



HÖGSKOLAN
DALARNA

Examensarbete 1 för ämneslärarexamen inriktning 7–9

Grundnivå 2

Programmering för problemlösning i matematik

Programming for problem solving in mathematics

Författare: Charis Snell
Handledare: Eva-Lena Erixon
Examinator: Eva Taflin
Ämne/huvudområde: Matematikdidaktik
Kurskod: MD2020
Poäng: 15 hp
Examinationsdatum: 29 maj 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 deltagande i undervisning i programmering på grundskolan påverkar elevernas förmåga att lösa matematiska problem och att ta reda på vilka elevaktiviteter som används när programmeringsundervisningen som är kopplad till matematisk problemlösning genomförs. Studien genomfördes som en systematisk litteraturstudie vilket innebär att tidigare forskning har sökts igenom för att kunna besvara frågeställningarna. Resultaten visar att ämnet är komplext och att det är svårt att dra några tydliga slutsatser utifrån den forskning som finns om ämnet. Många studier är positiva till programmering som ett sätt för elever att formulera och lösa problem. Den största kvantitativa undersökningen finner dock inget bevis för att elever får en ökad problemlösning förmåga genom arbete med programmering. Andra studier, som har undersökt både arbete med robotar och spelbyggande med skärmbaserade programmeringsspråk, beskriver hur elever får en mängd positiva effekter från arbete med programmering, som ökad förståelse av matematiska begrepp, ökad problemlösning förmåga och ökad uthållighet och motivation. Elevaktiviteter som används inkluderar strukturerade problem, fria utforskningar, styrning av robotar, spelbyggande, kollaborativt arbete och att lära ut matematik till andra elever.

Nyckelord:

Programming in school, problem solving, mathematics teaching.

Programmering i skolan, problemlösning, matematikundervisning.

Innehållsförteckning

1	Inledning	1
2	Syfte och frågeställningar	2
3	Bakgrund.....	3
3.1	Matematiska förmågor och kompetenser	3
3.2	Problemlösning i matematik.....	4
3.2.1	Problemlösningsförmågan	4
3.2.2	Kognitiva och metakognitiva perspektiv på problemlösningsförmåga... 6	
3.2.3	Affektiva faktorer som påverkar problemlösningsförmågan	7
3.3	Programmering i skolan	7
3.3.1	Programmering i skolan ur ett historiskt perspektiv	7
3.3.2	Programmeringsmiljöer, programmeringsspråk och digitala verktyg	8
3.3.3	Datavetenskapligt tankesätt.....	9
3.3.4	Överföring av förmågor mellan programmering och problemlösning....	9
4	Metod	10
4.1	Studiens design.....	10
4.2	Etiska övervägande.....	11
4.3	Beskrivning av sökprocessen	11
4.3.1	Databaser.....	11
4.3.2	Urvalskriterier	12
4.3.3	Sökstrategi.....	13
4.3.4	Urval av artiklarna.....	14
4.4	Sökresultat	17
4.4.1	Utvald litteratur	17
4.4.2	Kvalitet i den utvalda litteraturen.....	19
4.4.3	Analys av de utvalda artiklarna.....	19
5	Resultat.....	26
5.1	Hur påverkar undervisning i programmering elevernas förmåga att lösa matematiska problem?	26
5.1.1	Påverkan på elevernas matematiska problemlösningsförmåga generellt	26
5.1.2	Påverkan på matematiska faktorer	28
5.1.3	Påverkan på kognitiva faktorer	29
5.1.4	Påverkan på övriga faktorer	30

5.2 Med hjälp av vilka elevaktiviteter genomförs programmeringsundervisning som är kopplad till matematisk problemlösning?	32
5.3 Sammanfattning av resultat.....	33
6 Diskussion.....	34
6.1 Metoddiskussion	34
6.2 Resultatdiskussion.....	35
6.2.1 Påverkan på problemlösningförmåga generellt	36
6.2.2 Påverkan på matematiska faktorer	36
6.2.3 Påverkan på kognitiva faktorer	37
6.2.4 Påverkan på övriga faktorer	37
6.2.5 Elevaktiviteter	38
6.2.6 Studiestorlek.....	39
6.2.7 Språket	39
6.3 Avslutande reflektion	39
7 Förslag till framtida forskning	40
Litteratur.....	41

1 Inledning

Inom matematikämnet i årskurs 7-9 förväntas eleverna ”utveckla sin förmåga att kunna formulera och lösa problem med hjälp av matematik” (Skolverket 2016a, s. 56). Matematik beskrivs i läroplanen som ”en kreativ, reflekterande och problemlösande aktivitet som är nära kopplad till den samhälleliga, sociala och tekniska utvecklingen” (Skolverket 2016a, s. 55). Matematikämnet är viktigt för eleverna att behärska och bidrar till att eleverna kommer att kunna ”fatta välgrundade beslut i vardagslivets många valsituationer och ... delta i samhällets beslutsprocesser” (Skolverket 2016a, s. 55).

Problemlösning tillskrivs en central roll i läroplanen för matematik och den centrala rollen betonas även i målen för PISA-undersökningen i matematik, där problemlösningens förmåga är det som främst utvärderas:

Målet med matematik i PISA (*mathematical literacy*) är att utvärdera elevers förmåga att integrera och tillämpa matematiska kunskaper och färdigheter i en mängd olika realistiska situationer. Detta innebär en förskjutning i synen på matematik, från att se matematik som en samling begrepp och färdigheter att bemästra till att förstå matematik som en meningsfull problemlösande aktivitet. (Skolverket 2016b, s. 24)

Att eleverna förstår och kan använda matematiken som de lär sig i skolan för att lösa verkliga problem utanför klassrummet kan förstås som ett av målen med matematikundervisning. Att lära sig problemlösning och därmed utveckla en problemlösningens förmåga är komplex. Kilpatrick (1985) menar att forskare redan mycket tidigare visste att problemlösning var komplex, men förstod inte hur komplex det var (s. 11).

Redan 1945 publicerades den första utgåvan av Polya *How to solve it*, med förslag till strategier för lyckad problemlösning som matematiklärare än idag skulle kunna använda sig av i klassrummet. Grundstrategin består av fyra steg:

- Skapa en förståelse för problemet och det som är okänt
 - Hitta relationer mellan det kända och okända, gör upp en plan, rita en figur
 - Genomför planen, kontrollera att varje steg är korrekt och logisk
 - Reflektera över och kontrollera resultat.
- (Polya 1957, s. xvi-xvii)

Trots tiden som har gått sedan denna tidiga publikation förklarar Nesher, Hershkovitz och Novotna i 2003 att misslyckandet i undervisning i textbaserade problemlösning påvisar svårigheterna i att överföra expertkunskap till nybörjare (s. 153).

Under 2016 gav den svenska regeringen Skolverket uppdraget att undersöka hur programmering skulle kunna införas i grundskolan (Skolverket 2016c). Undervisningsrådet föreslog att programmering skulle inkluderas i kursplanen för matematik och att eleverna skulle tränas ”i att skapa, beskriva och följa entydiga och stegvisa instruktioner och att använda programmering för att lösa problem.” (Skolverket 2016c). År 2017 beslutade regeringen att alla elever i grundskolan skulle få undervisning i programmering inom ramen för matematikämnet

(Regeringskansliet 2017). Argumentationen för ändringarna i kursplanen är ett tydliggörande av ”skolans uppdrag att stärka elevernas digitala kompetens” (Regeringskansliet 2017, s. 1). Ändringarna i kursplanen för matematik i årskurs 7-9 är i de delar av det centrala innehållet som behandlar algebra och problemlösning. Inom problemlösning i årskurs 7-9 inkluderas formuleringen ”hur algoritmer kan skapas, testas och förbättras vid programmering för matematisk problemlösning” (Regeringskansliet 2017, s. 2). Skolverket presenterar dock ingen forskning som stödjer användningen av programmering inom matematikundervisning eller förklarar hur programmeringsundervisning med koppling till problemlösning bör genomföras.

Matematiklärare har vanligtvis varit de som har utvecklat programmering i skolan, ofta utan lämplig kompetens i ämnet (Rolandsson och Skogh 2014, s. 19). Eftersom programmering nu ska införas på grundskolenivå inom ämnet matematik känns det aktuellt för mig som blivande lärare att tillägna mig mer kunskap kring hur programmering bör användas inom matematikämnet.

Ett exempel från litteraturen som väckte mitt intresse för att undersöka en möjlig relation mellan programmering och problemlösning var Narodes (1987, s. 20-22) beskrivning av en observation av en student som arbetade med att lösa ett matematiskt problem. Studenten skulle skriva en ekvation för att beskriva relationen, ”det bor åtta gånger så många människor i Kina (C) som i England (E)”. Först skrev studenten (felaktigt) $8C = E$. När studenten skrev programmeringskod för att lösa problemet formulerade studenten om sin felaktiga ekvation till den korrekta $8E = C$. Studenten tvingades granska logiken i sin matematiska modell kring populationerna i Kina och England. Narode relaterar denna observation till teorier om metakognition i problemlösning och menar att genomförandet av programmeringsprocessen tvingar studenter att använda heuristik, lösa element som står i konflikt med varandra och kontrollera om svaret är logiskt möjligt.

Om programmering kan användas inom matematikämnet både som ett ändamål, som en del av det centrala innehållet, och som ett pedagogiskt verktyg för att utveckla elevernas problemlösningsförmåga är det viktigt att lyfta fram forskning som stödjer detta perspektiv.

2 Syfte och frågeställningar

Syftet med litteraturstudien är att undersöka forskningsläget gällande hur undervisning i programmering på grundskolan påverkar elevernas förmåga att lösa matematiska problem. Som en del i detta syfte undersöks även vilka elevaktiviteter som används inom programmeringsundervisning som är kopplad till matematisk problemlösning.

Utifrån detta syfte formuleras följande frågor:

- Hur påverkar undervisning i programmering elevernas förmåga att lösa matematiska problem?

- Med hjälp av vilka elevaktiviteter genomförs programmeringsundervisning som är kopplad till matematisk problemlösning?

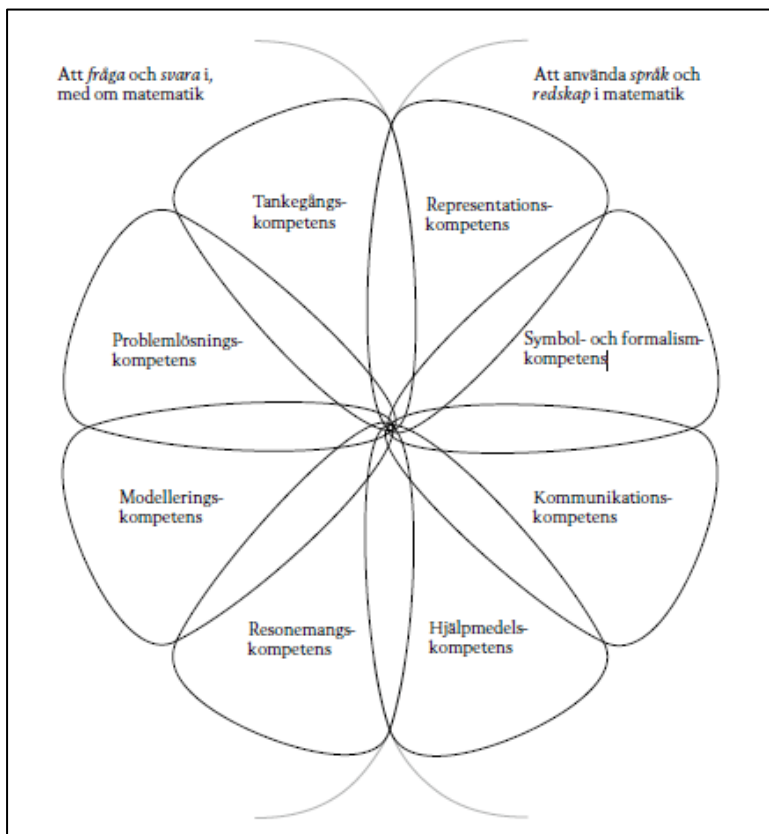
3 Bakgrund

Flera begrepp behöver definieras och relateras till varandra för att kunna förstå hur programmering kan påverka elevernas problemlösningsförmåga. Det första avsnittet behandlar matematiska förmågor generellt. Sedan följer en beskrivning av problemlösning i matematik samt en definition av problemlösningsförmågan. Därefter följer en beskrivning av vad som menas med programmering och datavetenskapligt tankesätt.

3.1 Matematiska förmågor och kompetenser

Elevernas matematiska förmågor kan delas upp i mindre komponenter som beskrivs som förmågor eller kompetenser. Enligt Skolverkets (2016a) beskrivning av matematikämnets syfte i kursplanen ska eleverna ”ges förutsättningar att utveckla” (s. 56) problemlösningsförmågan, begreppsförmågan, val och användning av matematiska metoder, resonemangsförmågan och förmågan att använda matematikens uttrycksformer. Med denna formulering menar Skolverket att eleverna ska utveckla dessa fem övergripande förmågor i matematikämnet.

Niss och Højgaard Jensen (2002) presenterar forskning om matematiska kompetenser som gjordes i det danska projektet *Kompetencer og matematiklæring (KOM)*. Under projektet utvecklades en modell baserad på åtta kompetenser (se Figur 1), där kompetenserna inte ska ses som skilda kompetenser utan som sammanvävda. Utformningen av PISA-undersökningen i matematik 2015 baserades på Niss och Højgaard Jensens modell (OECD 2016, s. 68). Där används begreppet *capabilities* istället för *competencies* (OECD 2016, s. 64).



Figur 1: Matematiska kompetenser enligt Niss och Jensen (2002, s. 45), översättning Helenius (2006, s. 13)

Kilpatrick, Swafford och Findell (2001, s.116) menar istället att matematisk kompetens består av fem element. Dessa fem element måste behärskas för att kunna matematik, där den komplexa relationen och det ömsesidiga beroendet mellan elementen betonas. De fem elementen de beskriver är:

- förståelse för matematiska begrepp
 - att kunna genomföra matematiska procedurer eller operationer flytande
 - att kunna använda strategier för att formulera och lösa matematiska problem
 - att ha en förmåga att reflektera och resonera matematiskt
 - att ha en positiv syn på nyttan med matematiken.
- (Kilpatrick et al. 2001, s.116)

Det finns således många olika termer och modeller för att beskriva det som ska utvecklas i matematikämnet och ingå i att kunna matematik. Alla beskrivningar innehåller dock minst en komponent som fokuserar på problemlösningsförmågan. I kursplanen (Skolverket 2016) nämns inte kompetenser utan istället används begreppen kunskaper och förmågor. I Skolverkets publikation om PISA-undersökning 2015 (Skolverket 2016b) används formuleringen ”elevers förmåga att integrera och tillämpa matematiska kunskaper och färdigheter” (s. 24) och begreppet kompetenser nämns inte. Formuleringen i OECD-publikationen om PISA-undersökningen 2015 är elevens “capacity to formulate, employ and interpret mathematics” (OECD 2016, s. 64). Eftersom Skolverket använder begreppet *förmåga* istället för något alternativ som *kompetens* eller *kapacitet* används *förmåga* i fortsättningen i denna text.

3.2 Problemlösning i matematik

Problemlösning har en särskild ställning i kursplanen för matematik. Den är både en av de övergripande förmågorna som eleverna ska utveckla och en del i det centrala innehållet (Skolverket 2016a, s.55-60). Enligt Skolverket är matematiska problem ”situationer eller uppgifter där eleverna inte på förhand känner till hur problemet ska lösas” (Skolverket 2011a, s. 25). Det som eleverna förväntas lära sig inom problemlösningssområdet är strategier för att lösa problem, att kunna tolka innehållet i problemet samt att kunna ”formulera frågeställningar med matematiska uttrycksformer” (Skolverket 2011a, s. 25).

Det som upplevs som ett problem för en person är inte nödvändigtvis ett problem för en annan person eftersom personerna kan ha olika förmågor gällande matematisk problemlösning (Skolverket 2011a, s. 9). Även vid en senare tidpunkt kan en persons problemlösningsförmåga ha utvecklats så att en matematisk uppgift som tidigare upplevdes som ett problem sedan kan behandlas som en rutinuppgift (Kilpatrick 1985, s. 3). Detta perspektiv kan jämföras med Skolverkets definition på ett problem ovan. Om eleven vet hur problemet ska lösas så definieras det inte som ett problem.

