial kopplade till verktygen som kan användas i undervisning för programmering för matematisk problemlösning.

*RoboBlockly* är ett blockprogrammeringsverktyg som utvecklats specifikt för en integrerad undervisning i matematik med programmering (UC Davis Center for Integrated Computing and STEM Education 2017a, s. 1). Det finns undervisningsmaterial tillgängligt som är anpassad till matematikundervisning i grundskolan (dock följer materialet kursplanen i USA). Uppgifterna finns integrerade i verktyget. Verktyget är delvis översatt till svenska. *RoboBlockly* refereras som *blockprogrammeringsverktyg med integrerade matematikuppgifter* i textens fortsättning.

*Kojo* är ett textbaserat programmeringsverktyg som baserats på programmeringsspråket Scala som används i professionella programmeringsuppdrag (Regnell 2015). Regnell menar att *Kojo* har en låg lärandetröskel men tillåter fördjupning upp till universitetsnivå. Många kommandon finns både på svenska och på engelska (Lunds Tekniska Högskola). Det finns undervisningsmaterial tillgängligt både på svenska och på engelska som delvis behandlar matematik. *Kojo* refereras som *textbaserad programmeringsverktyg* i textens fortsättning. Undervisningsmaterialet som analyseras i samband med det textbaserade programmeringsverktyget är Regnells (2016) *Uppdrag med Kojo* samt Pants (2015) *Explorations in Math, Art, Programming, Learning, and Science with Kojo*.

#### 4.1.5 Analysmetod

I denna undersökning används Bernsteins teori om klassifikation och inramning för att analysera både programmeringsverktygen och lärarnas tankar kring integrering av programmering för matematisk problemlösning i matematikundervisning. Klassifikation analyseras utifrån avgränsningar mellan skolan och samhället utanför skolan, mellan skolämnen och mellan ämnesområden inom matematik. Inramning analyseras utifrån elevernas och lärarnas möjligheter att påverka följande faktorer i arbete med programmering för matematisk problemlösning:

- innehållet i undervisning
  - ordningen i vilket innehållet presenteras och arbetas igenom
  - tempot av det som presenteras och uppgifterna som ska genomföras
  - kriterierna för vad som behandlas som acceptabelt arbete, till exempel att det finns en tydlighet kring vad som anses vara en korrekt lösning på ett problem
- (Bernstein 2000, s. 12-13).

Den sista av Bernsteins faktorer som påverkar inramning, som handlar om påverkan över sociala grupper, analyseras inte i denna studie. Fokus är på matematikdidaktik och omfånget skulle bli alldeles för stort för ramen för denna studie om denna faktor undersöktes. De element som analyseras i studien för att undersöka hur programmering för matematisk problemlösning kan integreras i matematikundervisning sammanfattas i Tabell 1.

Tabell 1: Klassifikation och inramning som analysverktyg i denna studie

Klassifikation (stark/svag)	Inramning (stark/svag)
Avgränsningar mellan skolan och samhället utanför skolan	Innehållet i undervisning Arbetsordningen

Avgränsningar mellan skolämnen	Tempot
Avgränsningar inom matematikämnet	Kriterier för acceptabelt arbete

### *Analys av intervjuerna*

Intervjuerna analyseras utifrån Bernsteins teori om klassifikation och inramning med fokus på frågeställningen, ”Hur anser lärare att programmering för matematisk problemlösning kan integreras i matematikundervisning?”. Lärarnas svar analyserades genom att uppskatta värden på klassifikation och inramning för aspekten av undervisning som lärarna beskriver. Klassifikation analyserades på olika nivåer: mellan skolan och samhället, mellan skolämnen och inom matematik. Resultaten presenteras utifrån dessa nivåer. Inramning presenteras utifrån de faktorerna som Bernstein (2000, s. 12) menar bidrar till graden av inramning. Dessa är innehållet i undervisning, arbetsordning, arbetstempot och kriterierna för acceptabelt arbete.

Citat som är av relevans utifrån studiens syfte presenteras för att exemplifiera resultaten. För att kunna presentera en bredd i svaret på frågan om hur lärare tänker kring integrering av programmering för matematisk problemlösning i matematikundervisning presenteras intressanta svar som belyser olika aspekter av studiens syfte, oavsett antal lärare som lyfte synpunkten.

### *Analys av programmeringsverktyg och uppgifter*

För att skapa en översikt av möjligheter att arbeta med programmering för matematisk problemlösning i programmeringsverktygen skapades en matris. I matrisen registrerades analysen av verktygen utifrån möjligheter till arbete med matematisk problemlösning i de olika delområdena i det centrala innehållet för matematik. Analysmetoden liknar den som används av Larkin (2014, s. 30) för att analysera Ipad appar för matematik, utifrån apparnas matematiska innehåll och dess relevans till det centrala innehållet i kursplanen för matematik i Australien. Även det tillhörande undervisningsmaterialet till programmeringsverktygen analyserades och inkluderades i matrisen. Om det fanns en explicit funktion eller kommando i programmeringsverktyget som kunde användas för att arbeta med ett specifikt delområde av det centrala innehållet i matematik registrerades det i matrisen. Om delområdet undersöktes i det tillhörande undervisningsmaterialet registrerades även detta. Denna analys gav en överblick av möjligheterna att arbeta med matematik i de olika verktygen.

Verktygen och undervisningsmaterialet analyserades sedan utifrån Bernsteins teori om klassifikation och inramning (se Tabell 1). För varje verktyg finns det tillhörande uppgifter som behandlar matematik, antingen som dokument med skriftliga uppgifter (till det textbaserade programmeringsverktyget) eller som integrerade uppgifter i verktyget (i blockprogrammeringsverktyget med integrerade matematikuppgifter). Graden av klassifikation och inramning av verktygen och uppgifterna bestämdes genom att använda verktygen och bedöma graden av klassifikation och inramning som presenterades som stark eller svag enligt Bernsteins teori. Graden av inramning som skapades i användning av verktygen bedömdes som stark eller svag beroende på hur mycket arbetsinnehållet, arbetsordning, arbetstempo och kriterier för acceptabelt arbete styrdes av verktyget och uppgifterna.

## 4.2 Etiska frågor

Efter formulering av syfte och val av metod och utifrån *Forskningsetiska anvisningar för examens- och uppsatsarbeten vid Högskolan Dalarna* (Högskolan Dalarna 2008, Bilaga 1) bedömdes det att studien inte behövde etikprövas. I övrigt följdes de fyra huvudkraven inom forskningsetik (Vetenskapsrådet, s. 6). Informationskravet uppfylldes genom att informera intervjudeltagare genom ett informantbrev (*Bilaga 1 – Informantbrev*) och muntlig information vid intervjutillfället om studiens forskningssyfte och att deltagande i studien var frivilligt. Deltagare hade rätt att välja om de ville medverka i studien och hade rätt att avbryta medverkan och därigenom uppfylldes samtyckeskravet. Konfidentialitetskravet uppfylldes genom att garantera konfidentialitet för deltagare och genom att hålla kommun och skolor anonyma. Intervjumaterialet ska förstöras efter studiens avslut. Nyttjandekravet uppfylldes genom att information som insamlades om studiens deltagare endast användes för forskningsändamål.

## 4.3 Validitet och reliabilitet

Validitet innebär att data som samlas in är av relevans för frågan som ska besvaras. Valet av kvalitativa metoder och möjligheten som fanns att kunna be om tydliggörande under intervjuernas gång gav möjlighet att nå en hög nivå av validitet (Larsen 2009, s. 81). Kvale och Brinkmann (2009, s. 267) delar upp forskningsprocessen i sju stadier: tematisering, planering, intervju, utskrift, analys, validering och rapportering. De menar att validering ska genomsyra hela forskningsprocessen (s. 268). Därför har en hög nivå av relevans och validitet varit ett mål i alla stadier i studien.

Reliabilitet handlar om samma forskningsresultat skulle kunna reproduceras vid en annan tidpunkt av en annan forskare. Vid intervjuer kan intervjuarens intervjuteknik påverka svaren, till exempel genom att ställa ledande frågor. Transkribering av intervjuer är en annan punkt där reliabilitet kan påverkas (Kvale & Brinkmann 2009, s. 263). Därför har transkriberingsarbetet genomförts noggrant. När analysen av data genomförs innebär det ett tolkande, något som kan förminska reliabilitet (Larsen 2009, s. 81). Dessa frågor behandlas mer ingående i *6.1 Metoddiskussion*.

## 5 Resultat

I detta avsnitt presenteras resultaten från analysen av både intervjuerna och verktygen som svar på forskningsfrågan om hur programmering för matematisk problemlösning kan integreras i matematikundervisning i årskurs 7-9. Den första delen behandlar hur lärare anser att programmering för matematisk problemlösning kan integreras i matematikundervisning och den andra delen beskriver vilka möjligheter till arbete med matematisk problemlösning programmeringsverktyg erbjuder.

### 5.1 Hur anser lärare att programmering för matematisk problemlösning kan integreras i matematikundervisning?

Resultaten av intervjuanalysen som belyser hur lärare anser att programmering för matematisk problemlösning kan integreras i matematikundervisning presenteras här. Resultaten av analysen presenteras utifrån Bernsteins begrepp klassifikation och inramning. De olika nivåer av klassifikation som analyseras är den mellan skolan och samhället, mellan skolämnen och inom matematik. Analysen av

inramning presenteras utifrån de faktorer som Bernstein (2000, s. 12) menar bidrar till graden av inramning. Dessa är innehållet i undervisning, arbetsordning, arbetstempot och kriterierna för acceptabelt arbete.