3.2.1 Problemlösningsförmågan

I alla de olika beskrivningarna av matematiska förmågor eller kompetenser som skildras i 3.1 *Matematiska förmågor och kompetenser* ovan finns problemlösning som en del. Enligt Skolverket (2011a) innebär problemlösningsförmåga att eleverna ska förstå att det kan finnas olika sätt att lösa ett problem. Eleverna ska

även ”kunna reflektera över och värdera rimligheten i resultatet i relation till problemet” (s. 8). Skolverket menar att det inte finns rutinproblem eftersom problem är ”uppgifter som inte är av rutinkaraktär” (s. 25) och eleverna måste således ”prova sig fram” (s. 25). Kilpatrick et al. (2001, s. 124) skiljer dock mellan rutin och icke-rutin problem (s. 126). De förklarar att det kan finnas flera sätt att lösa ett problem och att flexibilitet är viktigt för att lyckas med lösning av icke-rutin problem (2001, s. 127). Skolverket (2011a, s. 25) menar att problem kan kräva en matematisk tolkning av en situation, som till exempel i textbaserade problem där eleverna måste läsa en text och formulera en beskrivning av situationen med matematiska uttryck. Problem kan också vara rent matematiska, till exempel när det gäller att hitta mönster i en talserie.

För att eleverna ska förstå ett problem och angripa sig på matematiken i problemet måste de kunna skapa någon sorts modell över de olika delarna som ingår i problemet. Kilpatrick et al. (2001, s. 124) förklarar att det är viktigt att eleverna kan bygga en tankemodell av ett problem som representerar relationerna mellan de olika elementen i ett problem. Niss och Højgaard Jensen (2002) menar att problembehandlingskompetens innebär att kunna formulera matematiska frågor och lösa dem (s. 49). Även det som Niss och Højgaard Jensen beskriver som tankegångskompetens kan ses som en del av det som benämns problemlösningsförmåga av Skolverket. Tankegångskompetens innefattar generalisering, att ställa matematiska frågor och att veta vilka sorts svar man ska förvänta sig (s. 47). Modelleringskompetens innebär att kunna tolka en situation och beskriva den med ett matematiskt språk, samt att kunna översätta resultaten tillbaka till verkligheten (s. 52). Från Niss och Højgaard Jensens modell av matematiska kompetenser är det främst problembehandlingskompetens, modelleringskompetens och tankegångskompetens som ingår i det som beskrivs av Skolverket som problemlösningsförmåga.

Garofalo och Lester (1987, s. 166) påpekar att det tar lång tid att utveckla både strategier för problemlösning och de kontrollmekanismer som krävs för att effektivt kunna använda strategierna, och därför tar det lång tid för elever att utveckla sin problemlösningsförmåga. Lesh och Zawojewski (2007, s. 784-785) har genomfört en översikt av forskning kring lärande i problemlösning och modellering och utvecklingen av elevers problemlösningsförmåga. De förespråkar en problemlösningspedagogik som baseras på en holistisk modelleringsprocess och inkluderar alla faser av modelleringsprocessen. Utveckling av problemlösningsförmågan kräver att eleven utvecklar en förståelse av en komplex process och engagerar sig i utveckling av prognoser, verifikationer och utveckling av modeller.

Det finns således många faktorer som ingår i problemlösningsförmågan. Möllehed (2001, s. 158) undersökte vilka faktorer som påverkar elevernas problemlösningsförmåga i årskurserna 4-9. Han visar att ungefär 60 % av elevernas fel vid lösning av matematiska problem kan förklaras med hänvisning till kognitiva faktorer. 15% av felen kan förklaras med bristande uppmärksamhet och 25 % med hänvisning till matematiska faktorer. Möllehed (2001, s. 104) har utvecklat en modell över påverkansfaktorerna (*Appendix A: Mölleheds modell av påverkansfaktorer för matematisk problemlösning*). I modellen samlas påverkansfaktorerna under tre överrubriker: matematiska faktorer, kognitiva faktorer och övriga faktorer. De

matematiska faktorerna är räkneförmåga, noggrannhet, förståelse av matematiska begrepp, förståelse av samband mellan storheter, förståelse av samband mellan enheter och talförståelse (s. 102). De *kognitiva faktorerna* är separation (att undvika att blanda samman olika ting), förståelse av proportionalitet och konstans eller beständighet, kombinationsförmåga, förståelse av relationer mellan helheten och delar, logiskt tänkande, textförståelse, visualiseringsförmåga och verklighetsuppfattning (s. 99). Det som benämns *övriga faktorer* är uppmärksamhet, tidigare erfarenheter och kunskaper och affektiva faktorer som motivation, attityder, känslor och värderingar (s. 103). Av dessa övriga faktorer är det endast uppmärksamhet som har studerats ingående som en påverkansfaktor av Möllehed. Påverkan av de andra aspekterna (tidigare erfarenheter och kunskaper och affektiva faktorer) förklaras utifrån bakgrundsforskning och samlas i något som kallas för ett *utgångsläge* för problemlösning.

3.2.2 Kognitiva och metakognitiva perspektiv på problemlösning

Metakognitiva funktioner är de funktioner som gör att eleverna bli medvetna om och kan styra sitt eget tänkande (Garofalo och Lester 1985, s. 163). En modell (Tabell 1) för problemlösning som identifierar kognitiva och metakognitiva funktioner som används i lösning av matematiska uppgifter presenteras av Garofalo och Lester (1985, s. 171). Tabellen beskriver fyra kategorier av aktiviteter som är relevanta för matematisk problemlösning. I början av problemlösningmomenten är *orientering* viktigt, för att analysera och förstå informationen i problemet. Sedan underlättar funktionen *organisation* skapandet av en plan för att kunna lösa problemet. Under *utförandet* av planen måste procedurer och beräkningar genomföras. Detta kan till exempel innebära att eleverna konstruerar en matematisk modell och genomför aritmetiska beräkningar. Avslutningsvis behöver eleverna kontrollera att delresultaten är rimliga och precisa och förändra planen om det krävs. Efter problemlösning måste resultatet *verifieras*.

Tabell 1: Kognitiva och metakognitiva funktioner vid matematisk problemlösning (Garofalo och Lesters modell 1985, s. 171)

Kognitiv/metakognitiv funktion	Handlingar
Orientering - Strategisk bedömning och förståelse av ett problem.	Analys av informationen, om uppgiften är känd, svårighetsnivå.
Organisation – planering och val av strategi.	Identifikation av mål och delmål. Övergripande och delmålsplanering.
Utförande – följ planerna.	Utför aktioner, kontrollera utveckling, val t.ex. snabbhet eller noggrannhet.
Verifiera – reflektera över planerna, val och resultat.	Reflektion över förståelse, vald matematisk modell och planerna. Reflektion över utförandet och rimlighet av resultaten.

Möllehed (2001, s. 158) menar att metakognitiva faktorer har en stor påverkan på elevernas förmåga att lösa matematiska problem. Även Lesh och Zawojewski (2007, s. 773), i deras översyn av forskningsläget kring lärande i problemlösning och modellering, förklarar att det är viktigt för elever att utveckla metakognitiva funktioner.

3.2.3 Affektiva faktorer som påverkar problemlösningsförmågan

Möllehed (2001, s. 103) förklarar att det finns andra faktorer än de matematiska och metakognitiva som påverkar elevernas förmåga att lösa matematiska problem. Några av dessa är elevernas tidigare erfarenheter, motivation, attityd till matematik och problemlösning och deras känslor. Även McLeod (1985) beskriver hur vissa faktorer, som kallas för affektiva faktorer, påverkar elevens problemlösningsförmåga. Några affektiva faktorer som McLeod nämner är oro eller stress i problemlösningssituationer, fysiologiska faktorer som spänning och avslappning och elevernas självständighet vid problemlösning (s. 274-5). En annan faktor som påverkar problemlösningsförmågan är vad det är som eleverna tror ha orsakat deras framgång eller misslyckande vid problemlösning. Exempel på detta är om eleverna tror att det var deras tur, förmåga, ansträngning eller svårighetsnivån på problemet som har gjort att de har lyckats eller misslyckats (McLeod 1985, s. 275).

Lampert (i Schoenfeld 1992, s. 359) har sammanställt en lista över vanliga elevuppfattningar kring matematiska problem i USA. Dessa inkluderar att det bara finns ett korrekt svar och ett sätt att lösa problemet på och en tro på att om eleven har förstått matematiken ska de kunna lösa problemet inom fem minuter. Schoenfeld (1992, s. 359) förklarar att tron på att alla problem ska kunna lösas inom fem minuter betyder att elever är benägna att ge upp om de inte lyckas lösa ett problem inom några få minuter, även om de skulle ha kunnat lösa det om de fortsatte. Elevernas uppfattning om naturen av lösningar på matematiska problem påverkar därför också deras problemlösningsförmåga.

3.3 Programmering i skolan

I dokumentation kring ändringarna som kommer att införas i läroplanen gällande programmering definieras inte vad som menas med programmering eller syftet med införandet mer än att elevernas digitala kompetens ska stärkas (Regeringskansliet 2017, s. 1). I detta avsnitt presenteras en kort överblick av programmering i den svenska skolan historiskt sett samt en presentation av de tillgängliga programmeringsmiljöerna och de digitala verktygen som skulle kunna beskrivas som programmeringsverktyg. Det datavetenskapliga tankesättet beskrivs och potentiell överföring av förmågor mellan programmering och problemlösning diskuteras.

3.3.1 Programmering i skolan ur ett historiskt perspektiv

Ämnet programmering finns redan idag på gymnasienivå, där syftet är att eleverna ska få kunskaper om grunderna för programmering och programmeringsprocessen. De ska utveckla förmågan ”att analysera, designa, implementera, testa, resultatbedöma och vidareutveckla program” (Skolverket 2011b, s. 1).

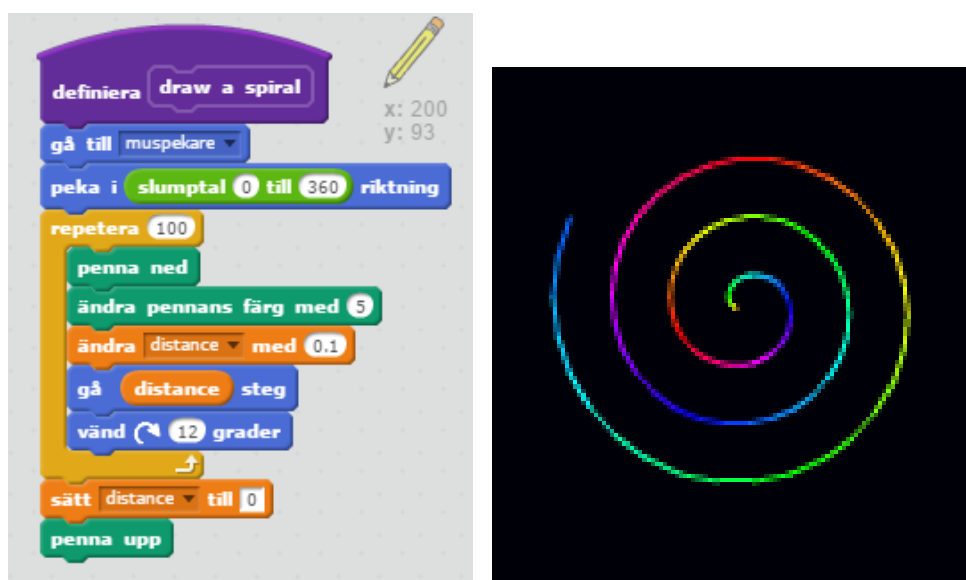
Rolandsson och Skogh (2014, s. 13) rapporterar om olika projekt med programmering i gymnasieskolor och beskriver forskning om programmering under 1970- och 1980-talet. Rolandsson och Skogh beskriver även hur det fanns en positivism som omgav programmering på 70-talet. Det fanns förhoppningar om att programmering skulle kunna användas som ett verktyg i matematikämnet och att eleverna skulle utveckla sin problemlösningsförmåga genom att lära sig att

konstruera algoritmer. Det upplevdes dock som att programmering var för tidskrävande och svår att lära ut till eleverna (s. 11). Detta ledde till en kursplan med minskad fokus på kodskrivningsprocessen och mer fokus på att använda mjukvaror för att utveckla datorprogram utan att behöva skriva kod från grunden (s. 16).

3.3.2 Programmeringsmiljöer, programmeringsspråk och digitala verktyg

Rolandsson (2015, s. 4) beskriver hur vissa forskare menar att den moderna utvecklingen av teknik och nyare programmeringsmiljöer betyder att eleverna kan lägga mer fokus på de logiska tankeprocesserna som krävs för programmering än på färdigheter i att kunna skriva med ett visst programmeringsspråk. Med tidigare programmeringsmiljöer behövde eleverna skriva ut all kod med text. Detta innebar att elevernas kod ofta innehöll syntaktiska fel som innebar att programmet inte fungerade som det skulle. Felsökning och brist på tydlig återkoppling från programmet gjorde att programmering tog mycket tid och var frustrerande. Tekniken med pusselbitar av kod betyder att många av de tidigare problemen undviks.

Under 1980-talet utvecklades de första programmeringsmiljöerna som anpassades för programmering i skolan. Den första av dessa var LOGO som utvecklades av Papert (Healy och Kynigos 2009, s. 64). I regeringens beslut (2017) om nytt innehåll i matematik står det att i det centrala innehållet för årskurs 7-9 ska eleverna lära sig ”programmering i visuella/olika programmeringsmiljöer” (s. 2). *Scratch* är ett exempel på en så kallad visuell programmeringsmiljö. I *Scratch* används färdiga pusselbitar av programkod som placeras i en ordning för att skapa ett program (Scratch 2013). *Figur 2: Exempel från Scratch-programmeringsmiljö* visar ett program som ska rita en spiralform utifrån punkten där muspekaren befinner sig på skärmen (programmet finns på <https://scratch.mit.edu/projects/11641125/#editor> för den som vill se mer eller utforska själv). Programmet använder matematiska begrepp som grad och avstånd. Det använder även repetition av programmeringsinstruktioner, något som är grunden till de iterativa procedurerna som finns i algoritmer.



Figur 2: Exempel från *Scratch*-programmeringsmiljö (Scratch 2013)

Det finns många olika programmeringsmiljöer och programmeringsspråk som används i undervisning i skolor idag i olika länder. Hubweiser et al. (2015, s.76) har sammanställt en översikt av vilka programmeringsspråk som används i skolor (i alla ämnen). Dessa inkluderar visuella miljöer som *Scratch* och *Kodu* samt språk som i huvudsak används för programmering av robotar som *LEGO Mindstorms*. Hubweisers et al. (2015, s. 76) sammanställningen av programmeringsspråk återges i *Appendix B: Programmeringsspråk i skolor*.

Det digitala verktyget kalkylblad kan också beskrivas som en programmeringsmiljö, eftersom cellerna i kalkylbladet kan programmeras för att interagera med varandra. Programmeringsspråk som *Visual Basic* kan användas för att skapa *macros* (korta datorprogram som fungerar inuti kalkylbladsprogrammet) som utför olika operationer. Aydin (2005, s. 28) skiljer mellan programmering och matematiska verktyg som kalkylblad. Han menar att verktyg som kalkylblad kan användas som ett alternativ till programmering för att utveckla koncept som algoritmer. Även en grafräknare skulle kunna klassificeras som en programmeringsmiljö. Grafräknare och kalkylblad (interaktiva tabeller där cellerna kan länkas tillsammans genom olika formler) kan med fördel användas i matematikundervisningen. Sollervall och Hellenius (2016, s. 2) i en publikation för Skolverket beskriver hur arbete med kalkylblad kan hjälpa elever att se matematiska samband och förstå algebra.

3.3.3 Datavetenskapligt tankesätt

Wing (2006, s. 35) skiljer mellan programmering och datavetenskap och menar att datavetenskap är mycket mer än färdigheter i programmering. Hon menar att alla barn har rätt att lära sig ett datavetenskapligt tankesätt och att det är ett grundläggande analytiskt verktyg. Wing beskriver hur människors kreativa förmågor tillåter oss att använda datorer som verktyg för att lösa komplexa problem. Hon förklarar att datavetenskap innefattar många element, som planering, effektivitet, reflektion, reduktion och abstraktion (2006, s. 33). Många av dessa element används även vid matematisk problemlösning.

3.3.4 Överföring av förmågor mellan programmering och problemlösning

Mayer och Wittrock (1996, s. 48) definierar *transfer* eller *överföring* som lärande eller kunskap som påverkar nytt lärande eller problemlösning i en ny situation. Det kan vara något som en elev har lärt sig tidigare, till exempel en strategi för att lösa en sorts problem, som de sedan kan använda för att lösa en annan sorts problem. De former av överföring som har mest relevans med tanke på programmering och problemlösning är:

- överföring av en generell förmåga eller strategi från ett specifikt område till ett annat
- utvecklande av metakognitiva processer (1996, s. 50).

Mayer och Wittrock (1996, s. 55) beskriver även skapandet av analogi mellan två olika problem som ett sätt att överföra strategier och kunna lösa nya problem. De menar dock att det finns mycket forskning som pekar på de svårigheter elever har i att känna igen liknande strukturer i problem. Det kan tänkas vara logiskt att använda strategier som fungerar på ett problem på ett annat problem som har en

liknande struktur men det kan vara just momentet att *se* likheter mellan strukturer som vållar svårigheter. Lesh och Zawojewski (2007, s. 773) varnar för förväntningar på att kunna förbättra problemlösningssförmågan hos icke-expert elever genom att lära ut beteende och strategier som experter använder för att lösa problem. Det kan således inte tas för givet, bara för att programmering och problemlösning kräver liknande kognitiva och metakognitiva processer och modelleringsprocesser, att eleverna kommer att lyckas med att överföra några nyvunna strategier från det ena området till det andra.

4 Metod

I detta avsnitt presenteras den valda metoden för studien. En överblick av studiens design presenteras och de etiska överväganden som har gjorts beskrivs. Sökprocessen redogörs för, med en presentation av databaser, urvalskriterier och sökstrategin. Sökresultaten framförs med en presentation av den utvalda litteraturen och en värdering av kvaliteten i den utvalda litteraturen. Analysprocessen beskrivs och en sammanfattning av artiklarnas resultat presenteras.

4.1 Studiens design

Studien genomförs som en systematisk litteraturstudie. Litteraturstudie är en metod som kan användas för att undersöka och sammanställa kunskapsläget i ett ämne, på ett systematiskt sätt (Eriksson Barajas, Forsberg & Wengström 2013, s. 17). Syftet och frågeställningarna, som handlar om att undersöka forskningsläget kring ett bestämt ämne, leder till att en systematisk litteraturstudie anses vara den lämpligaste metoden att användas för studien. Andra metoder hade inte gett samma möjlighet att på ett systematiskt sätt granska en stor mängd forskningsresultat.

Enligt Skollagen ska undervisning ”vila på vetenskaplig grund” (SFS 2010:800). Läraren har ett ansvar att utveckla undervisning i enlighet med resultat från vetenskaplig forskning för att efterleva Skollagen. Eftersom programmering nu införs som en obligatorisk del av det centrala innehållet i matematik är det nödvändigt för lärare att ha kunskaper om forskningsläget kring programmering som en del av matematikundervisning. Eriksson Barajas et al. (s. 19) menar att resultaten av en systematisk litteraturstudie kan användas som underlag för att utveckla undervisningen.

En systematisk litteraturstudie innebär att vetenskaplig litteratur söks igenom systematiskt, granskas och sammanställs för att hitta litteratur som uppfyller syfte och besvarar frågeställningarna (Eriksson Barajas et al. 2013, s. 31). Arbetet utförs i flera steg, där det första steget innebär formulering av ett problem och syfte samt frågeställningar. Sedan görs en plan för studien, som beskriver vilka sökord och strategier som ska användas och vilka databaser som ska användas i sökprocessen. Kriterierna för urval av artiklarna och sökmetoderna beskrivs explicit. Artiklar som påträffas i sökningar bedöms utifrån relevans för studiens syfte och kvalitetskriterier. Artiklarna som inkluderas i litteraturstudien analyseras genom en innehållsanalys (Eriksson Barajas et al. 2013, s. 164). Innehållsanalysen är baserad på kriterier som är relevanta för att hitta svar till frågeställningarna och

beskrivs mer i detalj i 4.3.4 *Urval av artiklarna*. Resultatet av analysen sammanställs och diskuteras (Eriksson Barajas et al. 2013, s. 168).

4.2 Etiska övervägande

Etiska aspekter som är viktiga i en litteraturstudie är att metoden beskrivs och följs på ett systematiskt sätt och att en representativ analys och presentation av resultatet genomförs (Eriksson Barajas et al. 2013, s. 69). Det är viktigt att inte vilseleda läsaren och att presentera alla resultat, även de som inte stödjer forskarens åsikt. De studier som ingår i den inkluderade litteraturen bör ha reflekterat över etiska frågor och om nödvändigt fått tillstånd från en etisk kommitté (s. 70). I denna studie har urvalet av artiklarna genomförts utifrån i förhand bestämde kriterier som redovisas i 4.3.2 *Urvalskriterier*. Analys av artiklarnas resultat genomförs systematiskt och analyskategorierna redovisas. Alla relevanta resultat presenteras.

Litteraturen som är inkluderad ska granskas för att försäkra att den håller hög vetenskaplig kvalitet. Som forskare som ska sprida vetenskapliga resultat vidare är det ändå viktigt att granska kvaliteten av de inkluderade artiklarna genom att värdera bland annat urvalsstrategi, metodbeskrivning och analysen av resultatet (Eriksson Barajas et al. 2013, s. 155). Alla artiklar som inkluderas i denna studie har granskats av oberoende personer (*peer-reviewed*) vilket är ett krav för publicering i vetenskapliga tidskrifter och ska vara ett märke av hög kvalitet.

4.3 Beskrivning av sökprocessen

I detta avsnitt presenteras de databaser och sökord som har använts för sökning av litteratur. En beskrivning av sökmetoderna ges och de urvalskriterier som har använts dokumenteras. Resultaten av sökningarna och de utvalde artiklarna presenteras.

4.3.1 Databaser

Flera databaser har sökts igenom för att täcka så stort utbud av vetenskapliga texter som möjligt. De databaser som ansågs mest lämpliga för det valde ämnet (pedagogiskt arbete) valdes. Tillgång till databaserna har skett genom Högskolan Dalarnas bibliotek, vilket ger tillgång till artiklar i fulltext från många av de tidskrifter som finns på internet. De databaser som sökningarna har skett i är *Summon* (egentligen en sökmotor som söker igenom bibliotekets kataloger), *ERIC* (*Education Resources Information Centre*), *Teacher's Reference Center*, *Libris* och *DIVA*. I vissa fall har *Google Scholar* använts för att få tillgång till fulltext av artiklar i de fall där den inte har funnits tillgänglig i databasen.

- *Summon* är en sökmotor som ger en överblick av Högskolan Dalarnas biblioteks artiklar och böcker inom ett område. Sökningar kan begränsas genom att välja *peer-reviewed* (granskat av oberoende personer) och genom att välja datumbegränsningar.
- *ERIC* är den största databasen i världen av litteratur inom utbildningsvetenskap. Texterna är huvudsakligen på engelska. Sökningarna har gjorts genom tjänsten *Proquest*, något som ger tillgång till mer

avancerade sökbegränsningar, till exempel vilken utbildningsnivå artiklarna behandlar.

- Teacher's Reference Center (EBSCO) innehåller titlar och abstrakt från tidskrifter om utbildningsvetenskapliga ämnen. Här kan sökträffen begränsas till peer-reviewed och datumbegränsningar för publicering användas.
- Libris innehåller litteratur som finns på svenska bibliotek, inklusive avhandlingar. Här finns det inte möjlighet att begränsa sökträffen till peer-reviewed litteratur.
- DIVA (Digitala Vetenskapliga Arkivet) är ett öppet arkiv som innehåller svenska forskningspublikationer.

(Högskolan Dalarna 2017)

4.3.2 Urvalskriterier

Flera kriterier har använts vid sökningarna för att hitta de mest relevanta artiklarna för studiens syfte. Syftet med litteraturstudien är att undersöka hur deltagande i undervisningen i programmering på grundskolan påverkar elevernas förmåga att lösa matematiska problem och även att ta reda på vilka aktiviteter som används i genomförandet av denna undervisning. Därför begränsas sökningar för att hitta studier och artiklar som behandlar matematikundervisning och matematisk problemlösningsförmåga på grundskolan. Jag inriktar mig mot de högre årskurserna i grundskolan men det kan vara så att resultat av studier som har undersökt yngre barn ändå är relevanta för besvarandet av studiens syfte. Syftet med matematikämnet och förmågorna som eleverna ska utveckla i matematikämnet i Sverige är detsamma från årskurs 1-9, även om det centrala innehållet utvecklas genom årskurserna (Skolverket 2016a, s. 55-56). Eftersom studiens fokus är elever i grundskolan inkluderas inte resultat som behandlar matematikundervisning på gymnasienivå, högskolan eller vuxenutbildning. Därför begränsas sökningarna i ERIC genom att inte välja utbildningsnivåerna *kindergarten, pre-school education, early childhood education, adult education, higher education, postsecondary education, two year colleges*. Alla andra utbildningsnivåer väljs vid varje sökning. Dessa avgränsningsmöjligheter finns inte i de andra databaserna. Här avgränsas sökningarna manuellt vid läsning av titlar, sammanfattningar eller artiklar.

Artiklar som behandlar extra undervisning utanför ramen av ordinarie matematikundervisning skulle ändå kunna innehålla resultat av relevans för besvarandet av litteraturstudiens frågeställningar. Därför väljs inte artiklar bort på grund av att de handlar om undervisning utanför ramen för ordinarie undervisning.

Ett kriterium som används är att litteraturen ska vara publicerad efter år 2000, eftersom forskningen ska vara relevant för dagens utbildningsfrågor. När datorer började göra intrång i skolans värld under 1980-talet var programmering i skolor ett intressant forskningsämne och många artiklar skrevs. Enligt sökningar i ERIC, skedde det under 1990-talet en minskning av antalet publicerade artiklar kring

programmering i skolan, för att sedan börja öka igen efter år 2000. Eftersom datorer och tillgängliga programmeringsmiljöer utvecklas relativt fort anser jag att det är de nyare artiklarna som har mest relevans för frågor rörande programmering och problemlösning i skolan idag. Även läroplaner utvecklas och den forskning som skrevs under 1980- och 90-talen har möjligen ett annat och mindre relevant perspektiv på skolan och lärande än de artiklar som publicerats senare.

Artiklarna ska vara oberoende granskade, vilket ökar trovärdigheten av resultaten. Därför har sökningar begränsats till *peer-reviewed* litteratur i alla databaser förutom Libris, som saknar denna funktion. I Libris påträffades få artiklar vid sökningar vilket underlättade den nödvändiga manuella granskningen av detta kriterium.

Det nämndes redan 2011 i *Kommentarmaterial till kursplanen i matematik* (Skolverket 2011a, s. 10) att eleverna ska använda digital teknik som grafräknare och datorer för visualisering och skapandet av matematiska modeller. Fokuset i denna studie är på den del av programmering som beskrivs i Regeringskansliets besked (2017) som ”visuella och andra miljöer” (s. 2) och inte på programmering i andra digitala verktyg som kalkylblad. Därför väljs studier som handlar om programmering i kalkylblad eller grafräknare bort.

Studiens inriktning är på elevers problemlösningsförmåga i matematikämnet och programmeringsfärdigheter. Därför väljs studier som fokuserar på lärare eller lärarstudenter bort. Även vissa studier som har ett mer generellt fokus på problemlösning utan koppling till matematikämnet väljs bort.

Vid läsning av sammanfattningar och artiklarna upptäcktes ytterligare en betydelse av *programming* i US-kontext. Där kan ordet ha en betydelse som ligger nära *kursplan* eller *schemaplanering*. En annan betydelse som påträffades var som i uttrycket *linear programming* i matematik. Artiklar som fokuserar på dessa fenomen är inte relevanta för studien och väljs därför bort.

4.3.3 Sökstrategi

Syftet med studien är att undersöka hur undervisning i programmering på grundskolan påverkar elevernas förmåga att lösa matematiska problem och vilka aktiviteter som används inom programmeringsundervisning som är kopplad till problemlösning. Syftet ledde fram till att sökorden *problemlösning*, *programmering*, *matematikundervisning*, *matematik* och *skolor* valdes, samt deras engelska motsvarigheter *problem solving*, *word problems*, *story problems*, *programming*, *mathematics education*, *mathematics* och *schools*. Trunkeringen *program*ing* användes vilket ger träff för både den vanligaste stavningen *programming* och den mer ovanliga *programing*. I samråd med bibliotekarie hittades även sökfrasen ”*programmering i skolan*”/ ”*programming in schools*” som ger en snävare sökträff. Citattecken användes med fraserna ”*problem solving*” och ”*mathematics education*” vilket kräver att orden måste hittas i anslutning till varandra i dokumentet.

De olika beskrivningarna och organisationerna av utbildningsnivå i olika länder medför svårigheter. *High school* i USA kan motsvara delar av grundskolan och

gymnasiet i Sverige. Därför har artiklar som behandlar *high school* i första hand inkluderats och irrelevanta studier sållats bort manuellt. Under försök att raffinera sökträffen exkluderades *higher education*, men eftersom det medförde att vissa studier som involverade studenter på både högskolenivå och grundskolan inte inkluderades i sökträffarna ansågs det bättre att välja alla nivåer som skulle inkluderas istället.

Vid generella sökningar efter bakgrundsinformation hittades flera artiklar som bedömdes vara av relevans för studien men som inte upptäcktes vid sökning efter sökorden ovan. Därför bedömdes det lämpligt att göra ytterligare sökningar baserad på sökorden "problem solving" och namnen på de programmeringsspråk som används i skolor och som beskrivs i bakgrunden. Syftet med denna sökning var att hitta artiklar som behandlar de programmeringsspråk som beskrivs i bakgrunden och som är aktuella i skolmiljöer (Hubweiser et al. 2015, s.76). Detta medförde att alla databaser söktes igenom med de nya sökordskombinationerna: ("problem solving") AND ((scratch) OR (kodu) OR (logo) OR (agentcube) OR (agentsheet) OR (Alice) OR (blockly) OR ("game maker") OR (microworlds) OR ("Robot karol") OR (etoys) OR (BlueJ) OR (greenfoot) OR ("Java's cool") OR (jeroo) OR (lego) OR ("raspberry pi")) AND ((student) OR (pupil)). Denna sökning hittade artiklar som innehåller *problem solving* och minst ett av programmeringsspråken samt ordet *student* or *pupil*. Sökningen gjordes även med de svenska orden *problemlösning* och *elev*.

4.3.4 Urval av artiklarna

Tabell 2-7 nedan visar en översikt av urvalsprocessen. Tabellerna redovisar antalet träffar i de olika databaserna baserat på olika kombinationer av sökord. Sökningar med svenska sökord i ERIC och Teacher's Reference Center gav inga träffar och därför utelämnas dessa sökningar från tabellen. När antalet sökträffar översteg 200 lästes bara de första 200 titlarna, som sorterades utifrån relevans. En bedömning gjordes att träffar längre ner på sorteringslistan inte var av relevans för litteraturstudien. Annars lästes alla titlar som påträffades i sökningarna. Efter att titlarna lästes gjordes ett preliminärt urval. Sedan lästes sammanfattningarna av de artiklarna som bedömdes potentiellt vara av intresse. Slutligen översiktislästes de artiklarna som utifrån sammanfattningen möjligen kunde vara relevanta för studiens syfte. Översiktsläsningen innebar en snabb läsning med fokus på metod och resultat, för att få reda på åldern på eleverna som ingick i studien och studiens fokus, till exempel om den handlade om problemlösning inom matematik eller mer generell problemlösning. De urvalskriterier som presenteras i 4.3.2 användes för att bestämma om en artikel skulle tas med i litteraturstudien.

Många av artiklarna påträffades i flera av databaserna och med flera alternativa kombinationer av sökord. Totalt var det 94 olika artiklar som lästes översiktligt. Resultatet av översiktsläsningen blev att 15 artiklar bedömdes som intressanta utifrån studiens syfte och frågeställningar. En av artiklarna fanns inte tillgänglig på internet som fulltext men kunde beställas i tryckt form. Efter att den hade lästs igenom valdes den också bort på grund av brist på koppling till matematisk problemlösning. Två andra artiklar valdes bort efter en mer noggrann granskning eftersom de undersökte en mer generell problemlösning och därför inte visade sig ha tillräcklig koppling till matematik. Det blev slutligen tolv artiklar som valdes

för inkludering i studien. Två av dessa artiklar, Hussein, Lindh och Shukur (2006) och Lindh och Holgersson (2007), baseras på samma studie.

Tabell 2: Sökträffar med olika sökord i Summon¹

Sökord	Antal träffar	Urval utifrån titel	Urval utifrån abstrakt	Urval från översiktlig läsning	Urval till studien
”Programmering i skolan”, problemlösning	1	1	0	0	0
Programmering, problemlösning	3	1	0	0	0
”Programmering i skolan”, matematik	2	2	1	0	0
Programmering, skolan, matematik	2	1	1	0	0
Program*ing, “problem solving”, “mathematics education”	505 ²	70	28	5	5
”Story problems”, program*ing	99	35	3	0	0
“word problems”, program*ing, “mathematics education”	157	46	3	3	3
Program*ing, school, mathematics	2377 ²	29	13	3	3

Tabell 3: Sökträffar med olika sökord i ERIC³

Sökord	Antal träffar	Urval utifrån titel	Urval utifrån abstrakt	Urval från översiktlig läsning	Urval till studien
Program*ing, “problem solving”, “mathematics education”	25	19	12	3	3
”Story problems”, program*ing	5	1	0	0	0
”Word problems”, program*ing	0	0	0	0	0
Program*ing, school, mathematics	102	86	28	4	4

Tabell 4: Sökträffar med olika sökord i Teacher’s Reference Center

Sökord	Antal träffar	Urval utifrån titel	Urval utifrån abstrakt	Urval från översiktlig läsning	Urval till studien
Program*ing, “problem solving”, “mathematics education”	22	20	5	2	2

¹ Extra avgränsningar valde i Summon vid sökningar på engelska: Peer-review, Tidsskriftsartikel, Branschtidningsartikel, Tidskrift/e-tidskrift. Ämnesord: Computer programming, Programming, Mathematics education, Mathematics instruction, Problem solving.