### **5.1.1 Klassifikation mellan skolan och samhället vid integrering av programmering för matematisk problemlösning**

Lärare menar att ett syfte med programmering för matematisk problemlösning i matematikundervisning är att göra eleverna mer medvetna om hur programmering används i samhället och det vardagliga livet utanför skolan. De menar att detta kan leda till en ökning av elevernas förståelse för matematik som ett relevant och användbart ämne. Ett annat syfte är att ge alla elever möjlighet att testa programmering i skolan för att få kunskap om det som en möjlig yrkesbana. Denna syn på programmering för matematisk problemlösning med fokus på den stora bilden och nyttan med programmering och matematik i vardagslivet och utanför skolämnet ger en svag grad av klassifikation utifrån Bernsteins teori.

### **5.1.2 Klassifikation mellan skolämnen vid integrering av programmering för matematisk problemlösning**

Flera av lärarna föreslår ämnesövergripande arbete som ett sätt att både effektivisera undervisningstiden och skapa ett sammanhang för eleverna. Andra lärare föreslår ett möjligt samarbete med engelska för att ge stöd till elever i arbete med programmeringsspråk som baseras på engelska. Ämnesöverskridande arbete karakteriseras av svaga gränser mellan skolämnen. Detta leder till en svag klassifikation.

Jag tror att det kommer bli mer och mer att man får jobba ämnesövergripande och kanske mer projektliknande att om jag jobba nu med i teknik med programmering då kommer jag kanske kunna koppla det till matten på ett naturligt sätt. (Lärare C)

### **5.1.3 Klassifikation inom matematikämnet vid integrering av programmering för matematisk problemlösning**

Lärare diskuterade olika ämnesområden inom matematik och hur programmering för matematisk problemlösning kunde användas inom ämnet. Vissa lärare uttrycker sig på ett sätt som påvisar en stark klassifikation mellan de olika ämnesområdena, där de hölls avskilda från varandra. Detta kom till uttryck i formuleringar om ett begränsade antal områden där programmering kunde användas. Andra lärare ser programmering för problemlösning i matematik-undervisning som ett visualiseringsverktyg och menar att det kan vara ett sätt för elever att utforska algebra och konkretisera arbete med funktioner. Vissa lärare menar att programmering ska användas som ett verktyg bland andra för att lösa matematiska problem inom hela matematikämnet. Denna synsätt kan leda till svagare klassifikation mellan ämnesområden. En lärare menar att en del av lärarens roll är att försvaga elevernas egna starka klassificering av ämnesområden och underlätta för dem att se likheter mellan programmering och andra matematiska problem.

För dem är det två skilda saker, då gäller det att skapa en brygga emellan, titta här genom programmering kan du faktiskt, och så tvärtom. (Lärare C)

Lärare beskriver hur arbete med programmering för matematisk problemlösning kan leda till att eleverna utvecklar övergripande matematiska förmågor, till exempel resonemangsförmågan och problemlösningsförmågan. Fokus på de övergripande förmågorna ge en svag grad av klassifikation inom matematikämnet. En lärare beskriver hur programmeringsprocessen underlättar för eleverna att kunna förklara matematiskt resonemang. Eleverna genom programmering skriver stegvisa instruktioner vilket är ett stöd för dem i att kommunicera matematiskt resonemang, och underlätta även för läraren att kunna få syn på detta.

De har fått ett sätt att förklara logiken bakom problemlösning. (Lärare B)

Kan de skapa ett program som kan lösa en svår uträkning då har de förstått vad som ligger bakom, hela matematiken. (Lärare B)

Flera lärare förklarar att de anser att arbete med kalkylblad är ett bra sätt för eleverna att utveckla ett grundläggande datalogiskt tänkande. Utveckling av datalogiskt tänkande och problemlösningsförmågan med logiska steg i en viss ordning är en förmåga som kan användas i hela matematikämnet och bidrar därför till en svag klassifikation.

#### **5.1.4 Inramning av undervisningsinnehåll vid integrering av programmering för matematisk problemlösning**

Förslag till innehåll i undervisning i programmering för matematisk problemlösning inkluderar problem som handlar bland annat om talserier och geometriska figurer. Innehållsområdet som nämndes av alla lärare som självklart att använda programmering i var geometri. Många av lärarna hade tidigare arbetat med programmering för att rita geometriska figurer. Vissa lärare har svårt att se vad det matematiska innehållet i arbete med programmering skulle kunna vara och efterlyser mer stöd och en starkare inramning av innehållet från Skolverket. Andra lärare menar att nästan alla områden i det centrala innehållet kan undersökas genom programmering. Lärare anser att det är de själva som ska skapa och kontrollera innehållet i undervisningen i programmering. Lärare föreslår arbete med blockprogrammering, textbaserad programmering, robotar och programmeringsspel. Graden av inramning i programmeringsverktyg anses av lärare vara låg. Ett undantag är i arbete med programmeringsspel där innehållet bestäms av verktyget och eleverna spelar sig igenom olika nivåer där de lär sig olika programmeringskommandon. I detta fall är graden av inramning stark.

#### **5.1.5 Inramning av arbetsordning vid integrering av programmering för matematisk problemlösning**

Lärarna vill själva kontrollera ordningen av innehållet i arbete med programmering för matematisk problemlösning. De menar att en bra progression i uppgifterna både på kort och på lång sikt är viktig för elevernas utveckling, både matematiskt och i att använda programmering som ett verktyg. Att först styra arbete hårt för att eleverna ska få en grundläggande kompetens i att använda verktyget och sedan använder friare uppgifter med en tydlig fokus var ett planeringsätt som föredrogs.

Presenterar verktyget för dem och sen ger dem uppgifter som styr dem och visar dem vad man kan använda det till. (Lärare C)

Inramning är i så fall stark i början av arbetet och svagare när eleverna har utvecklat den grundläggande kompetens som behövs för att använda programmeringsverktyget.

### **5.1.6 Inramning av arbetstempot vid integrering av programmering för matematisk problemlösning**

Lärare menar att det kan vara en utmaning att integrera programmering för matematisk problemlösning för alla elever och menar att blockprogrammering är lämpligare än textprogrammering för att introducera programmering. Lärare påpekar att det kan vara väldigt svårt för vissa elever att kunna använda textbaserad programmering som ett verktyg.

Andra lärare menar att blockprogrammering kan vara en begränsning för vissa elever som vill utvecklas snabbare. Inramning av tempot i programmeringsarbete är oftast låg och ger elever möjlighet att arbeta i ett tempo som de själva bestämmer. Lärare lyfter möjligheten att kunna anpassa undervisning genom integrering av programmering för matematisk problemlösning i matematik-undervisning. Genom att utmana elever att utforska svårare problem under lång tid i små grupper kan de elever som inte alltid utmanas i vanlig undervisning ges möjlighet att utvecklas. Graden av inramning i så fall är svag och tempot bestäms av eleverna.

### **5.1.7 Inramning av kriterier för acceptabelt arbete vid integrering av programmering för matematisk problemlösning**

Inramning av kriterier för acceptabelt arbete analyseras på två sätt. Det är både verktyget och lärare som har kontroll över det som bedöms som acceptabelt i arbete med programmering för matematisk problemlösning.

Textbaserade programmeringsverktyg har ofta osynliga kriterier för vad som är acceptabel som kod. Detta kan leda till hög belastning på läraren som behöver hjälpa många elever samtidigt med felsökning.

Det är ganska korta lektioner som vi har nu och är man en vuxen eller ibland två så är det ganska ohållbart tror jag med felsökning. (Lärare E)

Inramningen är stark men osynlig. Detta ställer höga krav på både läraren och eleverna. Inramning i blockprogrammeringsverktyg är svagare eftersom verktyget oftast hjälper användaren att felsöka programkoden. Lärare menar att blockprogrammering är mer användarvänlig eftersom det är lättare att undvika syntaktiska fel. De menar att blockprogrammering är att föredra på högstadiet för att nå syftet att eleverna ska utveckla en förståelse för programmering, utveckla datalogiska tänkande och använda programmering för att lösa matematiska problem.

Att skriva kod är ganska svårt och då lägger de tiden på det istället för att få fram den här kunskapen som de ska få fram och därför tror jag på blockprogrammering i första hand, mera tidseffektivt helt enkelt. (Lärare D)

Lärare tycker att det är svårt att ta fram tydliga kriterier som kan presenteras för eleverna vid arbete med programmering för matematisk problemlösning. Ett orosmoment för vissa lärare är bedömning och en lärare menar att det är svårt att

förena det stora övergripande syftet med bedömning utifrån kunskapskraven. En utmaning är att förena dessa två syften i undervisning.

Men sen tänker jag aha, hur ska jag bedöma det här sen, vad är det jag bedömer, eller vad kan jag säga, de kan geometriska figurer, ah, bedömning blir nästa steg för vad är det jag vill att de ska kunna egentligen? (Lärare D)

### **5.1.8 Sammanfattning av resultat kring hur lärare anser att programmering för matematisk problemlösning kan integreras i matematikundervisning**

Programmering i matematikundervisning kan bidra till en lägre grad av klassifikation mellan skolan och samhället utanför, mellan ämnen och inom matematikämnet. Programmering för matematisk problemlösning ger eleverna ett verktyg för att tydligare kunna visa sitt matematiska resonemang och kommunicera detta genom att skriva stegvisa instruktioner. Lärare anser att programmering för matematisk problemlösning kan integreras i matematik-undervisning genom att välja matematiska problem som kan analyseras med programmering som verktyg. De menar att det kan vara svårt att veta vilka problem som lämpar sig bäst för arbetet och efterlyser mer stöd och en starkare inramning av innehållet från Skolverket. Innehåll, ordning och tempot bestäms av lärare och delvis elever och programmering erbjuder möjligheter till utmaningar i form av svårare problem och fördjupningsarbete. Inramningsgraden i arbete med programmering är generellt svag.