² Första 200 träffar granskade.

³ Extra avgränsningar i ERIC: Education level: alla förutom preschool/kindergarten och post-secondary, higher, adult education.

”Story problems”, program*ing	2	1	1	0	0
”Word problems”, program*ing	4	1	1	0	0
Program*ing, school, mathematics	89	54	15	2	2

Tabell 5: Sökträffar med olika sökord i Libris

Sökord	Antal träffar	Urval utifrån titel	Urval utifrån abstrakt	Urval från översiktlig läsning	Urval till studien
”Programmering i skolan”, problemlösning	0	0	0	0	0
Programmering, problemlösning	16	1	1	0	0
”Programmering i skolan”, matematik	1	0	0	0	0
Programmering, skolan, matematik	2	1	0	0	0
Program*ing, “problem solving”, “mathematics education”	4	0	0	0	0
Program*ing, mathematics education	38	1	0	0	0
”Story problems”, program*ing	12	0	0	0	0
Program*ing, school, mathematics	44	0	0	0	0

Tabell 6: Sökträffar med olika sökord i DIVA

Sökord	Antal träffar	Urval utifrån titel	Urval utifrån abstrakt	Urval från översiktlig läsning	Urval till studien
”Programmering i skolan”, problemlösning ⁴	1	0	0	0	0
Programmering, problemlösning ⁴	5	1	1	0	0
Programmering, skolan, matematik ⁴	0	0	0	0	0
Program*ing, “problem solving”, mathematics education ⁴	1	0	0	0	0
”Story problems”, program*ing ⁴	0	0	0	0	0
”Word problems”, program*ing ⁴	0	0	0	0	0
Program*ing, school, mathematics ⁵	6	2	0	0	0

Tabell 7 visar sökträffar baserade på följande söksträng: ("problem solving") AND ((scratch) OR (kodu) OR (logo) OR (agentcube) OR (agentsheet) OR (Alice) OR (blockly) OR (“game maker”) OR (microworlds) OR (“Robot karol”) OR (etoys)

⁴ Avgränsning i DIVA: Forskningspublikationer.

⁵ Avgränsningar i DIVA: Forskningspublikationer. Ämneskategori Utbildningsvetenskap.

OR (BlueJ) OR (greenfoot) OR (“Java’s cool”) OR (jeroo) OR (lego) OR (“raspberry pi”)) AND ((student) OR (pupil)). DIVA kunde inte behandla en så lång logisk sträng och därför söktes igen alla “OR” kombinationer manuellt. Orden *problem solving* och *pupil* söktes både på engelska och på svenska. Inga artiklar påträffades med de svenska sökorden förutom i SUMMON. Därför presenteras inte de övriga svenska sökningarna i Tabell 7.

Tabell 7: Sökningar med specifika programmeringsspråk som sökord

Databas	Antal träffar	Urval utifrån titel	Urval utifrån abstrakt	Urval från översiktlig läsning	Urval till studien
SUMMON Sökning på engelska ⁶	1121	53	13	7	5
SUMMON Sökning på svenska ⁶	3	0	0	0	0
ERIC ⁷	26	21	12	8	6
Teacher’s Reference Center	22	10	7	2	2
LIBRIS	1	1	0	0	0
DIVA (“problem solving” AND lego AND (student OR pupil) ⁸	3	3	3	2	2

4.4 Sökresultat

Tolv artiklar bedömdes vara intressanta utifrån syftet och frågeställningarna. De beskriver studier som behandlar hur programmeringsundervisning påverkar elevers problemlösningsförmåga och de elevaktiviteter som användes inom programmeringsundervisning som är kopplad till matematisk problemlösning. Dessa tolv artiklar valdes för inkludering i studien. I följande avsnitt presenteras innehållet och kvalitetsgranskningen av de valda artiklarna. Utförandet av innehållsanalysen och kvalitetsgranskning beskrivs.

4.4.1 Utvald litteratur

I detta avsnitt presenteras en översikt av artiklarna som valdes för inkludering i studien (Tabell 8). De flesta studierna är utförda i USA men även Sverige, Kanada, Australien, Nya Zeeland och Turkiet finns representerade. Den största studien är utförd i Sverige.

Av de tolv artiklarna som inkluderas i studien finns en som är en litteraturoversikt (Aydin 2005). Sex artiklar beskriver kvalitativa studier av en mindre grupp elever (upp till en klass). Fem artiklar beskriver studier som använde blandade metoder med kvantitativa element. Två av dessa (Hussain, S., Lindh, J., Shukur, G., 2006, Lindh, J., Holgersson, T., 2007) beskriver en studie av ett större antal elever (322 i interventionsgrupperna).

⁶ Extra avgränsningar valde i Summon: Peer-review, Tidsskriftsartikel, Branschtidningsartikel. Ämnesord: Programming, Mathematics education, Mathematics instruction, Problem solving.

⁷ Extra avgränsningar i ERIC: Education level: alla förutom preschool/kindergarten och post-secondary, higher, adult education.

⁸ Avgränsning i DIVA: Forskningspublikationer.

Tabell 8: Utvalde artiklar

Författare	År	Titel	Programmeringsspråk	Publikation	Land
Ardito, G., Mosley, P., Scollings, L.	2014	WE, ROBOT. Using Robotics to Promote Collaborative and Mathematics Learning in a Middle School	LEGO Mindstorm	Middle Grades Research Journal	USA
Aydin, E.	2005	The Use of Computers in Mathematics Education: A Paradigm Shift from “Computer Assisted Instruction” towards “Student Programming”	LOGO, BASIC	The Turkish Online Journal of Educational Technology	Turkiet
Calder, N.	2010	Using Scratch: An Integrated Problem-solving Approach to Mathematical Thinking	Scratch	Australian Primary Mathematics Classroom	Nya Zeeland
Castledine, A.-R., Chalmer, D.	2011	LEGO Robotics: An authentic problem solving tool?	LEGO Mindstorm	Design and Technology Education: An International Journal	Australien
Feurzeig, W., Papert, S., Lawler, B.	2011	Programming-languages as a conceptual framework for teaching mathematics	LOGO	Interactive Learning Environments	USA
Grubbs, M.	2013	Robotics Intrigue Middle School Students and Build STEM Skills	Robotics generellt	Technology and Engineering Teacher	USA
Hussain, S., Lindh, J., Shukur, G.	2006	The effect of LEGO Training on Pupils’ School Performance in Mathematics, Problem Solving Ability and Attitude: Swedish Data	LEGO Dacta	Educational Technology & Society	Sverige
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	Kodu	International Journal of Technology in Mathematics Education	Kanada
Lindh, J., Holgersson, T.	2007	Does lego training stimulate pupils’ ability to solve logical problems?	LEGO Dacta	Computers and Education	Sverige
Martínez Ortiz, A.	2015	Examining Students’ Proportional Reasoning Strategy Levels as Evidence of the Impact of an Integrated LEGO Robotics and Mathematics Learning Experience	LEGO	Journal of Technology Education	USA

Författare	År	Titel	Programmeringsspråk	Publikation	Land
Parker, T.	2012	ALICE in the Real World	ALICE	Mathematics Teaching in the Middle School	USA
Ratcliff, C., Anderson, S.	2011	Reviving the Turtle: Exploring the Use of Logo with Students with Mild Disabilities	LOGO	Computers in the Schools	USA

4.4.2 Kvalitet i den utvalda litteraturen

Eriksson Barajas et al. (2013) beskriver flera sätt att värdera studiers kvalitet, gällande både kvantitativa och kvalitativa studier. Ett sätt är genom en gradering av evidensstyrka baserat på om effekten som rapporteras i studier bedöms ligga nära den verkliga effekten och hur tillförlitlig forskningen är (s. 120). Faktorer som påverkar värderingen av kvalitativa studier är beskrivningar av miljö och kontext, metod och analys. Även urvalsstrategi och diskussion av motsägande argument är viktiga i en värdering av studiens kvalitet. I denna litteraturstudie har artiklarna granskats utifrån att metod och analys finns tydligt beskrivna, om studien bedöms vara reliabel och valid, och om en balanserad diskussion av metod och resultat inkluderas. Vid analys av artiklarna som inkluderas i litteraturstudien har kommentarer om studiernas kvalitet antecknats.

Ingen av de utvalda artiklarna diskuterar etiska aspekter. Detta kan ses som en brist i kvalitet men eftersom alla artiklar brister på denna punkt kan det antas att det anses praxis att utelämna en etisk diskussion, kanske på grund av begränsat tryckutrymme.

Alla artiklar som inkluderas har genomgått oberoende granskning vilket ska garantera en viss kvalitetsnivå. Det finns dock skillnader i den bedömda kvaliteten i de olika artiklarna som diskuteras i 6.1 *Metoddiskussion*.

4.4.3 Analys av de utvalda artiklarna

En systematisk innehållsanalys av artiklarna har genomförts (Eriksson Barajas et al. 2013, s. 164), något som är nödvändigt för att behålla objektivitet i sammanställningen och presentationen av resultaten.

Denna studie har som syfte att få kunskap om hur undervisning i programmering påverkar elevers problemlösningsförmåga och om vilka elevaktiviteter som används inom denna undervisning. För att få svar på den första frågeställningen ter det sig rimligt att utgå både ifrån problemlösningsförmågan generellt och ifrån faktorer som påverkar problemlösningsförmågan. Dessa påverkansfaktorer presenteras i 3.2 *Problemlösning i matematik* och användas för att forma kategorier för innehållsanalysen. För att få svar på den andra frågeställningen har läsning utgått från en öppen fråga och kategorierna bestämts utifrån det som presenteras i artiklarna. Anteckningar över vilka aktiviteter som beskrivs i artiklarna har förts och sedan sammanförts till kategorier efter att analysen av alla artiklar har genomförts.

Utifrån den första frågeställningen och Mölleheds (2001, s. 160) påverkansfaktorer för problemlösning som presenteras i bakgrunden, bestämdes analyskategorier. Till kategorierna hör matematiska faktorer, kognitiva faktorer och övriga faktorer inklusive affektiva faktorer. Som beskrivs i bakgrunden ovan är de matematiska faktorerna räkneförmåga, noggrannhet, förståelse av matematiska begrepp, förståelse av samband mellan storheter, förståelse av samband mellan enheter och talförståelse (s. 102). De kognitiva faktorerna är separation (att undvika att blanda samman olika ting), förståelse av proportionalitet och konstans eller beständighet, kombinationsförmåga, förståelse av relationer mellan helheten och delar, logiskt tänkande, textförståelse, visualiseringsförmåga och verklighetsuppfattning (s. 99). Övriga faktorer är koncentrationsförmåga, tidigare erfarenheter och kunskaper och affektiva faktorer som motivation, attityder, känslor och värderingar (s. 103).

Dessa kategorier, med tillägg av en kategori som täcker problemlösningens förmågan generellt, användes vid analys av artiklarnas innehåll för att få svar på den första frågeställningen. Analysen genomfördes genom att resultaten som presenteras i artiklarna lästes noggrant. Anteckningar fördes när ett resultat som kunde kopplas till en av kategorierna hittades i någon av artiklarna. Ett kodningssystem har använts för att presentera centrala resultat från de olika artiklarna (Eriksson Barajas et al. 2013, s. 163).

Följande kategorier har använts vid analys av artiklarna för att få svar på den första frågeställningen:

Frågeställning 1: Hur påverkar undervisning i programmering elevernas förmåga att lösa matematiska problem?

- 1A: Påverkan på elevernas problemlösningens förmåga generellt
- 1B: Påverkan på matematiska faktorer
- 1C: Påverkan på kognitiva faktorer
- 1D: Påverkan på övriga faktorer

Vid analys av artiklarna för att finna svar på den andra frågeställningen har en mer öppen analys genomfördes. Alla elevaktiviteter som nämndes i artiklarna nedtecknades och sammanställdes i kategorier. De kategorier som använts för kategorisering och presentationen av resultat som svara på den andra frågeställningen presenteras här:

Frågeställning 2: Med hjälp av vilka elevaktiviteter genomförs programmeringsundervisning som är kopplad till matematisk problemlösning?

- 2A: Styrning av robotar för att lösa fysiska problem
- 2B: Lösning av lärarbestämda och strukturerade problem
- 2C: Planering och utvecklande av spel eller egna matematiska problem
- 2D: Självstyrda undersökningar av verktyget
- 2E: Kollaborativt arbete
- 2F: Lära ut matematik till andra

Resultaten av analysen och kvalitetsgranskningen presenteras i Tabell 9. Där en kod registreras för en artikel betyder det att artikeln presenterar något bevis eller observation av en positiv påverkan med hänsyn till den faktorn. Där en kod presenteras inom parentes betyder det att faktorn studerats men att inget samband upptäcktes eller att ett negativt samband mellan programmeringsundervisning och faktorn har upptäckts. Tabellen är ordnat efter programmeringsspråk.

Tabell 9: Sammanfattning av artiklarnas resultat och kvalitet (formatet tolkat från Eriksson Barajas et al. 2013, s. 162)

Författare	Programmerings-språk	Studiedesign	Population Bortfall Tidsram	Undervisning Intervention (I) Kontroll (K)	Resultat Kod utan parentes indikerar positiv påverkan på elever Kod i parentes () indikerar negativ påverkan/inget samband	Kommentarer på studiekvalitet
Aydin, E. (2005)	LOGO, BASIC	Historisk bakgrund, litteraturstudie	Redovisas inte	-	1A problemlösningförmåga 1B begreppsförståelse 1C kognitiva förmågor, reflektionsförmåga 2D elevcentrerad, öppna uppgifter	Ingen metodbeskrivning. Språket otydligt ibland. Begränsad diskussion av metod och resultat.
Feurzeig, W., Papert, S., Lawler, B. (2011)	LOGO	Kvalitativ. Fallstudie med klassrums-observationer och teoretisk diskussion.	En klass i grade 7 (12-13 år) Sommarkurs	I: träning av elever med produktion av algoritmer, rekursion	1A problemlösning generellt 1B begreppsförståelse 1C planering, analogi, resonemang, reflektion 1D uthållighet 2B lösa strukturerade matematiska problem 2D självstyrda undersökningar	Skrivit av LOGO språkutvecklarna. Originalstudien är gammal (1967). Inga nackdelar med programmering presenteras.
Ratcliff, C., Anderson, S. (2011)	LOGO	Kvalitativ. Fallstudie. Klassrums-observationer och informella intervjuer med elever och lärare.	9 elever i grade 4 (9-10 år) med inlärnings-svårigheter. 3 x 90 minuters lektioner.	I: träning av elever med produktion av geometriska former	(1A) inget bevis på påverkan på problemlösningförmåga 1C rumsuppfattning, visualisering, resonemang 1D uthållighet, motivation 2B lösa strukturerade matematiska problem 2D självstyrda undersökningar 2E kollaborativt arbete	Metodbeskrivning och diskussion. Balanserad diskussion. Begränsad antal deltagare med inlärnings-svårigheter.
Li, Q., Vandermeiden, E., Lemieux, C., Nahoo, S. (2014)	Kodu, Scratch	Blandade metoder. Före och efter intervention kunskapstester, frågeformulärer, lärarjournaler.	40 elever Intervention: 27 elever Kontroll: 13 elever grades 10-11 (15-18 år) inom ramen för vanlig matematik-undervisning	I: planerade uppgifter, planering av eget spel baserat på bestämda matematiska problem K: självstyrt lärande i ett matematiskt område	1A problemlösningförmåga 1B begreppsförståelse 1C olika möjliga lösningsstrategier 1D uthållighet, motivation 2C planering och utvecklande av spel 2E kollaborativt arbete 2F lära ut matematik till andra	Begränsat antal deltagare. Oberoende kodning av kvalitativ data av fyra personer. Balanserad diskussion.

Författare	Programmerings-språk	Studiedesign	Population Bortfall Tidsram	Undervisning Intervention (I) Kontroll (K)	Resultat Kod utan parentes indikerar positiv påverkan på elever Kod i parentes () indikerar negativ påverkan/inget samband	Kommentarer på studiekvalitet
Calder, N. (2010)	Scratch	Kvalitativ. Fallstudie. Elevers skriftliga reflektioner, elev- och lärarintervjuer, klassrums- observationer.	26 elever i årskurs 6 (9-10 år) Två veckor	I: strukturerade programmerings- uppgifter, eget spel för att lära ut matematik till yngre elever	1A problemlösning generellt 1B begreppsförståelse 1C planeringsförmåga, rumsuppfattning, resonemangsförmåga, reflektionsförmåga 1D motivation 2C planering och utvecklande av spel 2E kollaborativt arbete 2F lära ut matematik till andra	Detaljerad metodbeskrivning. Liten undersökning. Inga nackdelar presenteras.
Parker, T. (2012)	ALICE	Kvalitativ. Klassrums- observationer.	En klass i grade 5 (10-11 år), fem dagar	I: Strukturerade programmerings- uppgifter, elev valde spel	1A problemlösningsförmåga 1C olika möjliga lösningsstrategier 1D uthållighet, motivation 2C planering och utvecklande av spel 2D självstyrda undersökningar	Liten undersökning. Inga nackdelar presenteras.
Grubbs, M. (2013)	Robot- program- mering generellt	Kvalitativ. Metoden presenteras inte. Troligen klassrums- observationer, diskussion med elever och lärare.	Ospecificerad <i>middle school</i>	I: Arbete med att bygga och programmera en robot	1B förståelse av matematiska begrepp 1C analogi mellan olika problem 1D motivation 2A styrning av robotar 2B lösning av strukturerade problem 2E kollaborativt arbete	Ingen metodbeskrivning. Inga nackdelar presenteras.

Författare	Programmerings- språk	Studiedesign	Population Bortfall Tidsram	Undervisning Intervention (I) Kontroll (K)	Resultat Kod utan parentes indikerar positiv påverkan på elever Kod i parentes () indikerar negativ påverkan/inget samband	Kommentarer på studiekvalitet
Ardito, G., Mosley, P., Scollings, L. (2014)	Turtle Art/ LEGO Mindstorm s	Blandade metoder. Blog för elevreflektion med innehålls- analys av texterna, lärarintervjuer, planerings- dokument, klassrums- observationer, matematik- provresultat.	I: En klass i grade 6 (11-12 år), fyra månader K: Andra grade 6 klasser på samma skola och i samma delstat.	I: Planering och organisation av nyckelbegrepp i teman. Arbete med robotar med syfte att se kopplingar mellan matematiska begrepp och arbetet med robotar, utvecklande av problemlösnings- och samarbetsförmågor. 14 veckor. Stöd från universitetsstudenter. K: Begränsat arbete med robotar, annars oklart.	1A problemlösningsförmåga 1B förståelse av matematiska begrepp 1C analogi mellan olika problem, rumsuppfattning 1D motivation 2A styrning av robotar 2B lösning av strukturerade problem 2D självstyrda undersökningar 2E kollaborativt arbete	Relativt balanserad diskussion. Detaljerad metodbeskrivning dock oklart hur mycket andra klasser arbetade med robotar eller vad de gjorde som alternativ.
Castledine, A.-R., Chalmer, D. (2011)	LEGO Mindstorm	Kvalitativ. Fallstudie. Observationer och elev fråge- formulär.	23 elever i grade 6 (11-12 år) Två veckor med en en- timmes lektion varje dag.	I: Bygga robotar och programmera dem för att lösa två problem.	1A problemlösningsförmåga 1C planering och val av strategi, reflektion (1C) analogi mellan olika problem 2A styrning av robotar 2B lösning av strukturerade problem 2E kollaborativt arbete	Liten undersökning. Balanserad diskussion.

Författare	Programmerings-språk	Studiedesign	Population Bortfall Tidsram	Undervisning Intervention (I) Kontroll (K)	Resultat Kod utan parentes indikerar positiv påverkan på elever Kod i parentes () indikerar negativ påverkan/inget samband	Kommentarer på studiekvalitet
Hussain, S., Lindh, J., Shukur, G. (2006)	LEGO Dacta	Blandade metoder. Observationer, intervjuer, lärar- dokumentation. Frågeformulärer. Före- och efter- intervention kunskapstest i matematik och problemlösning.	I: 12 klasser 322 elever 193 i årskurs 6 (12-13 år) 129 i årskurs 9 (15-16 år) 2 timmar varje vecka i ett år K: 12 klasser 374 elever 169 åk 6 205 åk 9	I: Bygga robotar och programmera K: Oklart vad eleverna gjorde under motsvarande skoltiden	(1A) negativ påverkan på problemlösningsförmågan (åk 6) 1D motivation i åk 6 och 9 (engagemang bedömde av lärare). Ökad självförtroende (åk 6). (1D) ingen generell positiv attityd mot LEGO arbete (elevskattning) 2A styrning av robotar 2B lösning av strukturerade problem 2E kollaborativt arbete	Stor studie över lång tidsperiod. Detaljerad metodbeskrivning dock saknas en beskrivning av undervisningen som kontrollgruppen deltog i. Definition av ”problemlösning” saknas. Språket otydligt ibland. Balanserad diskussion.
Lindh, J., Holgersson, T. (2007)	LEGO Dacta	Samma studie som Hussain (2006). Blandade metoder. Observationer, intervjuer, lärar- dokumentation. Frågeformulärer. Före- och efter- intervention kunskapstest i matematik och problemlösning.	I: 12 klasser 322 elever 193 i årskurs 6 (12-13 år) 129 i årskurs 9 (15-16 år) 2 timmar varje vecka i ett år K: 12 klasser 374 elever 169 åk 6 205 åk 9	I: Bygga robotar och programmera K: Oklart vad eleverna gjorde under motsvarande skoltiden	(1A) ingen påverkan på problemlösningsförmågan, med respekt till helklass och delade grupper jämfört med resultat av före intervention testresultat 2A styrning av robotar 2E kollaborativt arbete	Stor studie över lång tidsperiod. Detaljerad metodbeskrivning dock saknas en beskrivning av undervisningen som kontrollgruppen deltog i. Definition av ”problemlösning” saknas. Språket otydligt ibland. Balanserad diskussion.
Martínez Ortiz, A. (2015)	LEGO/ Robolab	Blandade metoder. Före och efter intevention kunskapstest, fördröjt eftertest i matematik, problemlösning och teknik (10 veckor)	I: 15 elever i grade 5 (10-11 år) K: 15 elever. 15 timmar under en vecka (frivillig lovskola)	I: Integrerad teknik och matematik undervisning i förhållande och proportionalitet med LEGO robotar K: Läroboksbaserad undervisning i proportionalitet	1A problemlösningsförmåga (inom proportionalitet) 1B förståelse av matematiska begrepp (1C) resonemangsförmåga (inom proportionalitet) 2A styrning av robotar Oklart vilka andra aktiviteter ingick i undervisningen.	Liten grupp. Kontrollgrupp, fördröjt eftertest. Vissa resultat presenteras i diskussionen. Begränsningar av studien diskuteras.

5 Resultat

I detta avsnitt presenteras resultaten av litteraturstudien utifrån kategorierna som presenterades i 4.4.3 *Analys av de utvalda artiklarna* ovan. Resultat som svarar på de första och andra frågeställningarna presenteras var för sig och sammanfattas sedan avslutningsvis.

5.1 Hur påverkar undervisning i programmering elevernas förmåga att lösa matematiska problem?

I detta avsnitt presenteras resultat som svarar på litteraturstudiens första frågeställning om hur undervisning i programmering påverkar elevernas förmåga att lösa matematiska problem. Många av artiklarna ger en begränsad eller ingen definition av vad de avser med ”problemlösning” eller ”problemlösningsförmåga”. Därför har resultaten tolkats utifrån kontexten som presenteras i artiklarna. Resultaten presenteras utifrån analyskategorierna. Först presenteras resultat kring påverkan på problemlösningsförmåga generellt, sedan presenteras resultat kring påverkan på matematiska faktorer, kognitiva faktorer och övriga faktorer.

5.1.1 Påverkan på elevernas matematiska problemlösningsförmåga generellt

Alla studier har undersökt påverkan av programmeringsundervisning på elevernas matematiska problemlösningsförmåga och alla förutom Grubbs (2013) presenterar resultat som handlar om en generell matematisk problemlösningsförmåga. Forskarna har studerat olika programmeringsspråk och aktiviteter. I artiklarna uttrycker många av forskarna ett positivt samband mellan elevernas problemlösningsförmåga och deltagande i programmeringsundervisning (Aydin 2005; Feurzeig, Papert och Lawler 2011; Li, Vandermeiden, Lemieux och Nahoo 2014; Calder 2010; Parker 2012; Ardito, Mosley och Scollings 2014; Castledine och Chalmer 2011; Martínez Ortiz 2015). Andra forskare har undersökt men inte hittat något samband mellan problemlösningsförmåga och deltagande i programmeringsundervisning (Ratcliff och Anderson 2011; Hussain et al. 2006; Lindh och Holgersson 2007).

Ardito et al. (2014) jämförde provresultat efter intervention i ett matematikprov. Jämförelsen var mellan en interventionsklass och andra klasser i samma årskurs (11-12 åringar) på samma skola och i hela New York State (s. 77). I interventionsklassen arbetade eleverna med att programmera robotar med LEGO *Mindstorms* under en 14-veckorsperiod. Eleverna fick under robotprojektet även mentorstöd från högskolestudenter. Läraren i klassen var målmedveten och tydlig med syftet med arbetet med robotar. Syftet var att tydliggöra organisationen av matematiska arbetsområden, att koppla matematiska förmågor till robotarbetet och att främja elevernas problemlösnings- och samarbetsförmåga (s. 77). Ingen statistiskt signifikant skillnad i elevernas provresultat fanns i jämförelse med kontrollgrupperna, men interventionsklassen hade färre elever som inte nådde godkände nivå jämfört med kontrollgrupperna (s. 79). Ardito et al. (2014, s. 79, 85) förklarar att resultaten från prov i matematik reflekterar lärarens observationer att eleverna utvecklade sin problemlösningsförmåga under arbetet med robotar. Martínez Ortiz (2015) undersökte en grupp bestående av 15 elever som arbetade med LEGO-robotar för att utveckla förmågor inom det matematiska området proportionalitet. Dessa elever jämfördes med en kontrollgrupp som fick

läroboksbaserad undervisning i samma arbetsområde och från samma lärare. Interventionsgruppen fick även undervisning kring en design- och planeringsprocess inom teknik. Deltagare deltog frivilligt i en veckas extra matematikundervisning under en lovperiod. Martínez Ortiz (2015) rapporterar en signifikant skillnad i testresultat från textbaserade matematiska problem om proportionalitet för elever som fick undervisning i LEGO-programmering och designprocessen, i jämförelse med elever som fick läroboksbaserad undervisning. Skillnaden i resultat bestod även vid ett fördröjt test som genomfördes tio veckor efter interventionen.

Feurzeig et al. (2011, s. 491) redogör för en äldre forskningsartikel där en fallstudie med programmeringsspråket LOGO genomfördes i en klass under en månad, med syfte att utveckla elevers förståelse av matematiska begrepp och algoritmer genom programmering. Författarna argumenterar för en överföring av problemlösningsförmågor från datorprogrammering till matematisk problemlösning (s. 499). Även Aydin (2005, s. 30) förklarar i sin litteraturstudie att processen av att planera, skriva, utveckla och felsöka datorprogram kan ge eleverna möjlighet att utveckla sin problemlösningsförmåga. Han menar vidare att forskning under 80-talet om LOGO-programmering fann en signifikant påverkan på elevernas problemlösningsförmåga efter arbete med LOGO (s. 31).

Li et al. (2014) undersökte 27 elever i åldern 15-18 år som byggde spel i *Kodu*- och *Scratch*-programmeringsmiljöerna för att lära ut matematik till andra elever. Li et al. drar slutsatsen, från intervjuer med elever och observationer, att programmeringsarbete gjorde att eleverna utvecklade sin problemlösningsförmåga (s. 29). Calder (2010, s. 13) observerade i sin forskning en klass av 9–10-åriga elever som byggde spel för yngre elever med *Scratch* under en veckas tid. Läraren bekräftade att efter arbete med *Scratch* hade eleverna utvecklat sin problemlösningsförmåga. Under en vecka undervisade Parker (2012) i programmeringsmiljön *Alice* i en klass med 10–11-åriga elever, som ett arbetsområde inom problemlösning. Eleverna använde programmeringsmiljön för att skapa en berättelse med ett övergripande problem som skulle lösas. Samtidigt löste de konkreta problem för att programmera sina berättelser. Han beskriver hur eleverna utvecklade avancerade problemlösningsförmågor (s. 415). Castledine och Chalmer (2011, s. 26) observerade en klass 10–11-åriga elever som byggde och programmerade robotar en timme varje dag under två veckors tid. Efter observationerna drog de slutsatsen att arbete med robotar och LEGO-programmering är ett effektivt sätt att utveckla elevers problemlösningsförmåga.

Ratcliff och Anderson (2011) undersökte en grupp bestående av nio elever med inläringssvårigheter och beskriver svårigheterna med att ta reda på vilka inlärningsprocesser och kognitiva processer som sker under programmeringsprocessen (s. 253). De menar att trots att LOGO inte helt uppfyller Paperts ursprungliga intentioner gällande ökad problemlösningsförmåga kan programmering ändå vara en meningsfull aktivitet för vissa elever (s. 252). Hussain, Lindh och Shukur (2006), samt Lindh och Holgersson (2007) analyserade en undersökning av tolv svenska klasser, med totalt 322 elever, fördelade mellan årskurs 6 och 9. Det är den största undersökningen som ingår i litteraturstudien och den pågick under ett års tid. De hittade inget bevis att träning med LEGO Dacta programmering och robotar hade någon positiv påverkan på elevernas

förmåga att lösa problem som liknar de som förekommer på de svenska nationella proven. Enligt analysen som gjordes av Hussain et al. (2006, s. 190) finns även en liten negativ påverkan på eleverna i årskurs 6 avseende problemlösningsförmåga efter deltagande i programmeringsundervisning i jämförelse med kontrollgrupperna.

5.1.2 Påverkan på matematiska faktorer

Många av artiklarna rapporterar en ökning av elevernas förståelse av vissa matematiska begrepp under arbetet med programmering. Aydin (2005, s. 30) lyfter programmeringsnyttan utifrån en konstruktivistisk syn på lärande och menar att programmering underlättar för barnen att undersöka matematiska begrepp. Även Feurzeig et al. (2011, s. 498) menar att programmering kan användas som ett kraftfullt verktyg för att undersöka nya matematiska begrepp. Dessa påståenden bekräftas av resultaten i de olika studierna (Martínez Ortiz 2015; Li et al. 2014; Grubbs 2013; Ardito et al. 2014; Calder 2010). Studierna har haft olika matematiska fokus och problemlösningsfokus. Martínez Ortiz (2015) fokuserade på elevers förståelse av proportionalitet under arbete med LEGO-robotar i en teknikbaserad undervisningskontext. Han menar att elever som lärde sig om proportionalitet i en teknikbaserad kontext utvecklade en djupare förståelse av proportionalitet och behöll denna utveckling över tid. Eleverna fick högre resultat än kontrollgruppen i ett test som genomfördes 10 veckor efter interventionen, när de skulle lösa textbaserade matematiska problem som handlade om proportionalitet. Grubbs (2013, s. 16) rapporterar om elevers utveckling av en fördjupad förståelse av ekvationer och variabler när de arbetade med att programmera en robot för att köra ett bestämt avstånd. Ardito et al. (2014, s. 79) menar att eleverna utvecklade en fördjupad förståelse av geometriska begrepp som area och omkrets efter arbete med robotar.

Li et al. (2014, s. 29) redogör för hur elever upplevde matematiklärande som svårare (jämfört med vanlig undervisning) vid byggandet av ett matematiskt spel med *Kodu*. *Kodu* är en skärmbaserad programmeringsmiljö som används för att bygga virtuella världar med figurer och objekt som programmeras för att reagera på olika sätt vid olika händelser. Trots upplevelsen att matematiklärandet var svårare, kände eleverna att de förstod mer eftersom de var tvungna att fördjupa sin förståelse av matematiska begrepp för att kunna bygga spelet. Calder (2010, s. 12-14) förklarar att de flesta aktiviteterna som eleverna gjorde med *Scratch*, en annan skärmbaserad programmeringsmiljö, krävde användning av matematiska begrepp. Han lyfter specifikt vinklar, avstånd och tid som områden som eleverna arbetade med. Han menar att den omedelbara återkopplingen som fanns under programmeringsprocessen underlättade för elevernas undersökningar och utveckling av förståelse.

Vissa artiklar redovisar en påverkan på elevernas resultat i matematikprov men beskriver inte i vilka aspekter av den matematiska förmågan resultaten påverkades. Lindh och Holgersson (2007, s. 1105) menar att vissa elevgrupper (de elever som hade medelgoda resultat före interventionen) fick en ökad matematisk förmåga i jämförelse med kontrollgrupperna. Detta påvisades på ett matematikprov som genomfördes efter arbetet med LEGO-robotar. Ardito et al. (2014, s. 79) rapporterar att andelen elever som inte nådde godkänd nivå i ett New York State

matematikprov var något mindre i klassen som arbetade med LEGO-robotar än kontrollgrupperna, men skillnaden var inte statistiskt signifikant. Martínez Ortiz (2015) hittade ingen statistisk skillnad i resultat på rena matematiska frågor mellan interventions- och kontrollgrupperna efter arbete med robotar.