## **5.2 Vilka möjligheter till arbete med matematisk problemlösning erbjuder programmeringsverktyg?**

Följande avsnitt presenterar resultaten av analysen av verktygen och det tillgängliga undervisningsmaterialet utifrån möjligheterna till arbete med matematisk problemlösning i programmeringsverktyg och Bernsteins (2000) teori om klassifikation och inramning.

### **5.2.1 Klassifikation mellan skolan och samhället vid integrering av programmering för matematisk problemlösning**

Vissa problem som erbjuds i undervisningsmaterialet till det textbaserade programmeringsverktyget kan användas för att visa elever hur matematiken kan användas i situationer i samhället utanför skolan. En uppgift handlar om att skapa ett program som simulerar ett bankkonto (Regnell 2016, s. 32). Genom att använda programmering för matematisk problemlösning på detta sätt kan en svagare klassifikation mellan skolan och samhället skapas.

### **5.2.2 Klassifikation mellan skolämnen vid integrering av programmering för matematisk problemlösning**

Båda programmeringsverktygen erbjuder möjligheter till ämnesövergripande arbete. Vid arbete i både blockprogrammeringsverktyget med integrerade matematikuppgifter och det textbaserade programmeringsverktyget kan den skrivna koden användas för styrning av robotar. Det finns möjligheter att dela projekt, idéer och uppgifter online med andra lärare som arbetar med båda verktygen. Möjligheter till ämnesövergripande arbete med till exempel bild, teknik och fysik finns. Det är

möjligt att rita och skapa animationer med båda programmeringsverktygen. Instruktioner till ett fysikprojekt som handlar om solsystemet och modellering finns tillgängligt till blockprogrammeringsverktyget med integrerade matematikuppgifter. Graden av klassifikation mellan skolämnen är därför svag.

### 5.2.3 Klassifikation inom matematikämnet vid integrering av programmering för matematisk problemlösning

I blockprogrammeringsverktyget med integrerade matematikuppgifter finns det tydliga rubriker till alla uppgifter. Dessa förklarar det matematiska innehållet, till exempel om uppgiften behandlar addition, multiplikation, area, koordinatgeometri, algebra eller funktioner, bland annat. Vissa uppgifter innehåller en blandning av matematiska områden, till exempel, algebra, tal och geometri, men huvudfokus för varje uppgift är tydligt avgränsat och verktyget i samband med materialet har en stark grad av klassifikation inom matematikämnet. Uppgifterna till det textbaserade programmeringsverktyget som behandlar matematik presenteras som en del i en bredare samling av programmeringsuppgifter. Programmering presenteras som ett verktyg som kan användas för att lösa många olika sorters matematiska problem. Det textbaserade programmeringsverktyget har en svag grad av klassifikation inom matematikämnet.

### 5.2.4 Inramning av undervisningsinnehåll vid integrering av programmering för matematisk problemlösning

I blockprogrammeringsverktyget med integrerade uppgifter är det matematiska innehållet i undervisningsmaterialet en integrerad del av verktyget. Graden av inramning är därför stark om uppgifterna följs. Det finns dock möjlighet att arbeta med verktyget för att undersöka andra matematiska problem och i så fall blir graden av inramning svagare.

De föreslagna uppdragen till det textbaserade verktyget har inte ett specifikt matematiskt fokus men många av dem innehåller mycket matematik. Pant (2015, s. 6) presenterar mer explicita kopplingar till matematik i sitt material. Undervisningsinnehållet bestäms av läraren och eleverna med möjlighet till stöd från det tillhörande materialet. Graden av inramning av innehållet är svag.

Ämnesområden som täcks i de olika uppgifterna i blockprogrammeringsverktyget och det textbaserade programmeringsverktyget visas i Tabell 2.

Tabell 2: Möjligheter att arbeta med det centrala innehållet i matematik för årkurs 7-9 i undervisningsmaterialet till programmeringsverktyg (centrala innehåll från Skolverket 2017a, s. 60-61)

Det centrala innehållet	Blockprogrammeringsverktyg	Textbaserat programmeringsverktyg
Taluppfattning och tals användning	Till viss del: <ul style="list-style-type: none"> <li>– reella tal och deras egenskaper</li> <li>– talsystemets utveckling från naturliga tal till reella tal</li> </ul>	Till viss del: <ul style="list-style-type: none"> <li>– spel som utnyttjar tals egenskaper, till exempel gissa tal och träna multiplikation.</li> <li>– naturliga tal och reella tal <ul style="list-style-type: none"> <li>○ verktyget skiljer på naturliga och reella tal</li> </ul> </li> </ul>



		– beräkningar med digital teknik
Algebra	Alla delar i det centrala innehållet	Till viss del: – alla delar i det centrala innehållet förutom metoder för ekvationslösning som inte behandlas explicit i uppgifterna
Geometri	Till viss del: – alla delar i det centrala innehållet förutom satser och formler	Till viss del: – nästan alla delar i det centrala innehållet behandlas i materialet, dock inte beräkningar av volym och enhetsbyten. – geometriska satser och formler behandlas i liten omfattning.
Sannolikhet och statistik	Till viss del: – grafer och medelvärde	Till viss del: – beräkning av medelvärde – produktion av stapeldiagram – skapande och användning av slumpantal
Samband och förändring	Alla delar i det centrala innehållet	Till viss del: – procent för att uttrycka förändring – koordinatsystemet
Problemlösning	Alla delar i det centrala innehållet, dock behandlas inte strategier och värdering av strategier explicit	Alla delar i kursplanen, dock behandlas inte problemlösningstrategier och värdering av strategier explicit. Strategier för att testa och förbättra algoritmer undersöks.

Se *Bilaga 3 – Resultat av verktygsanalys* för en analys av alla delar av kursplanen som berörs i programmeringsverktygen och uppgifterna.

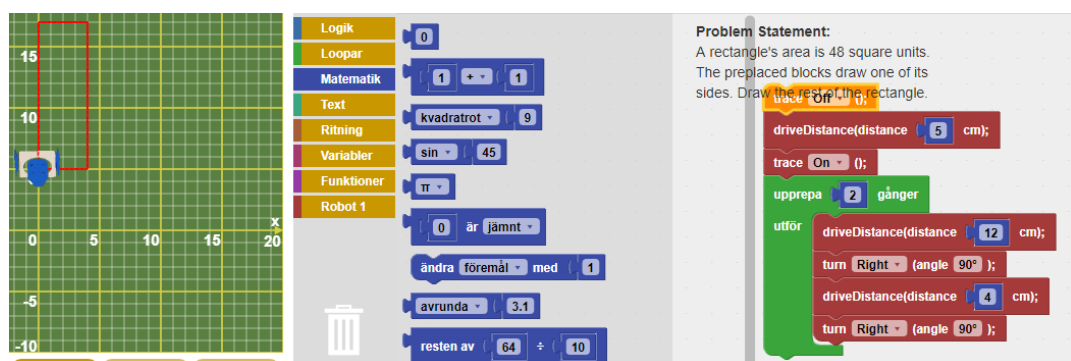
Det finns möjlighet att arbeta med nästan alla delar i det centrala innehållet i båda verktygen, men det finns inte färdiga uppgifter till detta. Till exempel i det textbaserade verktyget är tal i potensform, metoder för ekvationslösning, likformig sannolikhet, kombinatorik och bedömningar av risker och chanser områden från det centrala innehållet som skulle kunna undersökas med verktyget men som inte behandlas i undervisningsmaterialet.

Uppgifterna i de olika verktygen utnyttjar programmering på olika sätt. I blockprogrammeringsverktyget med integrerade matematikuppgifter används verktyget till stor del som ett visualiseringsverktyg och många av beräkningar förväntas utföras utanför verktyget. Verktyget erbjuder dock möjligheten att definiera nya variabler och därigenom utföra beräkningarna med verktyget. I det

textbaserade programmeringsverktyget används programmering mer genomgående som ett verktyg att utföra beräkningar och lösa matematiska problem.

Här presenteras några exempel på hur blockprogrammeringsverktyget med integrerade matematikuppgifter erbjuder möjligheter att arbeta med matematisk problemlösning.

Figur 4 visar ett problem inom geometri där eleverna uppmanas beräkna hur en rektangel ska ritas utifrån arean och längden på den ena sidan. Eleverna kan arbeta med att testa och förbättra algoritmer genom att till exempel utveckla deras program från att först skriva kod för att rita alla fyra sidor till att använda upprepa-funktionen för att skriva ett mer effektivt program. Uppgiften har stark klassifikation och inramning eftersom den har ett tydligt avgränsat matematiskt fokus och innehållet och kriterium för en korrekt lösning kontrolleras av verktyget.



Figur 4: Problemlösning i geometri i blockprogrammeringsverktyget (UC Davis 2017b, Grade 7 Uppgift 8)

Ett annat exempel på problem som kan undersökas med blockprogrammeringsverktyget med integrerade matematikuppgifter är inom funktioner (UC Davis, 2017c):

The 1st robot moves at the speed of 1.5 inches per second. After 8 seconds, the 2nd robot races at the speed of 3 inches per second. When and at what distance will the 2nd robot catch the 1st one?