5.1.3 Påverkan på kognitiva faktorer

Artiklarna presenterar många resultat som handlar om påverkan på kognitiva faktorer. Dessa faktorer kan delas upp i fyra kategorier: påverkan på elevernas förmåga att planera och välja strategier vid problemlösning, påverkan på rumsuppfattning och visualiseringsförmåga, resonemangsförmåga samt förmågan att reflektera.

Påverkan på elevernas förmåga att planera och välja strategier

Feurzeig, Papert och Lawler (2011) menar att problemlösning måste byggas på en plan som kan förändras och förbättras. De förklarar att genom programmering kan eleverna lära sig hur man konstruera och utvecklas en problemlösningssplan. Detta bekräftas genom observationer rapporterade i de olika artiklarna (Calder 2010; Li et al. 2014; Parker 2012; Castledine och Chalmer 2011). Calder (2010, s. 11) beskriver hur elever lärde sig av försök och misstag för att kunna förbättra sina program i *Scratch*. Han menar att programmeringsarbetet ledde till att eleverna utvecklade sin förmåga att generalisera. Han observerade att de utvärderade problemen och tänkte logiskt för att förändra delmål för att komma framåt i problemlösningssprocessen (s. 13). Li et al. (2014) undersökte spelbyggande med *Kodu* och *Scratch* och återger en elevs, Marylenes, beskrivning av spelbyggnadsprocessen. Marylene förklarar att ”spelbygge hjälpte mig att förstå det matematiska arbetsområdet. Till exempel var jag tvungen att tänka på i vilken ordning frågorna skulle komma och det påverkade min upplevelse av matematik. Frågorna behövde bygga på varandra” (s. 29). Marylene var tvungen att planera sin programmering så att det skulle fungera på ett pedagogiskt sätt. Parker (2012, s. 413) beskriver hur eleverna blev tvungna att använda olika strategier för att lära sig hur de kunde programmera i *Alice* för att nå de önskade resultaten i den virtuella världen som de byggde. Han menar att eftersom eleverna förstod att de måste misslyckas för att slutligen lyckas uppmuntrades de till att testa olika strategier (s. 414). Även Castledine och Chalmer (2011, s. 24) beskriver elevernas användning av olika strategier för problemlösning vid arbete med robotar. Eleverna utvecklade och använde sig av matematiska strategier för att på ett effektivt sätt lösa problem med robotar som försök och misstag, att mäta, uppskatta avstånd och hitta mönster. Castledine och Chalmer (2011, s. 24) återger hur eleverna, efter att de hade blivit mer effektiva med programmering av robotarna, började tänka mer strategiskt om hur de kunde förbättra robotarna och programmeringen.

Påverkan på rumsuppfattning och visualiseringsförmåga

Ratcliff och Anderson (2011) och Calder (2010) beskriver utveckling av elevernas rumsuppfattning eller visualiseringsförmåga vid arbete med programmering. Ratcliff och Anderson (2011, s. 249) observerade en elev, Jenny, som hade betydande svårigheter inom rumsuppfattning. Hon hade svårt att uppskatta avstånd men efter ungefär två veckors arbete med LOGO hade hon lyckats med att programmera LOGO för att skapa kvadrater i olika storlekar på skärmen. För Jenny var detta en stor bedrift som uppmärksammades av hennes ordinarie lärare.

Calder (2010) beskriver hur arbetet med *Scratch* engagerade eleverna i arbete med rumsuppfattning, vinklar och avstånd. De kunde till exempel experimentera med olika vinklar på ett sätt som inte skulle vara möjligt utan ett digitalt verktyg (s. 12). Även Ardito et al. (2014, s. 79) observerade förbättringar inom elevernas förmåga att mäta och förstå area och omkrets. Genom arbetet med robotar tränade eleverna visualisering och fick en förbättrad förståelse för avstånd, omkrets och hur de kan mätas.

Påverkan på elevernas resonemangsförmåga

Feurzeig et al. (2011, s. 490, 495) menar att skapandet och användning av de rekursiva processerna som uppstår i programmering med LOGO är ett bra sätt för elever att lära sig de grundläggande principerna inom matematisk resonemang. Calder (2010, s. 13) beskriver hur elever, genom felsökning och experiment för att utveckla sina program i *Scratch*, använde och utvecklade sin resonemangsförmåga för att förklara varför de olika problemen uppstod i programmet och vad de skulle göra för att lösa dem. Calder (2010, s. 14) menar att återkopplingen eleverna får när de programmerar leder dem till att utveckla sin resonemangsförmåga. Li et al. (2014, s. 31) menar att programmeringsprocessen och spelbyggnadsprocessen gör att eleverna blir tvungna att explicit förklara sitt matematiska resonemang. Tyst kunskap tvingas fram till medvetandet. Ratcliff och Anderson (2011, s. 251) beskriver en fallstudie om elever med inlärningssvårigheter som lär sig programmera med LOGO. De återger en observation av en elev, Jacob, som efter reflektion och diskussion med en lärare, får en ökad förståelse av hur programmet han har skapat fungerar och hur han kan förenkla det. Författarna lyfter också en nackdel som de upptäckte under studien. De kallar det för *turtle traps (fällor)*, där eleverna är nöjda bara för att de har skapat någonting och använder inte resonemang för att ändra eller förbättra sina program (s. 249).

Påverkan på elevernas förmåga att reflektera

Aydin (2005, s. 31) lyfter elevernas förmåga att reflektera över sitt arbete som ett område som kan påverkas genom undervisning i programmering. Feurzeig et al. (2011, s. 499) förklarar att, eftersom ett datorprogram är konkret, är det lättare för elever att utveckla det genom en felsökningsprocess där de reflekterar, försöker förstå processerna och förbättrar programmet. Med en algebraisk lösning på papper, skulle eleverna förmodligen kasta arbetet de har gjort och börjar om, om inte de når lösningen på en gång, menar Feurzeig et al. (2011, s. 499). Castledine och Chalmer (2011, s. 26) fokuserar på elevernas metakognition och reflektion under problemlösning med robotar och observerar hur eleverna reflekterar och sedan förbättrar sina program för att lösa specifika problem. Enligt Calder (2010, s. 12) får elever direkt feedback vid programmering, både från programmet och från det som händer på skärmen, och även från andra elever i gruppen som de arbetade med. De reflekterar och gör ändringar och förbättringar i programmet.

5.1.4 Påverkan på övriga faktorer

Resultat som behandlar påverkan på övriga faktorer som påverkar elevers matematiska problemlösningsförmåga fokuserar främst på affektiva faktorer som uthållighet och motivation. Några författare beskriver även hur elevers känslor, som självförtroende, kan påverkas vid arbete med programmering.

Påverkan på elevernas uthållighet

Programmeringsprocessen uppmuntrar elever att fortsätta med problemlösningsprocessen även när den upplevs som svår. Li et al. (2014, s. 30) menar att trots att eleverna ofta upplever programmering som frustrerande, utvecklar de själva svåra problem som de fastnar vid och undersöker. Eleverna reagerar på ett annat sätt till svåra problem när det är de själva som skapar problemen, och de upplever en känsla av tillfredsställelse när de löser problemen (Li et al. 2014, s. 30). En elev beskriver programmeringsarbete som "irriterande roligt!" (Li et al. 2014, s. 28). Ratcliff och Anderson (2011, s. 246) beskriver elevernas frustration vid programmering men förklarar att eleverna fortsätter med undersökandet trots frustrationen. När de lyckas med att lösa problemet blir frustration till glädje och upprymdhet istället. Parker beskriver hur elever accepterar att de kommer att misslyckas under programmeringsprocessen och att misslyckandet leder till förändringar och framgång (2012, s. 414). Feurzeig et al. (2011, s. 499) förklarar att eleverna får en utökad uthållighet på grund av datorprogrammets konkreta natur och menar att eftersom eleverna har skapat programmen är de mer benägna att behålla och utveckla dem istället för att kasta dem och börja om.

Påverkan på elevernas motivation

Många av artiklarna nämner motivation som en av fördelarna med programmering. Ratcliff och Anderson (2011, s. 252-3) beskriver elevers stolthet och deras känsla av att ha skapat någonting själv och att ha övervunnit problem. De menar att användning av LOGO-programmering kan hjälpa till att skapa ett målinriktat och fokuserat beteende hos eleverna i klassrummet. Parker (2012, s. 415) beskriver hur ett arbetsområde i programmering har bemötts med entusiasm från elever vid varje undervisningstillfälle. Ardito et al. (2014, s. 81) beskriver att eleverna som arbetade med robotar ofta var entusiastiska och fokuserade på arbetet. Li et al. (2014, s. 28) menar att eleverna fick ett nytt perspektiv på matematikämnet när de arbetade med programmering och problemlösning, och att det blev roligare och mer intressant för dem. Calder (2010, s. 13) beskriver programmeringsmiljön *Scratch* som en miljö som skapar motivation.

Programmering ger möjlighet till individualisering av och variation inom matematikarbete, vilket skulle kunna påverka elevernas motivation för problemlösning. Enligt Parker (2012, s. 415) kan programmeringsarbete underlätta individualisering av undervisning. Han beskriver hur elever som vanligtvis hade svårigheter inom matematikämnet utvecklade de mest effektiva och komplexa programmen. Li et al. (2014, s. 31) lyfter också elevernas möjlighet att välja och påverka hur de arbetar som en fördel av programmeringsarbete.

Påverkan på elevernas engagemang och självförtroende

Hussain et al. (2006, s. 190) rapporterar en högre nivå av engagemang och även självförtroende hos de elever som arbetade med LEGO-robotar i jämförelse med eleverna i kontrollgruppen. Engagemanget hos eleverna är skattat av lärarna. De upptäckte dock att det generellt inte fanns en positiv attityd bland eleverna för LEGO-arbetet (s. 192). Det står i motsats till resultaten som Li et al. (2014, s. 28) presenterar om sin undersökning om spelbaserat lärande i matematik, där minst 74% av eleverna i alla grupper var positiva till spelbaserat lärande. Castledine och

Chalmer (2011, s. 25) beskriver hur alla elevergrupper i studien hade högt förtroende för att de skulle lyckas med nya problemlösningssuppgifter i LEGO.

Påverkan på elevernas syn på matematikens relevans

Programmering utgår ofta från konkreta exempel och kan därför användas för att visa elever betydelsen av matematiken de lär sig i klassrummet för verkliga problem och på så sätt öka elevernas upplevelse av skolmatematik som relevant. Feurzeig et al. (2011, s. 490) förklarar att programmering kan användas för att underlätta elevers förståelse för abstrakta matematiska begrepp och bygga upp en förståelse utifrån mer konkreta exempel. Ardito et al. (2014, s. 79) beskriver en studie där 11-12-åriga elever i en klass arbetade med robotar inom matematikundervisning under fyra månader. Efter arbetet med robotar uppskattade deras lärare att de hade utvecklat sin förmåga att relatera matematiska problem som handlade om area och omkrets till problem utanför klassrummet. Grubbs (2013, s. 16) beskriver hur eleverna kom med förslag på alternativa problem i verkligheten som kunde lösas med robotar. Grubbs beskriver även en ökning av elevernas intresse för hur det de lärde sig i skolan kunde tillämpas på verkliga problem. Castledine och Chalmer (2011, s. 26) menar dock att även efter arbetet med robotar kunde eleverna bara relatera problemlösning till autentiska kontexter utanför klassrummet på en grundläggande nivå och behövde ledas av intervjuarna för att komma med förslag till liknande problem i andra kontexter.

5.2 Med hjälp av vilka elevaktiviteter genomförs programmeringsundervisning som är kopplad till matematisk problemlösning?

Artiklarna beskriver olika elevaktiviteter som eleverna ägnar sig åt under arbetet med programmering, som styrning av robotar för att lösa fysiska problem, planering och utvecklande av spel eller egna matematiska problem med skärmbaserade programmeringsspråk, att lära ut matematik till andra, lösning av olika strukturerade problem, självstyrda undersökningar av verktyget och kollaborativt arbete.

Forskningen om programmeringsundervisning som är kopplad till matematisk problemlösning kan delas upp i två kategorier. En kategori innehåller studierna där forskarna undersöker arbetet med programmering för att styra robotar i en verklighetsbaserad kontext. Den andra kategorin innehåller studierna där forskare undersöker skärmbaserad programmering. Grubbs (2013), Ardito et al. (2014), Castledine och Chalmer (2011), Hussain et al. (2006), Lindh och Holgersson (2007) och Martínez Ortiz (2015) fokuserar sitt forskningsarbete på robotar i en verklighetsbaserad kontext. Aydin (2005), Calder (2010), Feurzeig et al. (2011), Li et al. (2014), Parker (2012), och Ratcliff och Anderson (2011) beskriver arbete med skärmbaserade programmeringsspråk.

I de studier som observerade arbete med robotar beskrivs olika problem i fysiska miljöer som elever ska programmera robotarna att navigera igenom (Grubbs 2013, s. 14; Ardito et al. 2014; Castledine and Chalmer 2011). Grubbs (2013, s. 13) beskriver hur elever programmerar robotar för att kunna navigera genom en stad för att göra rent gatorna. Ardito et al. (2014, s. 79) beskriver en utmaning där eleverna programmerar robotarna för att köra upp och ner för korridoren utanför klassrummet. Castledine och Chalmer (2011, s. 22, 24) beskriver utmaningar där

eleverna ska navigera robotarna genom en labyrinth och även köra robotarna så nära ett mål (en LEGO figur) som möjligt utan att knuffa ner det.

I tre av studierna där elever använde skärmbaserade programmeringsspråk (*Kodu*, *Scratch* och *Alice*) skapade eleverna spel genom programmering (Li et al. 2014, Calder 2010, Parker 2012). Av dessa finns det två studier, Li et al. (2014) och Calder (2010) som rapporterar om utveckling av spel som ska användas för att lära ut matematik till andra elever. De redogör för att eleverna upplever denna del av programmering som väldigt lärorikt eftersom de blir tvungna att utveckla en djupare förståelse av de matematiska begreppen som inkluderas i spelen.

Introduktionen till verktygen sker oftast genom arbete med lärarbestämda, strukturerade problem som eleverna ska lösa (Castledine och Chalmer 2011; Hussain et al. 2006; Grubbs 2013; Ratcliff och Anderson 2011; Feurzeig et al. 2011; Ardito et al. 2014). Ratcliff och Anderson (2011, s. 253) redogör för att en strukturerad undervisning verkar mer effektivt för att utveckla lärande än att eleverna själva ska undersöka verktygen utan något bestämt mål. De varnar för *turtle traps* (fällor), vilket betyder att eleverna arbetar med verktygen utan att engageras i resonemang eller förstå målet med arbetet. Andra forskare som observerar elevstyrd utforskning som en del i arbetet med programmering är Aydin (2005), Ardito et al. (2014) och Parker (2012). Aydin (2005, s. 31) förklarar att LOGO ofta används i elevcentrerad undervisning med öppna uppgifter där eleven styr sitt eget lärande. Ardito et al. (2014, s. 78) beskriver att eleverna först fick undersöka programmeringsspråket *Turtle Art* och experimentera och lära av varandra, innan de började arbeta med att programmera robotar. Ardito et al. förklarar att denna utforskningsbaserade undervisningsmetod valdes eftersom den hade visats av tidigare forskning vara effektiv för projektbaserat lärande. Parker (2012, s. 414) använde en del av undervisningstiden till elevernas egen utforskning av programmeringsmiljön *Alice* i kombination med lärarledda genomgångar.

Många av artiklarna redogör för att eleverna har arbetat kollaborativt under programmeringsarbetet (Ardito et al. 2014, s. 78; Calder 2010, s. 10; Castledine och Chalmer 2011, s. 21; Grubbs 2013, s. 13; Hussain et al. 2006, s. 185; Li et al. 2014, s. 26; Lindh och Holgersson 2007, s. 1101). Även i ett klassrum där arbetet inte var organiserat på ett sätt där eleverna skulle arbeta i grupper observerades att eleverna hjälpte varandra och arbetade kollaborativt för att utveckla sina programmeringskunskaper (Ratcliff och Anderson 2011, s. 246).

5.3 Sammanfattning av resultat

Resultaten av litteraturstudien visar att arbete med programmering kan ha en positiv påverkan på elevernas problemlösningsförmåga, både generellt och i vissa specifika aspekter. Forskning visar att matematiska faktorer, kognitiva faktorer och affektiva faktorer kan påverkas positivt genom arbete med programmering. Det framkommer även att vissa elevgrupper kan dra större nytta än andra när det gäller utveckling av problemlösningsförmåga genom programmering. Det finns dock inget tydligt och högt värderat bevis på en signifikant förbättring av elevers problemlösningsförmåga efter arbete med programmering. Det finns inte heller något bevis för att eleverna skulle ha blivit sämre än kontrollgrupperna när de har arbetat med programmering. Genom forskningen framträder det att eleverna har utvecklat programmeringsförmågor och ett datavetenskapligt tankesätt, samtidigt

som de har arbetat med problemlösning genom programmering. Under programmeringsarbete engageras elever i styrning av robotar, spelskapande, att lära ut matematik till andra elever, lösning av strukturerade uppgifter, självstyrd utforskning av verktygen och kollaborativt arbete.