**Typical Solution:**

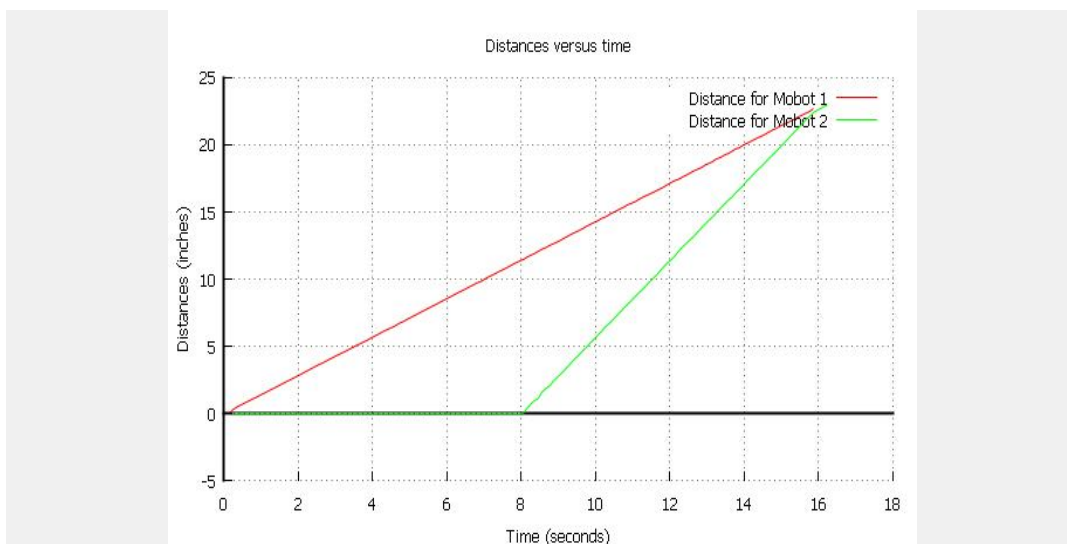
$$y = 1.5t$$

$$y = 3(t-8)$$

$$1.5*t = 3(t-8)$$

$$t = 16, y = 24$$

**Output from a Ch program:**



Figur 5: Problemlösningssuppgift i blockprogrammeringsverktyget (UC Davis, 2017c)

Blockprogrammeringsverktyget med integrerade uppgifter ger eleverna möjlighet att undersöka matematiska begrepp som hastighet och avstånd, formulera en matematisk frågeställning och arbeta med metoder för ekvationslösning med ett konkret problem. Inga fysiska robotar behövs eftersom robotarnas rörelser visas på skärmen. Problemet som visas i Figur 5 har stark klassifikation med mycket fokus på matematik och svag inramning, eftersom lärare och elever kan arbeta med problemet på olika sätt och utveckla andra problem som kan undersökas med liknande strategier.

Ett exempel på en möjlighet till arbete med matematisk problemlösning som erbjuds i det textbaserade programmeringsverktyget visas i uppgiften i Figur 6, som behandlar geometri och algebra. Eleverna kan börja genom att rita en kvadrat och därefter utveckla och förbättra algoritmen så att den blir mer generell.

## Rita en polygon

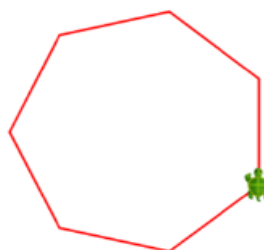
### Uppdrag:

- Prova koden nedan. Rita olika slags polygoner.
- Lägg till en parameter *sidlängd* och rita olika stora polygoner.
- Hur stort behöver *n* vara för att det ska se ut som en cirkel?

### Tips:

```
def polygon(n:Heeltal) = upprepa(n){
  fram(100)
  vänster(360.0/n)
}

sudda; sakta(100)
polygon(7)
```



Figur 6: Geometri i textbaserat programmeringsverktyg (Regnell 2016, s. 15)

Många av uppgifterna till det textbaserade verktyget börjar med en introduktion till ett kommando eller en funktion som följs av en fri utforskning av verktyget för att undersöka på en djupare nivå hur kommandon kan användas. Problemet i Figur 7 visar hur verktyget erbjuder arbete med samband och förändring, algebra, beräkningar med digital teknik och enkla matematiska modeller:

This activity involves the following:

- Learning to create new functions.
- Playing with *simple interest* to make a figure.

### Step 1

Type in the following code and run it using the *Run as Worksheet* button:

```
// Simple interest exploration
// P is the Principal
val P = 100.0
// R is the rate of interest
val R = 10.0
// si is a function that lets you calculate simple interest
// given a principal and a rate of interest
def si(p: Double, r: Double) = p * r / 100
// amt is a function that tells you the amount after interest
// given a principal, a rate of interest, and the number of years
def amt(p: Double, r: Double, t: Int) = {
  p + si(p, r) * t
}
amt(P, R, 0)
amt(P, R, 1)
amt(P, R, 2)
```

Q1a What do you think the code above does?

Q1b Looking at the results of the code, can you tell what the *amount* is after 2 years?

Figur 7: Programmeringsproblem i textbaserat programmeringsverktyg (Pant 2015, s. 82)

Inramning av innehållet vid programmering för matematisk problemlösning i det textbaserade programmeringsverktyget kontrolleras till stor del av läraren och deras val av uppgifter och till viss del av hur eleverna använder verktyget och är därför svag.

### 5.2.5 Arbetsordningen vid integrering av programmering för matematisk problemlösning

I blockprogrammeringsverktyget med integrerade matematikuppgifter är graden av inramning av arbetsordning stark om ordningen av de integrerade uppgifterna följs. Uppgifterna till det textbaserade programmeringsverktyget presenteras i en ordning som gör att eleverna lär sig nya kommandon och datalogiska koncept som variabler och kan bygga vidare på de tidigare övningarna för att gå vidare. Det finns dock möjlighet med båda verktygen för lärare och elever att bestämma ordningen själva och att skapa andra uppgifter. Detta ger en svag inramning.

### 5.2.6 Tempot vid integrering av programmering för matematisk problemlösning

Tempot vid arbete med båda programmeringsverktygen bestäms av användaren och därför är graden av inramning av tempot svag.

### 5.2.7 Kriterier för acceptabelt arbete vid integrering av programmering för matematisk problemlösning

På samma sätt som i analysen av lärarintervjuerna kan graden av inramning av kriterier för acceptabelt arbete analyseras på två sätt, genom att beakta programmeringskoden eller den matematiska lösningen.

Programmeringsverktygen kan bedöma att programmeringskoden är oacceptabel om den innehåller syntaktiska fel som inte kan tolkas av datorn. Detta gäller för båda

verktygen. Det är dock lättare att undvika syntaktiska fel i blockprogrammeringsverktyget. Från ett programmeringsperspektiv är inramningen av kriterierna för acceptabelt arbete stark i det textbaserade verktyget eftersom kommandon måste skrivas precist för att programmet ska fungera som det är tänkt. Det finns möjlighet att köra koden steg för steg för att felsöka i programmet. Om det finns ett problem med koden visas ett felmeddelande, som ibland är svårt att förstå. Uppgifterna visar oftast ett exempel på hur koden ska se ut vilket kan tänkas bidra till elevernas möjligheter att felsöka själv och reflektera på varför deras kod inte fungerar. Inramningsgraden kring kriterierna i programmeringskoden analyseras som stark på grund av de osynliga reglerna kring hur koden ska skrivas.

Lösningen till det matematiska problemet kan också bedömas som acceptabel eller inte. Eleven kan ha skrivit programmeringskod som är syntaktiskt acceptabelt för verktyget men som inte löser det matematiska problemet på ett tillfredställande sätt. I blockprogrammeringsverktyget med integrerade uppgifter inkluderas ett förslag till en lösning för varje uppgift. Det finns endast ett exempel på en korrekt lösning. Det kan dock finnas flera sätt att komma fram till lösningen. Om användaren kör en kod som misslyckas med att lösa problemet på det önskade sättet presenteras en ruta som förklarar att man har misslyckats. Blockprogrammeringsverktyget med integrerade matematikuppgifter har därför en hög grad av inramning med avseende på kriterier för acceptabla matematiska lösningar. Om verktyget används för programmering för matematisk problemlösning med andra problem som skapas av läraren eller elever är graden av inramning svag eftersom verktyget inte vet vad den förväntade lösningen är och ger därför ingen respons till användaren. I det textbaserade programmerings-verktyget är inramningen svag med avseende på kriterier för acceptabla matematiska lösningar, eftersom det finns många problem där många olika sätt att lösa problemet skulle accepteras. Många av uppgifterna uppmanar till utforskning av verktyget och användning av programmering som verktyg att lösa matematiska problem.

### **5.2.8 Sammanfattning av möjligheter till arbete med matematisk problemlösning i programmeringsverktyg**

Blockprogrammeringsverktyget med integrerade matematikuppgifter erbjuder många uppgifter med tydlig matematisk koppling. Genom att strikt följa uppgifterna erbjuds matematikundervisning med en hög grad av klassifikation och inramning. Verktöget erbjuder dock lärare och elever möjligheter att anpassa innehåll, ordning och arbetstempo och kan därför användas för att skapa en undervisning med hög klassifikation och matematisk fokus samt en svag inramning och möjligheter till utforskning av matematiska problem.

Det textbaserade programmeringsverktyget erbjuder möjligheter att använda programmering för att lösa olika matematiska problem. Många av de tillhörande programmeringsproblemen innehåller matematik som ingår i det centrala innehållet. Uppgifterna och verktyget erbjuder en svag grad av klassifikation och svag inramning. Verktöget erbjuder lärare och elever möjligheter att anpassa innehåll, ordning och arbetstempo. Verktöget kan därför användas för att skapa en undervisning med en svag inramning. Det erbjuder stora möjligheter att arbeta med matematisk problemlösning men matematiska kopplingar måste skapas av läraren.

## 6 Diskussion

I detta avsnitt diskuteras studiens metod och resultat.

### 6.1 Metoddiskussion

Metoden för undersökning och analys i denna studie har inneburit att ett urval har gjorts av både skolor och intervjupersoner, och även av programmeringsverktyg. Urval av skolorna har varit godtyckligt. Åtminstone en lärare som uppfyllde kriterierna för att delta i studien valde att inte delta på grund av tidsbrist. Det kan vara så att lärare som inte har kunskaper om programmering och därigenom exkluderas från studien skulle kunna ha bidragit på något sätt ändå. De lärare som har kunskaper om programmering kan naturligtvis tänkas vara mer positivt inställda till integrering av programmering i matematikundervisning. Det är möjligt att andra nya insikter skulle ha kunnat komma fram med flera och andra deltagare men det insamlade material gav ändå ett brett underlag som kunde analyseras för att besvara studiens syfte.

En intervjuguide formulerades utifrån studiens syfte och val av teori. Lärare fick i förväg tillgång till intervjuguiden (*Bilaga 2 – Intervjuguide*) för att kunna fundera på relevanta erfarenheter och kunna svara på ett utvecklat sätt vid intervjutillfällena. Det kan dock vara så att lärarna har förberett svar som de tror är önskvärda. Det är även viktigt att undvika ledande frågor (Kvale & Brinkmann 2009, s. 263). Genom att ställa ledande frågor kan intervjuaren påverka reliabilitet på ett negativt sätt. En fråga som kan tänkas vara ledande är den som handlar om de olika matematiska områden som programmering kan tänkas användas i. Denna fråga förutsätter på ett sätt en stark klassificering inom matematikämnet mellan olika ämnesområden. Detta motiveras utifrån den uppdelning som finns i det centrala innehållet för matematik i läroplanen (Skolverket 2017a, s. 60-61). Det fanns ändå möjligheter för lärare att lyfta möjligheter att arbeta med flera områden samtidigt och diskutera övergripande matematiska förmågor. Frågan bedömdes vara nödvändig för att kunna få syn på lärares tankar kring programmering för matematisk problem-lösning med anseende på intern klassifikation inom matematikämnet.

Valet av programmeringsverktyg gjordes utifrån kriterier som beskrivs ovan för att på mest relevant sätt kunna besvara studiens syfte. Digital teknik utvecklas fort och nya versioner och verktyg utvecklas fortlöpande. Det är möjligt att andra programmeringsverktyg erbjuder andra möjligheter att arbeta med matematisk problemlösning och skulle ha gett andra resultat.

### 6.2 Resultatdiskussion

Syftet med studien var att undersöka hur programmering för matematisk problemlösning kan integreras i matematikundervisning i årskurs 7-9. Resultaten diskuteras utifrån syftet, styrdokument och forskning. Resultaten från de två frågeställningarna diskuteras även i relation till varandra.

#### 6.2.1 Hur anser lärare att programmering för matematisk problemlösning kan integreras i matematikundervisning?

Från intervjuerna med lärare framkom det två huvudsyften med integrering av programmering för problemlösning i matematikundervisning. Det första är som ett verktyg för att kunna lösa matematiska problem, det andra är som ett sätt för

eleverna att få en ökad medvetenhet om programmering i det digitala samhället och därigenom en ökad digital kompetens. Dessa två syften kompletterar varandra. Båda dessa syften lyfts av Skolverket i styrdokumentet. Skolverket förklarar att programmering ”har stora likheter med generell problemlösning” (Skolverket 2017d, s. 10) och eleverna ska ”kunna använda programmering som ett verktyg i matematik” (Skolverket 2017c, s. 17) men även att ”programmering ska ses i ett vidare perspektiv som även omfattar kreativt skapande, styrning och reglering, simulering samt demokratiska dimensioner” (Skolverket 2017d, s. 10). Att kunna förstå att det är människor som programmerar datorer för att skapa innovativa lösningar som begränsas enbart av våra fantasier är en del av datalogisk tänkande (Wing 2006, s. 35) och något som lyftes av lärare som ett syfte med integrering av programmering för matematisk problemlösning i matematikundervisning.

Lärare lyfter fram kalkylblad som en programmeringsmiljö som med fördel kan användas för att utveckla ett datalogisk tänkande hos elever. Många av idéerna som kan utforskas i ett kalkylblad som variabler och funktioner kan sedan användas i arbete i andra programmeringsmiljöer, något som beskrivs även av Berry (2017, s. 51). Logiskt tänkande och datalogiskt tänkande är något som lärare hoppas att eleverna utvecklar genom arbete med programmering. Lärare lyfter vikten av att skapa broar mellan programmering och annan matematisk problemlösning och underlättar för eleverna att se likheter mellan dem. Detta kan vara en klok strategi eftersom forskning visar att det kan vara svårt för eleverna att se likheter mellan olika problem och kunna överföra strategier mellan problem (Mayer & Wittrock 1996, s. 55). Lärare menar också att programmering ska användas som ett verktyg för matematisk problemlösning när det är det mest effektiva sättet att angripa problemet. Eleverna ska få tillgång till programmering som ett verktyg och även kunna värdera olika strategier för att lösa problem. Eleverna ska lära sig att analysera matematiska problem och välja mellan miniräknare, kalkylblad, blockprogrammering och textbaserad programmering som verktyg för att lösa problemet.

Lärare förklarar hur integrering av programmering för matematisk problemlösning kan vara ett stöd för elever i utvecklingen av deras matematiska resonemangsförmåga. De kan även utveckla sin förmåga att skriftligt uttrycka matematisk resonemang eftersom de tvingas att stegvis beskriva en lösning till ett problem för att kunna skriva programmeringskod. Detta kan utnyttjas av lärare för att få syn på elevernas resonemangsförmåga och problemlösningsförmåga. Att eleverna utvecklade sin matematiska resonemangsförmåga i arbete med programmering är något som beskrivs i studier av Calder (2010, s. 13-14), Feurzeig et al. (2011, s. 490, 495) och Ratcliff och Anderson (2011, s. 251). Även Palmér (2015, s. 377) fann att en svag inramning i matematikappar för förskolebarn ledde till ökat elevdeltagande och en utveckling av resonemangsförmåga.

Programmering som ett visualiseringsverktyg lyfts också av lärare. Detta överensstämmer med det syfte Feurzeig et al. (2011, s. 490) hade med programmering i LOGO som ursprungligen presenterades 1968. Utveckling av elevernas visualiseringsförmåga efter arbete med programmering beskrivs även i studier av Ardito et al. (2014, s. 79), Calder (2010, s. 12) och Ratcliff och Anderson (2011, s. 249).

Lärare efterlyser tydligare råd kring bedömning av programmeringsarbete. Den bredare syn på programmering i samhället och att eleverna kan se nyttan med programmering i vardagslivet finns inte explicit uttryckt i kunskapskraven. Lärare menar att det är viktigt att utgå ifrån kunskapskraven och att ha ett tydligt mål med arbetet. Att lärare och elever förstår lärandemålen med undervisningen är en viktig faktor för att skapa en effektiv undervisning (Hattie 2009, s. 320). I arbete med programmering för matematisk problemlösning är det därför viktigt som lärare att fundera över kriterierna för acceptabelt arbete i programmering för matematisk problemlösning och hur det presenteras för elever. Eleverna kanske inte behöver lösa problemet i sin helhet för att ha lyckats med arbetet om det är utvecklande av en förmåga som är målet.

Flera lärare påstår att programmering med textbaserade verktyg kan vara för svår och tidskrävande för vissa elever, något som stöds av Rolandssons resultat (2015, s. 60). Okita (2014, s. 858) fann dock att elever i åldern 9-11 år som hade lärt sig programmera med ett textbaserat verktyg hade lättare att föra över kunskaper till nya miljöer och problem än de som hade lärt sig att programmera med ett blockprogrammeringsverktyg. Detta kan ha viktiga implikationer för valet av programmeringsverktyg på grundskolan, särskilt med tanke på att de intervjuade lärarna menar att blockprogrammeringsverktyg är det lämpligaste valet för programmering i årskurs 7-9.

En viktig faktor som kan påverka lärares möjligheter i val av programmeringsverktyg är de digitala plattformarna som eleverna har tillgång till. Lärare påpekar att många programmeringsverktyg som skulle kunna användas för integrering av programmering för matematisk problemlösning fungerar inte på surfplattor eller Ipads som eleverna har tillgång till eftersom de kräver nerladdningar. Detta begränsar lärares val till webbaserade verktyg.