6 Diskussion

I följande avsnitt diskuteras litteraturstudiens metod och resultat.

6.1 Metoddiskussion

Metoden som har använts för detta arbete är en systematisk litteraturstudie. Sökorden, databaserna och urvalskriterierna beskrivs i detalj för att möjliggöra en oberoende upprepning av studien med samma resultat. En begränsning som har gjorts är att begränsa sökningar för att få fram artiklar från år 2000 och framåt. Detta innebär att artiklar som skrevs tidigare än år 2000 och ändå skulle kunna vara av relevans för studien har utelämnats. Begränsningen bedömdes dock vara rimlig för att få fram de artiklar som beskriver forskning som är relevant för situationen i skolan idag.

I databasen *Summon* påträffades många artiklar med vissa kombinationer av sökord, även när extra avgränsningar gjordes för att få fram de mest relevanta träffarna. För att hitta relevant forskning sorterades titlarna efter relevans och de första 200 titlarna lästes därefter. Det kan vara så att någon artikel som låg längre ner på sorteringslistan var relevant och i så fall blev den inte inkluderad i studien. Genom att söka genom flera olika databaser är det förhoppningen att detta inte har skett och att alla relevanta forskningsartiklar har fångats upp.

De sökorden som valdes togs fram för att motsvara syftet och frågeställningarna på bästa möjliga sätt. Matematiska problem och problemlösningsförmåga är svåra att definiera och tolkas på många olika sätt. Olika länder och olika forskare har olika sätt att uppfatta problemlösning. I litteraturstudien har tolkningar av betydelse av problemlösning i olika artiklar gjorts för att välja ut artiklar som handlar om matematisk problemlösning och inte de som handlar om mer generell problemlösning. Många av artiklarna definierar inte vad de menar med begreppet *problemlösning* och därför kan min tolkning skiljer sig från författarens tolkning av *problemlösning*. Det kan ha lett till att mina slutsatser inte är giltiga utifrån ett matematiskt didaktiskt perspektiv, om artikeln inte egentligen handlade om matematisk problemlösning. Genom att noggrant läsa artiklarna och väljer bort några som initialt hade valts för inkludering är det förhoppningen att detta ha undvikits.

Vid sökprocessen beslutades det att inkludera artiklar som handlade om grundskoleutbildning men inte de som handlade om gymnasie- eller högskoleutbildning, eftersom programmering och matematik blir mer avancerade på de nivåerna. Det kan vara fallet att vissa resultat ändå hade varit relevanta för frågan om kopplingar mellan programmering och problemlösning i grundskolan. Det kan även vara fallet att resultaten som är inkluderade i litteraturstudien och som handlar om yngre barn inte är relevanta för elever på högstadiet. Beslutet togs för att få fram ett brett men ändå relevant utbud av artiklar.

Den största studien som beskrivs i artiklarna är utförd i Sverige. Detta är positivt för relevansen för litteraturstudien utifrån den svenska skolans perspektiv. Andra studier är utförda i USA, Kanada, Australien, Nya Zeeland och Turkiet. Det kan diskuteras hur relevanta deras resultat är utifrån svenskt perspektiv. Det kan rimligen antas att det finns både likheter och skillnader mellan det svenska systemet och skolsystemen i länderna där de andra studierna utfördes.

Språket som alla artiklarna är skrivna på är engelska. Forskning söktes både på svenska och engelska men inga relevanta resultat hittades på svenska. Det kan även finnas forskning publicerad på andra språk som är relevant men inte tillgänglig för denna studie på grund av språket.

Innehållsanalysen har gjorts efter en modell som skulle kunna upprepas. Koderna, kategorierna och metoden som användes vid analysen har beskrivits men det finns en möjlighet att en annan forskare skulle ha tolkat resultaten som presenteras i artiklarna på ett annat sätt och därefter kommit fram till andra slutsatser.

De allra flesta studierna är baserade på kvalitativa metoder, vilka ger bra möjligheter till att upptäcka nyanser i hur programmeringsarbetet har påverkat eleverna och en god validitet i studierna (Larsen 2009, s. 27). Det är dock inte möjligt att generalisera utifrån resultat framtagna med kvalitativa metoder (Larsen 2009, s. 27). En fördel är att den största kvantitativa undersökningen som ingår i litteraturstudien är gjort i Sverige och därför kan anses ha hög relevans för den svenska kontexten.

Vissa av artiklarna är av tveksam kvalitet som vetenskapliga undersökningar, trots att de är *peer-reviewed* och har granskats för att hålla en viss vetenskaplig kvalitetsnivå. Några av artiklarna har bristande metodbeskrivningar (Aydin 2005, Grubbs 2013), några har begränsade diskussioner som inte presenterar några nackdelar av programmeringsarbete (Feurzeig et al. 2011, Calder 2010, Parker 2012, Grubbs 2013). Artiklarna av Ardito et al. (2014), Hussain et al. (2006) och Lindh och Holgersson (2007) innehåller metodbeskrivningar och kvantitativa undersökningar men ingen beskrivning av vad kontrollgrupperna gjorde under tiden där interventionsgrupperna arbetade med programmering. Feurzeig et al. (2011) är publicerat först 1967 men återtrycks 2011, och inkluderas därför i studien. Det kan ifrågasättas hur relevant den är idag, även med ett nytt förord. Alla artiklar har inkluderats i litteraturstudien eftersom de är granskade av oberoende personer och ska därför hålla en viss nivå av vetenskaplig kvalitet, men trovärdigheten av resultaten av dessa studier minskas något på grund av dessa brister.

6.2 Resultatdiskussion

Ett av mina syften med att undersöka kopplingar mellan programmering och matematisk problemlösning var att få en bättre förståelse för och förslag på hur implementeringen av Regeringskansliets (2017) införande av programmering i matematikämnet skulle kunna hanteras. Resultaten av studien visar att ämnet är komplext och att det är svårt att dra tydliga slutsatser kring hur programmeringsundervisning påverkar elevernas problemlösningsförmåga. Många av artiklarna är baserade på kvalitativa metoder som gör att det är svårt att dra generaliserade slutsatser utifrån deras resultat. Ardito et al. (2014) varnar för den föreställningen

av att tro att integrering av digital teknik och matematik kan vara en ”*silver bullet*” (s. 74) som löser alla problem som finns i utbildningssystemet.

6.2.1 Påverkan på problemlösningsförmåga generellt

Problemlösningsförmåga är komplex och täcker många matematiska områden. Garofalo och Lester (1987, s. 166) menar att det tar lång tid att utveckla problemlösningsförmågan så det ska inte förvåna om en begränsad tids undervisning i programmering inte påverkar elevernas problemlösningsförmåga. De flesta artiklarna beskriver studierna som pågick under en kortare tid, från några dagar upp till två veckor. Studien som pågick under längre tid visade inte lika positiva resultat (Hussain et al. 2006; Lindh och Holgersson 2007). Det kan vara så att eleverna missade vanlig undervisning i matematik som kanske var lika effektiv som programmeringsundervisning för utvecklingen av deras problemlösningsförmåga.

Enligt analysen gjord av Hussain et al. (2006, s. 190) finns en liten negativ påverkan på eleverna i årskurs sex problemlösningsförmåga efter deltagande i programmeringsundervisning i jämförelse med kontrollgrupperna. Lindh och Holgerssons (2007, s. 1105) analys av samma studie hittar ingen statistisk signifikant skillnad. Det är svårt att genomsåda beräkningarna eftersom originaldata inte finns bifogade i artiklarna.

Ett argument emot undervisning i programmering i skolan är att det tar för mycket tid och att felsökning kan vara svårt (Rolandsson och Skogh 2014, s. 11). Resultaten från studierna som presenteras i denna litteraturstudie motvisar detta påstående. Li et al. (2014, s. 31) drar slutsatsen att spelbyggande är minst lika effektivt som traditionell självstyrd undervisning utifrån ett resultatperspektiv. Martínez Ortiz (2015) påpekar att eleverna lärde sig både det matematiska innehållet om proportionalitet lika effektivt som de som deltog i traditionell läroboksundervisning, samtidigt som de utvecklade kunskaper i programmering.

En slutsats som kan dras från resultaten av litteraturstudien är att programmering kan användas med fördel inom matematikundervisning för att arbeta med problemlösning och samtidigt lära elever programmering. Det ska dock inte förväntas någon omedelbar ökning i elevernas problemlösningsförmåga i jämförelse med traditionell undervisning.

6.2.2 Påverkan på matematiska faktorer

Många av studierna fokuserar på ett matematiskt område för problemlösning, som proportionalitet (Martínez Ortiz 2015) eller avstånd och tid (Grubbs 2013; Ardito et al. 2014). Inom dessa studier verkar eleverna ha utvecklat en djupare förståelse av matematiska begrepp, något som är en faktor i problemlösningsförmågan enligt Möllehed (2001, s. 160). Även Calder (2010) och Li et al. (2014) beskriver en ökad förståelse av matematiska begrepp efter arbete med skärmbaserad programmering. Möllehed fann att 25 % av felen vid elevernas problemlösning var på grund av brister i matematiska faktorer, inklusive förståelse av matematiska begrepp. Om arbete med programmering ökar elevers förståelse av matematiska begrepp är det rimligt att detta kommer att påverka deras problemlösningsförmåga.

6.2.3 Påverkan på kognitiva faktorer

Många av artikelförfattarna beskriver observationer av påverkan på kognitiva och metakognitiva faktorer, som de som beskrivs av Garofalo och Lester (1985, s. 171), under arbete med programmering. En positiv påverkan på elevernas planeringsförmåga (Calder 2010; Castledine och Chalmer 2011), val av strategi (Li et al. 2014; Parker 2012) och reflektionsförmåga (Calder 2010, Castledine och Chalmer 2011) observerades. Den del av det centrala innehållet för årskurs 7-9 som behandlar ”strategier för problemlösning” och ”värdering av valda strategier och metoder” (Skolverket 2016a, s. 56) är ett område i kursplanen som lämpar sig väl till arbete med programmering. Den omedelbara återkopplingen som sker under programmeringsarbete kan hjälpa elever att uppfatta och reflektera över om en strategi var framgångsrik eller inte.

Möllehed (2001, s. 160) listar även rumsuppfattning som en kognitiv faktor som påverkar elevernas problemlösningsförmåga. Ratcliff och Anderson (2011), Calder (2010) och Ardito et al. (2014) menar att eleverna utvecklade sin rumsuppfattning under arbetet med programmering. Programmering av geometriska former, styrning av virtuella objekter i ett virtuellt rum och styrning av robotar i ett verkligt rum kan vara sätt att skapa möjligheter för elever att öva och utveckla sin rumsuppfattning och kan bidra till att eleverna tvingas tänka medvetet om sina uppfattningar om rum, avstånd, hastighet och vinklar.

Trots dessa observationer om påverkan på kognitiva faktorer fanns det inga kvantitativa observationer om ökad problemlösningsförmåga. Mayer och Wittrock (1996, s. 55) menar att elever kan ha svårigheter med att känna igen liknande strukturer i olika problem. Det kan vara så att eleverna inte ser likheter mellan de processer de använde när de skapade program, till exempel med robotar, och lösningar på andra matematiska problem. Även fast många av studierna (Aydin 2005; Ardito et al. 2014; Calder 2010; Castledine och Chalmer 2011; Feurzeig et al. 2011; Li et al. 2014; Martínez Ortiz 2015; Parker 2012) är positiva till arbete med programmering kopplad till problemlösning är det inte säkert att påverkan på elevernas kognitiva förmåga som observerats i studierna kommer att överföras och påverka deras matematiska problemlösningsförmåga i andra kontexter.

En slutsats av litteraturstudien är att programmering påverkar liknande kognitiva och metakognitiva processer som de som används i matematisk problemlösning och kan användas för att möjliggöra övning inom kognitiva områden som används inom matematisk problemlösning. Påverkan på elevernas förmågor inom dessa kognitiva områden överförs dock inte nödvändigtvis från programmering till matematisk problemlösning.

6.2.4 Påverkan på övriga faktorer

Som beskrivs i bakgrunden, nämner McLeod (1985, s. 274-5) både stress inför problemlösningssituationer och olika tro på orsaken till framgång i problemlösning som faktorer som påverkar problemlösningsförmåga. Ett ökat självförtroende inför problemlösning skulle kunna påverka elevernas problemlösningsförmåga positivt. Hussain et al. (2006) och Castledine och Chalmer (2011) hävdar

att eleverna hade bra självförtroende på sina problemlösningsförmågor efter arbete med programmering av robotar.

Enligt Schoenfeld (1992, s. 359) finns det många elever i USA som ger upp om inte de lyckas lösa ett problem inom fem minuter. En ökad uthållighet vid problemlösning, något som framkommer i flera forskningsstudier om arbete med skärmbaserad programmering (Ratcliff och Anderson 2011; Li et al. 2014; Parker 2012), skulle kunna öka elevernas framgång vid problemlösning. Om eleverna är villiga att investera kraft och tid på lösning av problem i programmeringsarbete skulle detta kunna användas för att öka deras förståelse för att problemlösning får ta tid och att olika lösningsstrategier kan testas utan att det ska ses som ett misslyckande.

Möllehed (2001, s. 160) nämner motivation som en faktor som påverkar elevernas förmåga att lösa matematiska problem. En hög motivationsnivå vid programmeringsarbete beskrivs av flera forskare (Ratcliff och Anderson 2011; Li et al. 2014; Calder 2010; Parker 2012; Ardito et al. 2014). Det skulle kunna utnyttjas av lärare för att mer effektivt kunna arbeta med problemlösning, genom att använda programmering som medel för att öka motivation för arbetet.

En slutsats av litteraturstudien är att programmering med fördel kan användas inom matematikundervisning för att öka elevernas motivation och tydliggöra att problemlösning är en process som får ta tid.

6.2.5 Elevaktiviteter

Enligt Ratcliff och Anderson (2011, s. 248) är uppgifter med strukturerade problem effektiva för lärande i programmering. Detta påstående stöds av andra forskare (Clements och Sarama, 1995, s. 394; Delclos 1984, s. 13; Pea 1983, s. 7). Hussain et al. (2006, s. 188) och Lindh och Holgersson (2007, s. 1103) menar att elever måste ges konkreta, relevanta uppgifter för att främja lärande i arbetet med *LEGO Mindstorms*. När lärare planerar arbete med programmering bör detta tas i beaktning. Feurzeig et al. (2011, s. 499) och Aydin (2005, s. 30) menar att elevernas undersökningar och konstruerande av eget lärande är viktiga. Det bör ske i strukturerade uppgifter för att undvika fällor som de som beskrivas av Ratcliff och Anderson (2011, s. 247). Många av forskarna lyfter lärarens roll i programmeringsarbetet, och hur viktigt det är med bra planering och ett tydligt syfte. Hussain et al. (2006, s. 189) och Lindh och Holgersson (2007, s. 1104) beskriver lärarens roll som avgörande och lyfter också fördelar med att vara minst två lärare som kunde arbeta tillsammans i klassrummet under programmeringsarbete. För att programmering ska ha positiv påverkan på eleverna måste det finnas ett bra klassrumsklimat (Parker 2012, s. 413; Calder 2010, s. 11) och lärare som är intresserade av programmeringsarbete (Ardito et al. 2014, s. 77). Rolandsson och Skogh (2014, s. 21) framhäver problemet med bristande programmeringskunskaper hos verksamma lärare.

Ardito et al. (2014, s. 85) rapporterar en positiv utveckling i elevernas förmåga att effektivt arbeta tillsammans efter kollaborativt arbete med robotar. Kollaborativt arbete kan vara både ett medvetet val från lärare eller forskare och ett nödvändigt villkor eftersom det kan ha funnits en begränsning i antalet digitala verktyg eller

robotar tillgängliga. Att arbeta kollaborativt innebär även att eleverna diskuterar sitt arbete och resonerar kring sina val av strategier, något som ger de möjlighet att utveckla sin förmåga att föra matematiskt resonemang. Det är en förmåga eleverna ska få möjlighet att utveckla genom undervisning i matematik (Skolverket 2016, s. 56).

En slutsats av litteraturstudien är att det finns flera olika aktiviteter som kan användas för att genomföra programmeringsundervisning som är kopplad till matematisk problemlösning. Programmering kan genomföras i visuella miljöer eller andra miljöer (som med robotar) som beskrivs i den nya formulering som införs i kursplanen för matematik (Regeringskansliet 2017, s.2). Mer forskning behövs för att kunna yttra sig om vilka aktiviteter som är mest effektiva.