### **6.2.2 Vilka möjligheter till arbete med matematisk problemlösning erbjuder programmeringsverktyg?**

Analysen av programmeringsverktygen visar att programmeringsverktyg erbjuder många möjligheter till arbete med matematisk problemlösning. Den matematiska fokusen styrs till stor grad av innehållet i uppgifterna. Programmering har visats tidigare att kunna användas i samband med matematisk problemlösning för att eleverna ska utveckla sin begreppsförståelse (Ardito et al., 2014; Aydin, 2005; Calder, 2010; Feurzeig et al., 2011; Grubbs, 2013; Li et al., 2014; Martínez Ortiz, 2015). Många delområden i det centrala innehållet i kursplanen för matematik i årskurs 7-9 kan undersökas med de analyserade programmeringsverktygen. Verktygen erbjuder möjlighet till att skapa en undervisning med stark fokus på matematik och en svag inramning. Palmér (2015, s. 376) fann att elever som arbetade med appar med stark klassifikation och svag inramning verkade vara mer delaktiga men även att arbete med dessa appar krävde mer återkoppling och stöd från läraren.

Blockprogrammeringsverktyget med integrerade matematikuppgifter erbjuder en stark klassifikation och hög grad av inramning av innehåll i matematikuppgifterna som finns integrerade i verktyget. Uppgifterna i denna undersökning är dock på engelska och ibland komplicerade. För att kunna använda verktyget i en svensk klass



måste läraren översätta uppgifterna. En möjlighet skulle vara att utveckla liknande uppgifter för ett annat blockprogrammeringsverktyg som har en mer utvecklad svensköversättning och funktionalitet.

En av styrkorna med blockprogrammeringsverktyget som undersöks, ur matematikperspektiv, är koordinatrutnätet som hjälper eleverna att visualisera matematiken de arbetar med. Denna möjlighet finns även i andra programmeringsverktyg fast funktionaliteten inte är lika tydlig. Många av uppgifterna i blockprogrammeringsverktyget utnyttjar inte programmering som verktyg att lösa matematiska problem utan använder programmeringsverktyget mer som ett verktyg för att kunna visualisera olika lösningar. I uppgifterna till det textbaserade verktyget utnyttjas programmeringskraften genom att skapa och arbeta med variabler för beräkningar och utforskning av mönster på ett mer autentiskt sätt som liknar programmering utanför skolan. Det kan vara en styrka att kombinera arbeta med flera olika programmeringsverktyg i undervisning eftersom de ger olika perspektiv och utmaningar.

### **6.2.3 Samband mellan resultat av intervjuerna och verktygsanalysen för att få svar på hur programmering för matematisk problemlösning kan integreras i matematikundervisning**

Lärare anser att det finns möjlighet att integrera programmering i alla delar av det centrala innehållet, något som stöds av resultaten från analysen av programmeringsverktygen. Vissa områden inom matematik som geometri och funktioner lyfts både av lärare och i verktygsanalysen som särskilt lämpliga för arbete med programmering för matematisk problemlösning. En potentiell styrka av programmering som verktyg för matematisk problemlösning men också en svårighet för lärare som saknar programmeringskompetens är att båda programmeringsmiljöerna bedöms tillåta återskapande av alla de uppgifter som undersöktes i studien. En skillnad är att uppgifterna och möjliga korrekta lösningar finns inbyggda i blockprogrammeringsverktyget.

Programmeringsverktygen utan integrerade uppgifter har en svag klassifikation och inramning och det är lärarens val av lämpliga matematiska innehåll som leder till att fokuset blir på matematik. För att skapa en effektiv undervisning som integrerar programmering för matematisk problemlösning i matematik-undervisning måste läraren ha tillräckligt djupa programmeringskunskaper för att kunna välja ett lämpligt verktyg och matematiskt innehåll. Detta stöds av Mishra och Koehlers (2006, s. 1030) arbete kring Technological Pedagogical Content Knowledge (TPCK). De menar att lärare behöver inte bara ha ämneskunskaper men även kunskaper om hur ämnesinnehållet kan förändras genom användning av teknik. Mishra och Koehler (2006, s. 1030) påpekar att det inte finns någon universal lösning utan att lärare måste ansvara för att skapa undervisning som är anpassad till den specifika undervisningskontexten.

Lärare har olika förslag på hur de praktiskt skulle integrera programmering för matematisk problemlösning i matematikundervisning. En stark inramning med styrda uppgifter som kontrollerar arbetsordning och minimerar risken för att eleverna fastnar föredras av vissa. Denna sorts undervisningsplanering med stark inramning stöds av ett verktyg med integrerade matematikuppgifter. Palmér (2015,

s. 376), i sin studie av Ipad matematikappar för förskolan, fann att appar med stark inramning och klassifikation verkade minska elevernas delaktighet. Detta ledde till att eleverna använde en form av trial and error för att hitta rätt svar utan någon utveckling av matematisk förståelse eller resonemang. Palmér (s. 377) menar att elever fick få möjligheter till reflektion i arbete med appar med stark inramning. Andra intervjuade lärare menar att eleverna får en ökad motivation och utvecklar en djupare förståelse genom att utforska och upptäcka själva mot ett bestämt mål, i en undervisning med en svag inramning. Denna sorts undervisningsplanering liknar det som presenteras i undervisningsmaterialet till det textbaserade programmeringsverktyget. Ratcliff och Andersen (2011, s. 247) varnar för att fria utforskningar utan mål kan leda till att elever fastnar utan att förstå syftet med programmeringsarbetet. Undervisning med en svag inramning kräver mer av läraren i form av programmeringskunskaper, delaktighet i kommunikation kring arbetet och en hög nivå av medvetenhet om syftet med undervisningsmomentet för att försäkra att eleverna får möjlighet att utveckla de tilltänkta matematiska förmågorna.

#### **6.2.4 Avslutande reflektioner**

Introduktionen av programmering för matematisk problemlösning i kursplanen för matematik ses av lärare som en positiv förändring som kan öka elevernas digitala kompetens, skapa ökad relevans i matematikundervisning och rusta eleverna med ett nytt digitalt verktyg för att kunna lösa matematiska problem. Flera programmeringsverktyg som erbjuder möjligheter att utforska stora delar av det centrala innehållet finns tillgängliga. Lärare ser stor potential i att integrera programmering för matematisk problemlösning i matematikundervisning som ett verktyg för problemlösning och även som ett stöd till elever i visualisering av matematiskt arbete och i utvecklandet av resonemangsförmågan.

Lärare efterlyser mer stöd och en ökad inramning av innehållet för att kunna skapa en effektiv undervisning som integrerar programmering för matematisk problemlösning i matematikundervisning. Det finns dock en potentiell konflikt mellan en ökad grad av inramning av innehållet för programmering för matematisk problemlösning och de möjligheter som lyfts av lärarna. Integrering av programmering för matematisk problemlösning ger läraren möjligheten att skapa en undervisningsmiljö med en stark matematisk fokus och en svag inramning, något som kan leda till ökad elevdelaktighet och utveckling av elevernas matematiska förmågor. En ökad grad av inramning skulle underlätta planering av undervisning för lärare med bristande programmeringskunskap men förminska möjligheter till en kreativ undervisning med programmering som ett verktyg för problemlösning i matematik. Programmering för matematisk problemlösning erbjuder stora möjligheter men lärare måste utrustas med de kunskaper som krävs för att kunna integrera det i en dynamisk och effektiv matematikundervisning.

## **7 Förslag på framtida forskning**

Intervjuade lärare med mycket erfarenhet av programmering utanför skolan hade många förslag på hur programmering för matematisk problemlösning kan integreras i matematikundervisning. Antal deltagare är för få för att kunna dra generaliserbara slutsatser men det skulle vara intressant att studera vilka faktorer som påverkar lärares val i hur de integrerar programmering för matematisk problemlösning i matematikundervisning.

Lärare lyfter möjligheterna för eleverna att visa sin resonemangsförmåga genom programmering för matematisk problemlösning. Det skulle vara intressant att undersöka om ett medvetet arbete med att skapa kopplingar mellan programmering för matematisk problemlösning och elevernas förmågor att kommunicera sitt matematiska tänkande leder till en utveckling av elevernas resonemangsförmåga eller kommunikationsförmåga i andra delar av matematik.

Okita (2014, s. 858) fann att valet av det första programmeringsverktyget för eleverna påverkade deras förmåga att utveckla och överföra programmeringskunskaper vid senare tillfällen. Det skulle vara intressant att forska vidare kring en jämförelse av blockprogrammeringsverktyg och textbaserade verktyg utifrån utvecklingen av elevernas förmåga att använda programmering för matematisk problemlösning under en längre tid och på olika nivåer.

Det kräver förstås mycket noggrann planering och förmodligen mycket tid för att skapa en integrerad undervisning i programmering för matematisk problemlösning med den pedagogiska miljön som läraren anser mest lämplig. Läraren måste ta beslut för att skapa den inramning och klassifikation som de vill ha i undervisningen. Mer forskning krävs för att undersöka vilka kombinationer av klassifikation och inramning som ska väljas för att eleverna bäst ska kunna utveckla sina matematiska förmågor när de arbetar med programmering för matematisk problemlösning.

## Litteratur

- Ardito, G., Mosley, P. & Scollings, L. (2014). WE, ROBOT. Using Robotics to Promote Collaborative and Mathematics Learning in a Middle School. *Middle Grades Research Journal*, 9(3), 2014, 73–88.
- Aydin, E. (2005). The Use of Computers in Mathematics Education: A Paradigm Shift from “Computer Assisted Instruction” towards “Student Programming”. *The Turkish Online Journal of Educational Technology*, (4)2, 27-34.
- Bernstein, B. (2000). *Pedagogy, symbolic control and identity – theory, research, critique*. Oxford: Rowman & Littlefield Publishers, Inc.
- Berry, M. (2017). *QuickStart Computing – Subject Knowledge Enhancement for secondary teachers*. Swindon: BCS. Tillgänglig: [community.computingschool.org.uk/files/8213/original.pdf](http://community.computingschool.org.uk/files/8213/original.pdf)
- Calder, N. (2010). Using Scratch: An Integrated Problem-solving Approach to Mathematical Thinking. *Australian Primary Mathematics Classroom*, (15)4, 9-14. Tillgänglig: <http://files.eric.ed.gov/fulltext/EJ906680.pdf>
- Castledine, A.-R. & Chalmer, D. (2011). LEGO Robotics: An authentic problem solving tool? *Design and Technology Education: An International Journal*, (16)3, 19-27.
- CodeCombat (2017). *CodeCombat - Learn how to code by playing a game*. Hämtad 2017-10-09, från <https://codecombat.com/>
- Feurzeig, W., Papert, S. & Lawler, B. (2011). Programming-languages as a conceptual framework for teaching mathematics. *Interactive Learning Environments*, (19)5, 487-501. <http://dx.doi.org/10.1080/10494820903520040>
- Garofalo, J. & Lester, F. K. Jr. (1985). Metacognition, cognitive monitoring, and mathematical performance. *Journal for Research in Mathematics Education*, 16(3), 163-176. <https://doi.org/10.2307/748391>
- Grubbs, M. (2013). Robotics Intrigue Middle School Students and Build STEM Skills. *Technology and Engineering Teacher*, (72)6, 12-16.
- Hattie, J. (2009). *Synligt lärande. En syntes av mer än 800 metaanalyser om vad som påverkar elevers skolresultat*. Stockholm: Natur & Kultur.
- Hubweiser, P., Giannakos, M. N., Berges, M., Brinda, T., Diethelm, I., Magenheimer, J., ... Jasute, E. (2015). A Global Snapshot of Computer Science Education in K-12 Schools. *Transactions on Computing Education*. <http://dx.doi.org/10.1145/2858796.2858799>
- Högskolan Dalarna (2008). *Forskningssetiska anvisningar för examens- och uppsatsarbeten vid Högskolan Dalarna*. Högskolan Dalarna.
- Kjällander, S., Åkerfeldt, A. & Petersen, P. (2016). *Översikt avseende forskning och erfarenheter kring programmering i förskola och grundskola*. Hämtad 2017-09-12, från [http://omvarld.blogg.skolverket.se/wp-content/uploads/sites/2/2016/06/oversikt\\_programmering\\_i\\_skolan.pdf](http://omvarld.blogg.skolverket.se/wp-content/uploads/sites/2/2016/06/oversikt_programmering_i_skolan.pdf)
- Kvale, S. & Brinkmann, S. (2009). *Den kvalitativa forskningsintervjun*. Lund: Studentlitteratur.
- Larkin, K. (2014). iPad apps that promote mathematical knowledge? “Yes, they exist!” *Australian Primary Mathematics Classroom*, 19(2), 28-32. Tillgänglig: <http://files.eric.ed.gov/fulltext/EJ1093348.pdf>
- Larsen, A. K. (2009). *Metod helt enkelt*. Malmö: Gleerups.

- Li, Q., Vandermeiden, E., Lemieux, C. & Nahoo, S. (2014). Secondary Students Learning Mathematics Through Digital Game Building: A Study Of The Effects And Students' Perceptions. *International Journal of Technology in Mathematics Education*, (23)1, 25-34.
- Lunds Tekniska Högskola (u.å.). *Lär dig programmera*. Hämtad 2017-10-19, från <http://www.lth.se/programmera/programming-i-skolan/>
- Martínez Ortiz, A. (2015). Examining Students' Proportional Reasoning Strategy Levels as Evidence of the Impact of an Integrated LEGO Robotics and Mathematics Learning Experience. *Journal of Technology Education*, 26(2). <https://doi.org/10.21061/jte.v26i2.a.3>
- Mayer, R. E. & Wittrock, M. C. (1996). Problem-solving transfer. I D. C. Berliner & R. C. Calfee (Red.), *Handbook of educational psychology* (s. 47-62). New York: Routledge.
- Mishra, P. & Koehler, M. (2006). Technological Pedagogical Content Knowledge: A Framework for Teacher Knowledge. *Teachers College Record*, 108(6), 1017-1054.
- Möllerhed, E. (2001). *Problemlösning i matematik. En studie av påverkansfaktorer i årskurserna 4-9*. Malmö: Institutionen för pedagogik, Lärarhögskolan i Malmö.
- Okita, S. (2014). The relative merits of transparency: Investigating situations that support the use of robotics in developing student learning adaptability across virtual and physical computing platforms. *British Journal of Educational Technology* 45(5), 844–862. <https://doi.org/10.1111/bjet.12101>
- Palmér, H. (2015). Using tablet computers in preschool: How does the design of applications influence participation, interaction and dialogues? *International Journal of Early Years Education*, 23(4), 365-381. <http://dx.doi.org/10.1080/09669760.2015.1074553>
- Palmér, H. & Helenius, O. (2016). *Analys av digitala programvaror*. Stockholm: Skolverket. Tillgänglig: [https://larportalen.skolverket.se/webcenter/larportal/api-v2/document/path/larportalen/material/inriktningar/0-digitalisering/Grundskola/436\\_matematikundervisningmeddigitalaverktyg-%C3%A5k7-9/5\\_appariundervisningen/material/flikmeny/tabA/Artiklar/IKT7-9\\_05A\\_01\\_analys\\_av\\_digitala\\_programvaror.docx](https://larportalen.skolverket.se/webcenter/larportal/api-v2/document/path/larportalen/material/inriktningar/0-digitalisering/Grundskola/436_matematikundervisningmeddigitalaverktyg-%C3%A5k7-9/5_appariundervisningen/material/flikmeny/tabA/Artiklar/IKT7-9_05A_01_analys_av_digitala_programvaror.docx)
- Pant, L. (2015). *Explorations in Math, Art, Programming, Learning, and Science with Kajo*. Hämtad 2017-10-05, från <http://www.kogics.net/kajo-ebooks#explorations>
- Ratcliff, C., & Anderson, S. (2011). Reviving the Turtle: Exploring the Use of Logo with Students with Mild Disabilities. *Computers in the Schools*, (28)3, 241-255. <http://dx.doi.org/10.1080/07380569.2011.594987>
- Regnell, B. (2016). *Uppdrag med Kajo*. Lund: Lund University. Tillgänglig: [http://fileadmin.cs.lth.se/cs/Personal/Bjorn\\_Regnell/uppdrag.pdf](http://fileadmin.cs.lth.se/cs/Personal/Bjorn_Regnell/uppdrag.pdf)
- Regnell, B. (2015). *Lär dig programmera – Bakgrund*. Hämtad 2017-10-19, från <http://www.lth.se/programmera/bakgrund/>
- Rolandsson, L. (2015). *Programmed or not. A study about programming teachers' beliefs and intentions in relation to curriculum*. Stockholm: KTH. Tillgänglig: <http://kth.diva-portal.org/smash/get/diva2:791197/FULLTEXT02.pdf>

- Skolverket (2017a). *Läroplanen för grundskolan, förskoleklassen och fritidshemmet. Reviderad 2017*. Stockholm: Skolverket.
- Skolverket (2017b). *Tydligare om digital kompetens*. Hämtad 2017-09-12, från <https://www.skolverket.se/skolutveckling/resurser-for-larande/itiskolan/styrdokument>
- Skolverket (2017c). *Kommentarmaterial till kursplanen i matematik. Reviderad 2017*. Stockholm: Skolverket.
- Skolverket (2017d). *Få syn på digitaliseringen på grundskolenivå – Ett kommentarmaterial till läroplanerna för förskoleklass, fritidshem och grundskoleutbildning*. Stockholm: Skolverket.
- Skolverket (2016). *Redovisning av uppdraget om att föreslå nationella it-strategier för skolväsendet – förändringar i läroplaner, kursplaner, ämnesplaner och examensmål*. Stockholm: Skolverket.
- Snell, C. (2017). *Programmering för problemlösning i matematik*. Falun: Högskolan Dalarna. Tillgänglig: <http://www.diva-portal.org/smash/get/diva2:1113850/FULLTEXT01.pdf>
- SOU 2015:28. *Gör Sverige i framtiden – digital kompetens*. Stockholm: Fritzes Offentliga Publikationer.
- Thurfjell, K. (2017, 2 sep). Sju av tio lärare i programmering saknar utbildning. *Svenska Dagbladet*. Hämtad 2017-09-08, från <https://www.svd.se/larare-som-ska-undervisa-i-programmering-saknar-kunskap>
- UC Davis Center for Integrated Computing and STEM Education (C-STEM) (2017a). *RoboBlockly for Learning Coding and Math with Robots*. Hämtad 2017-10-04, från <http://roboblockly.ucdavis.edu/about.php>
- UC Davis Center for Integrated Computing and STEM Education (C-STEM) (2017b). *RoboBlockly for Learning Coding and Math with Robots*. Hämtad 2017-10-04, från <http://roboblockly.ucdavis.edu/math7/m8.php>
- UC Davis Center for Integrated Computing and STEM Education (C-STEM) (2017c). *Mathematics*. Hämtad 2017-10-04, från <http://c-stem.ucdavis.edu/program/math/>
- Vetenskapsrådet (2002). *Forskningsetiska principer inom humanistisk-samhällsvetenskaplig forskning*. Vetenskapsrådet.
- Widerberg, K. (2002). *Kvalitativ forskning i praktiken*. Lund: Studentlitteratur.
- Wing, J. M. (2006). Computational Thinking. *COMMUNICATIONS OF THE ACM*, 49(3), 33-35. Tillgänglig: <http://marvin.cs.uidaho.edu/Teaching/K12-CS/computationalThinkingWing.pdf>

## Bilaga 1 – Informantbrev

*Information om undersökning om hur lärare anser programmering för matematisk problemlösning kan integreras i matematikundervisning i årskurs 7-9.*

*Du tillfrågas härmed om deltagande i denna undersökning.*

Hej!