6.2.6 Studiestorlek

I många av studierna har små elevgrupper undersökts. Det kan postuleras att det är de lärare som är intresserade av och entusiastiska för programmering som har valt att införa programmeringsundervisning i sina lektioner. Dessa studier har något mindre bevisvärde på grund av det mindre antalet elever som undersöktes. Den studie som undersökt ett större antal elever (Hussain et al. 2006, Lindh och Holgersson 2007) och därmed hade ett högre bevisvärde har inte hittat lika positiva resultat. Det är oklart hur klasserna och lärarna valdes ut i denna studie men det kan tänkas att inte alla lärare kände sig lika väl förberedda eller entusiastiska över programmeringsarbete.

6.2.7 Språket

Ett särskilt övervägande som måste göras i Sverige är om programmeringsverktyget finns tillgängligt på svenska. Det framgår inte från artiklarna av Hussain et al. (2006) och Lindh och Holgersson (2007) om eleverna som arbetade med *LEGO Mindstorms* i Sverige programmerade med ett verktyg som baserades på svenska eller engelska. Om programmeringsspråket var engelska kan det tänkas att detta medförde en extra svårighet, speciellt för vissa elever.

6.3 Avslutande reflektion

Ett syfte med litteraturstudien var att undersöka forskningsläget gällande hur undervisning i programmering på grundskolan påverkar elevernas förmåga att lösa matematiska problem. Studien har visat att det finns relevant forskning som behandlar påverkan av programmeringsundervisning på elevernas problemlösningsförmåga, men att forskningen inte är entydig i sina svar på frågan om hur det påverkar problemlösningsförmågan. Det finns dock många tecken från forskning på att införandet av programmeringsundervisning i läroplanen för matematik kan fungera väl som en del i undervisning för utveckling av elevernas problemlösningsförmåga. En annan del i syftet var att undersöka vilka elevaktiviteter som används inom programmeringsundervisning som är kopplad till matematisk problemlösning. Flera elevaktiviteter som används inom programmeringsundervisning som är kopplad till matematisk problemlösning har beskrivits.

Problemlösning och utvecklandet av problemlösningsförmågan är komplexa fenomen. Programmering kan innefatta många olika moment (planering,

modellering, programmering av kod, verifikation, utveckling) och verktyg (programmering av kalkylblad, grafräknare, spelbyggande och styrning av robotar). Med tanke på Wings (2006, s. 35) påstående att ett datavetenskapligt tänkande bör jämföras med läsning, skrivande och matematik, som en grundläggande analytisk förmåga, kan det vara värt att fundera över vad programmering, som det benämns av Regeringskansliet (2017), innefattar, och syftet med införandet av programmering i grundskolans läroplan.

7 Förslag till framtida forskning

I framtiden skulle det vara intressant att undersöka vilka fördelar med respekt till matematikinlärning de olika programmeringsmiljöerna kan ge. Finns det fördelar med att programmera robotar som är mer konkreta i jämförelse med skärmbaserade program? Finns det fördelar med att eleverna lär sig både att programmera robotar och att bygga spel? Även en mer detaljerad jämförelse av de olika aktiviteter som används inom programmeringsundervisning och deras påverkan på problemlösnings- och matematikförmåga skulle vara intressant för att kunna stödja lärare i deras planering av programmeringsundervisning.

Det antyds från studieresultaten att utveckling av spel för att lära andra elever matematik är ett effektivt sätt för eleverna att utveckla en djupare förståelse av matematiska begrepp. Det skulle vara intressant att undersöka om byggandet av spel för att lära andra matematik är mer effektivt i utvecklingen av elevernas förståelse av matematiska begrepp än andra sätt att lära ut matematik till andra elever.

Ett annat intressant forskningsämne skulle vara verksamma matematiklärares kunskapsnivåer inom programmering. Även en undersökning av lärares tankar kring hur programmering bäst kan införas i matematikundervisning och vad de ser som potentiella möjligheter och hinder i arbetet med programmering i matematikundervisning skulle vara intressant.

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.
- Calder, N. (2010). Using Scratch: An Integrated Problem-solving Approach to Mathematical Thinking. *Australian Primary Mathematics Classroom*, (15)4, 9-14. Hämtad från <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.
- Clements, D. H., & Sarama, J. (1995). Design of a Logo Environment for Elementary Geometry. *The Journal of Mathematical Behavior*, (14)4, 381-398. [https://doi.org/10.1016/0732-3123\(95\)90037-3](https://doi.org/10.1016/0732-3123(95)90037-3)
- Delclos, V. R. (1984). *Teaching Thinking through LOGO: The Importance of Method. Technical Report Series, Report No. 84.1.2*. Nashville: George Peabody College for Teachers, Learning Technology Center. <http://files.eric.ed.gov/fulltext/ED262756.pdf>
- Eriksson Barajas, K., Forsberg, C. & Wengström, Y. (2013). *Systematiska litteraturstudier i utbildningsvetenskap*. Stockholm: Natur och Kultur.
- 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.
- Healy, L. & Kynigos, C. (2010). Charting the microworld territory over time: design and construction in mathematics education. *ZDM Mathematics Education* 42(1), 63–76. <https://doi.org/10.1007/s11858-009-0193-5>
- Helenius, O. (2006). Kompetenser och matematik. *Nämnnaren* (3), 11-15. Hämtad 2017-03-17 från http://ncm.gu.se/pdf/namnaren/1115_06_3.pdf
- 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>
- Hussain, S., Lindh, J., Shukur, G. (2006). The effect of LEGO Training on Pupils' School Performance in Mathematics, Problem Solving Ability and Attitude: Swedish Data. *Educational Technology & Society*. 9(3), 182-194.
- Högskolan Dalarna (2017). *Ämnesguide för pedagogik*. Hämtad 2017-04-27 från <http://libguides.du.se/pedagogik>
- Kilpatrick, J., Swafford, J. & Findell, B. (Red.) (2001). *Adding + it up. Helping children learn mathematics*. Washington D.C.: National Academy Press. Hämtad 2017-04-07 från <https://www.nap.edu/read/9822/chapter/1>

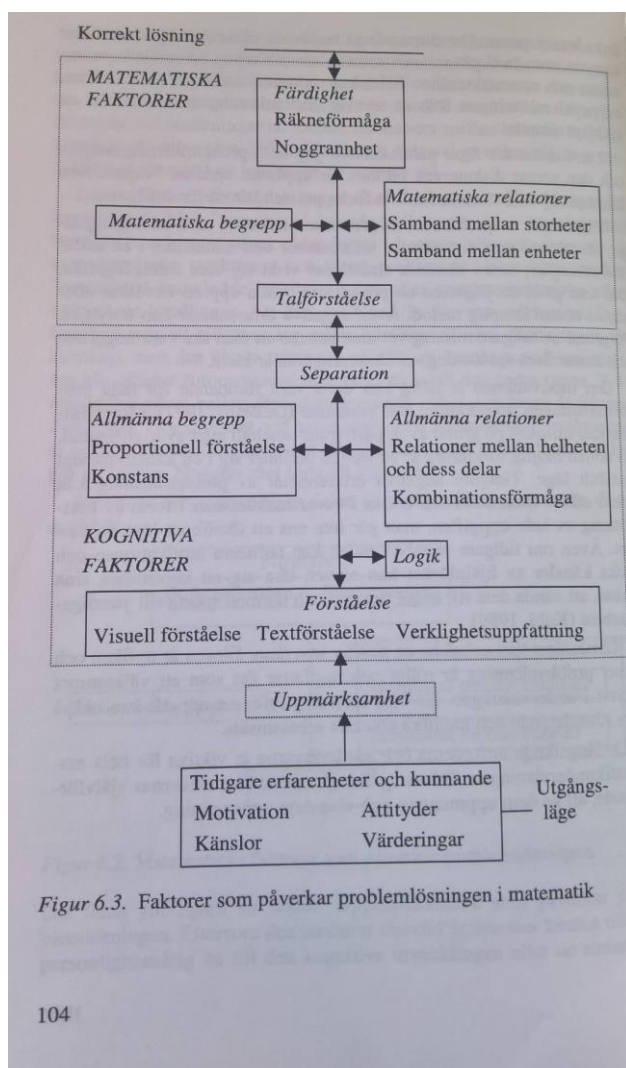
- Kilpatrick, J. (1985). A Retrospective Account of the Past 25 Years of Research on Teaching Mathematical Problem Solving. I E. A. Silver (Red.), *Teaching and Learning Mathematical Problem Solving: Multiple Research Perspectives* (s. 1-15). New Jersey: Lawrence Erlbaum Associates.
- Larsen, A. K. (2009). *Metod helt enkelt*. Malmö: Gleerups.
- Lesh, R., & Zawojewski, J. (2007). Problem Solving and Modeling. I F. K. Lester, Jr. (Red.) *Second Handbook of Research on Mathematics Teaching and Learning* (s. 763-804). Charlotte, NC: Information Age Publishing.
- 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.
- Lindh, J., Holgersson, T. (2007). Does lego training stimulate pupils' ability to solve logical problems? *Computers and Education*, 49(4), 1097–1111. <https://doi.org/10.1016/j.compedu.2005.12.008>
- 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.
- McLeod, D. B. (1985). Affective Issues in Research on Teaching Mathematical Problem Solving. I E. A. Silver (Red.), *Teaching and Learning Mathematical Problem Solving: Multiple Research Perspectives* (s. 267-279). New Jersey: Lawrence Erlbaum Associates.
- Möllehed, E. (2001). *Problemlösning i matematik. En studie av påverkansfaktorer i årskurserna 4-9*. Malmö: Institutionen för pedagogik, Lärarhögskolan i Malmö.
- Narode, R. (1987). *Metacognition i math and science education*. Educational Resources Center (ERIC). Hämtad 2017-03-27 från <http://files.eric.ed.gov/fulltext/ED291558.pdf>
- Nesher, P., Hershkovitz, S. & Novotna, J. (2003). Situation Model, Text Base and What Else? Factors Affecting Problem Solving. *Educational Studies in Mathematics*, 52(2), 151-176. Hämtad 2017-03-16 från www.jstor.org/stable/3483174
- Niss, M. & Højgaard Jensen, T. (Red.) (2002). *Kompetencer og matematiklæring: Ideer og inspiration til udvikling af matematikundervisning i Danmark*. København: Undervisningsministeriets forlag. Hämtad 2017-04-07 från <http://static.uvm.dk/Publikationer/2002/kom/hel.pdf>
- OECD (2016). *Pisa 2015 Assessment and Analytical Framework: Science, Reading, Mathematic and Financial Literacy*. Paris: PISA, OECD Publishing. Hämtad 2017-05-22 från <http://www.oecd.org/publications/pisa-2015-assessment-and-analytical-framework-9789264255425-en.htm>
- Parker, T. (2012). ALICE in the Real World. *Mathematics Teaching in the Middle School*, (17)7, 410-416.
- Pea, R. (1983). *Logo Programming and Problem Solving. Technical Report No. 12*. New York: Bank Street Coll. of Education, Center for Children and

- Technology. Hämtad 2017-05-22 från
<http://files.eric.ed.gov/fulltext/ED319371.pdf>
- Polya, G. (1957). *How to solve it*. Princeton: Princeton University Press. Hämtad 2017-03-17 från
https://notendur.hi.is/hei2/teaching/Polya_HowToSolveIt.pdf
- 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>
- Regeringskansliet (2017). *Stärkt digital kompetens i skolans styrdokument*. Hämtad 2017-04-06 från
<http://www.regeringen.se/493c41/contentassets/acd9a3987a8e4619bd6ed95c26ada236/informationsmaterial-starkt-digital-kompetens-i-skolans-styrdokument.pdf>
- Rolandsson, L. (2015). *Programmed or not. A study about programming teachers' beliefs and intentions in relation to curriculum*. Stockholm: KTH. Hämtad 2017-05-22 från <http://kth.diva-portal.org/smash/get/diva2:791197/FULLTEXT02.pdf>
- Rolandsson, L. & Skogh, I.-B. (2014). Programming in school: Look back to move forward. *ACM Transactions on Computing Education*, 14(2), Article 12. <http://dx.doi.org/10.1145/2602487>
- Schoenfeld, A. H. (1992). Learning to think mathematically: Problem solving, metacognition, and sense-making in mathematics. I D. Grouws (Red.), *Handbook for Research on Mathematics Teaching and Learning* (s. 334-370). New York: MacMillan.
- Scratch (2013). *Spiral Maker*. Hämtad 2017-05-18 från
<https://scratch.mit.edu/projects/11641125/#editor>
- SFS 2010:800. *Skollag*. Stockholm: Utbildningsdepartementet. Hämtad 2017-05-22 från http://www.riksdagen.se/sv/dokument-lagar/dokument/svensk-forfattningssamling/skollag-2010800_sfs-2010-800
- Skolverket (2016a). *Läroplan för grundskolan, förskoleklassen och fritidshemmet 2011. Reviderad 2016*. Stockholm: Skolverket. Hämtad 2017-05-22 från
https://www.skolverket.se/om-skolverket/publikationer/visa-enskild-publikation?_xurl_=http%3A%2F%2Fwww5.skolverket.se%2Fwtpub%2Fws%2Fskolbok%2Fwpubext%2Ftrycksak%2FBlob%2Fpdf2575.pdf%3Fk%3D2575
- Skolverket (2016b). *Rapport 450: PISA 2015 - 15-åringars kunskaper i naturvetenskap, läsförståelse och matematik*. Stockholm: Skolverket. Hämtad 2017-05-22 från https://www.skolverket.se/om-skolverket/publikationer/visa-enskild-publikation?_xurl_=http%3A%2F%2Fwww5.skolverket.se%2Fwtpub%2Fws%2Fskolbok%2Fwpubext%2Ftrycksak%2FBlob%2Fpdf3725.pdf%3Fk%3D3725
- Skolverket (2016c). *Digital kompetens och programmering ska stärkas i skolan*. Hämtad 2017-04-06 från
<https://www.skolverket.se/laroplaner-amnen-och-kurser/nyhetsarkiv/nyheter-2016/nyheter-2016-1.247899/digital-kompetens-och-programmering-ska-starkas-i-skolan-1.247906>
- Skolverket (2011a). Kommentarmaterial till kursplanen i matematik. Stockholm: Skolverket. Hämtad 2017-05-22 från https://www.skolverket.se/om-skolverket/publikationer/visa-enskild-publikation?_xurl_=http%3A%2F%2Fwww5.skolverket.se%2Fwtpub%2Fws%2Fskolbok%2Fwpubext%2Ftrycksak%2FBlob%2Fpdf3725.pdf%3Fk%3D3725

[publikation? xurl =http%3A%2F%2Fwww5.skolverket.se%2Fwtpub%2Fws%2Fskolbok%2Fwpubext%2Ftrycksak%2Fblob%2Fpdf2608.pdf%3Fk%3D2608](http://www.skolverket.se/publikation?xurl=http%3A%2F%2Fwww5.skolverket.se%2Fwtpub%2Fws%2Fskolbok%2Fwpubext%2Ftrycksak%2Fblob%2Fpdf2608.pdf%3Fk%3D2608)

- Skolverket (2011b). *Kursplan för programmering*. Stockholm: Skolverket. Hämtad 2017-04-18 från <https://www.skolverket.se/laroplaner-amnen-och-kurser/gymnasieutbildning/gymnasieskola/sok-amnen-kurser-och-program/subject.htm?lang=sv&subjectCode=pr&tos>
- Sollervall, H. & Helenius, O. (2016). *Matematiska undersökningar med kalkylprogram*. Stockholm: Skolverket. Hämtad 2017-04-27 från https://larportalen.skolverket.se/webcenter/larportal/api-v2/document/path/larportalen/material/inriktningar/1-matematik/Grundskola/436_matematikundervisningmeddigitalaverktyg_%C3%A5k7-9/7_matematiskaundersokningar_kalkylprogram/material/flikmeny/tabA/Artiklar/IKT7-9_07A_01_matematiskaundersokningarmedkalkylprogram.docx
- Wing, J. M. (2006). Computational Thinking. *COMMUNICATIONS OF THE ACM*, 49(3), 33-35. Hämtad 2017-04-18 från <http://marvin.cs.uidaho.edu/Teaching/K12-CS/computationalThinkingWing.pdf>

Appendix A: Mölleheds modell av påverkansfaktorer för matematisk problemlösning



Möllehed 2001, s. 104

Appendix B: Programmeringsspråk i skolor

Programmeringsmiljöer som används i skolor	Land
Scratch	USA, NZ, Korea, Storbritannien, Tyskland
Kodu	USA, Korea
LOGO	Korea, Storbritannien
AgentCube	USA
AgentSheet	USA
Alice	USA
Blockly	USA
Game Maker	USA
Micro Worlds	Tyskland
Robot Karol	Tyskland
Squeak Etoys	Korea
BlueJ	Tyskland
Greenfoot	USA
Java's Cool	Frankrike
Jeroo	USA
LEGO Mindstorms	Ospecifierad
Raspberry Pi	Storbritannien

Hubweiser et al. 2015, s.76