Jag heter Charis Snell och läser på ämneslärarprogrammet på Högskolan Dalarna. Jag ska genomföra en undersökning kring hur programmering kan integreras i matematikundervisning, med tanke på de förändringar i läroplanen som nu genomförs. Syftet med studien är att undersöka hur programmering för problemlösning kan integreras i matematikundervisning i årskurs 7-9. Jag vill därför undersöka hur lärare anser att programmering för matematisk problemlösning kan integreras i matematikundervisning.

Jag planerar att intervjua cirka fem lärare som har erfarenhet av programmering. Jag är väldigt tacksam om du kan bidra med information till undersökningen genom att ställa upp på en intervju. Även om du inte har direkt erfarenhet av programmering inom matematikundervisning är dina tankar ändå väldigt värdefulla för min studie. Intervjun beräknas ta ungefär en halvtimme och kommer att spelas in för att kunna analysera innehållet vid ett senare tillfälle. Jag har planerat att, förutom att intervjua lärare, undersöka tre programmeringsverktyg och analysera dem utifrån det centrala innehållet för matematik i årskurs 7-9.

Ditt deltagande i undersökningen är helt frivilligt. Du kan när som helst avbryta ditt deltagande utan närmare motivering. Om du väljer att avsluta din medverkan i studien kommer informationen som du har bidragit med inte att användas i studien. Ditt, skolans och kommunens namn behålls anonymt. Du får tillgång till transkriberingen av intervjun för att kontrollera transkriberings- och tolkningsarbete. Enbart jag, handledaren och examinatorn på Högskolan Dalarna kommer att ha tillgång till intervjumaterialet. Undersökningen kommer att presenteras i form av en uppsats vid Högskolan Dalarna. Efter godkännande av arbetet kommer intervjumaterialet att förstöras. Du kommer att få tillgång till den färdiga uppsatsen.

Tack på förhand!

Med vänliga hälsningar,

Charis

Ytterligare upplysningar lämnas av nedanstående ansvariga:

Charis Snell <i>Student</i> Mobil: Epost: <a href="mailto:v12chasn@du.se">v12chasn@du.se</a>	Eva-Lena Erixon <i>Handledare</i> Telefon: 023 77 84 28 Epost: <a href="mailto:eer@du.se">eer@du.se</a>
--	---

## Bilaga 2 – Intervjuguide

### Intervjuguide

Under intervjun skulle jag vilja få din tankar kring följande ämnen:

Kan du berätta för mig om när du har jobbat med programmering i skolan?

Vilka programmeringsverktyg har du använt?

Vilka matematiska områden har ni berört i användningen av dem?

I vilka av de olika matematiska ämnesområdena tänker du att programmering i matematik skulle kunna användas? Varför?

Finns det något område som du inte skulle jobba med programmering i? Varför?

Hur förstår du ”problemlösning i matematik”?

Hur tänker du kring begreppet ”matematiska modeller”? Hur kan man använda programmering för att arbeta med matematiska modeller?

Om ni skulle använda programmering i er undervisning, vilket syfte skulle det ha?

Om du tänker praktiskt, hur tror du att du skulle jobba med programmering i matematik?

Hur ser du på din roll i undervisning i programmering?

Vilka sorts uppgifter tror du är lämpliga att jobba med i programmering?

Vilka möjligheter ser du med att arbeta med programmering i matematik?

Vilka tankar har du kring de olika programmeringsmiljöerna (t.ex. blockprogrammering, textbaserad programmering) som finns tillgängliga?



### Bilaga 3 – Resultat av verktygsanalys

Förklaring av koder i tabellen:

0 = ingen tydlig funktion tillgänglig

1 = funktion finns i verktyget

2 = funktion finns och utnyttjas i föreslaget material

Ämnesområde	Delområde	Robo-blockly	RoboBlockly undervisningsmaterial	Kojo	Kojo undervisningsmaterial svenska (Regnell 2016)	Kojo undervisningsmaterial engelska (Pant 2015)
Taluppfattning och tals användning	Reella tal, egenskaper och användning	2	Grade 3: multiplikation, division, tid, Grade 2: pengar, Grade 4: primtal	2	Skapa en variabel, träna multiplikation, gör ditt eget bankkonto	s. 33 beräkningar
	Talsystemets utveckling, metoder för beräkningar	2	Grade 5: decimaltal och bråk	2	Värden och uttryck	s. 33 beräkningar
	Tal i potensform	1		1	Kommando $\text{pow}(x,n)$ $\text{pow}(10,2)=100$	
	Beräkningar med digital teknik	1		2	Värden och uttryck	s. 33 beräkningar, area s. 36, variabler s. 57-60, ränta s. 82
	Rimlighetsbedömning	0		0		

Ämnesområde	Delområde	Robo-blockly	RoboBlockly undervisningsmaterial	Kojo	Kojo undervisningsmaterial svenska (Regnell 2016)	Kojo undervisningsmaterial engelska (Pant 2015)
Algebra	Variabelbegreppet	2	Algebra: genomgående	2	Rita en kvadratgubbe, rita en polygon, sätt namnen på värden, Skapa en variabel	s. 17 rektangel s. 57-60 variabler s. 84 teori variabler
	Algebraiska uttryck, formler och ekvationer	2	Algebra: genomgående	2	Gör en While loop	s. 45 geometri s. 71 geometri s. 84 teori
	Metoder för ekvationslösning	2	Algebra: 4, 5, 6	1		
	Mönster i talföljder och geometriska mönster	2	Grade 4: talserier	2	Geometriska mönster - Rita många polygoner	s. 45, s. 47 geometri s. 77, s. 103 talserie
	Algoritmer vid programmering	2	Algebra: genomgående	2	Genomgående	Genomgående, s. 94-96 logik
Geometri	Geometriska objekt, inbördes relationer och egenskaper	2	Grade 6: geometriska figurer	2	Rita en polygon	s. 17 (rektangel), vinklar (inner- och yttervinkel) s. 30 cirkelbågar s. 42, 45, s. 71 polygon
	Avbildning, konstruktion av geometriska objekt	2	Grade 6: geometriska figurer	2	Rita en kvadratgubbe, polygon	s. 20 ratio s. 30 vinklar s. 39 geometriska former

Ämnesområde	Delområde	Robo-blockly	RoboBlockly undervisningsmaterial	Kojo	Kojo undervisningsmaterial svenska (Regnell 2016)	Kojo undervisningsmaterial engelska (Pant 2015)
Geometri forts.	Skala vid förminskning och förstoring av två- och tredimensionella objekt	2	Grade 4: 14, Grade 7: förminskning	2	Rita en kvadratgubbe	s. 20 ratio s. 62-63 procent, geometri
	Likformighet och symmetri i planet	2	Grade 8	2	Kvadrat med parameter	
	Beräkning av area, omkrets, volym, enhetsbyten	1	Grade 6, grade 7: omkrets, area	1		s. 36 beräkning av area
	Geometriska satser och formler, argumentation	1		1		s. 44-45 grader i figurer
Sannolikhet och statistik	Likformig sannolikhet, beräkningar	1		1		
	Kombinatorik	1		1		
	Tabeller, diagram och grafer	1	Genomgående: grafer	2	Grafer (stapeldiagram finns i "Exempel - Programmering allmänt - Indata/grafik"	s. 85 stapeldiagram
	Lägesmått, spridningsmått	2	Algebra 10: medelvärde	2	Beräkna med kod	s. 90 medelvärde
	Bedömningar av risker och chanser	0		1	Slumptal	

Ämnesområde	Delområde	Robo-blockly	RoboBlockly undervisningsmaterial	Kojo	Kojo undervisningsmaterial svenska (Regnell 2016)	Kojo undervisningsmaterial engelska (Pant 2015)
Samband och förändring	Procent för att uttrycka förändring och förändringsfaktor	2	Grade 5: procent, Grade 7: förminskning	2		s. 20 ratio s. 62-63 procent, geometri s. 82 ränta
	Funktioner och räta linjens ekvation	2	Algebra: 3, 4, 5, 6	2		s. 51 koordinatsystem
Problemlösning	Strategier för problemlösning	1	Genomgående men inte explicit	1	Genomgående men inte explicit	s. 49 geometri
	Värdering av valda strategier och metoder	1	Genomgående men inte explicit	1	Genomgående men inte explicit	s. 28 trace funktion
	Matematisk formulering av frågeställningar	2	Algebra 12	2	Gör en kapplöpning	s. 36 area s.97 rita kvadrat/rektangel s. 106 rita
	Enkla matematiska modeller	2	Algebra 12	2	Gör en kapplöpning	s. 36 area s. 82 ränta
	Hur algoritmer kan skapas, testas och förbättras	2	Genomgående	2	Genomgående	Genomgående, t.ex. s. 39, s. 49, s. 100-101