Linnéuniversitetet
Kalmar Växjö

Computational Thinking
Reflektion kring programmering

Författare: Rebecca Emmott
Termin: HT 19
Ämne: Digitala verktyg i förskolan
Kurskod: 1ME112
UG1 B2

Innehållsförteckning

Bilagor

Computational Thinking

Computational thinking kan översättas till datalogiskt tänkande. Wing (2006) skriver att datalogiskt tänkande är en grundläggande färdighet som varje medborgare kommer att behöva bemästra för att kunna fungera i det globala samhället då skriv-, aritmetisk och läsförmåga är viktigt idag. Lee et al. (2011) skriver att ”datalogiskt tänkande involverar att definiera, förstå och lösa problem, resonera på flera nivåer av abstraktion, förstå och tillämpa automatisering och analysera lämpligheten för de gjorda abstraktionerna” (s. 32).

Datalogiskt tänkande handlar om problemlösning och bygger på förståelse om vad problemet handlar om och utvecklandet av möjliga lösningar. International Society for Technology in Education (ISTE) och Computer Science Teachers Association (CSTA) (2011) skriver att det kan innebär att man ”formulera problem på ett sätt som gör det möjligt för oss att använda en dator och andra verktyg för att lösa dem” (s. 1).

Mannila et al. (2014) poängtera hur viktigt det är att datalogiskt tänkande introduceras till de yngre barnen och vikten av att deras lärare känna till och implementerar det i barnens utbildning. Grover och Pea (2013) skriver en lista som innehåller nio olika nyckelbegrepp som de anser att datalogiskt tänkande utvecklar. Jag har valt att fokusera på fem av dessa nyckelbegrepp och hur en medveten pedagog kan jobba med dem i förskolan tillsammans med barn av alla åldrar utan att använda sig av en dator.

1 Debugging and systematic error detection

Felsökning och systematisk feldetektering är en process där man aktivt söker efter och identifierar problem i en algoritm eller kod så att man kan förstå sig på orsaken och sedan kan korrigera dem (Barefoot, 2019a).

Pea och Kurland (1984) skriver att när barnen lär sig att programmera bygger de komplexa kognitiva kompetenser relaterade till problemlösning och logisk tänkande som de har användning för i alla andra ämnen i förskolan och skolan. Harvard Graduate School of Education (2017) har skrivit några strategier som kan vara hjälpsamma för barnen att tänka på när de har svårt att hitta eller lösa ett problem. De beskriva att barnen ska ”tänka som en dator, tar ut några kommandon för att se vad de gör, gissa och kolla och kolla upp dem” (s. 1).

En pedagog kan med hjälp av Bee-Bot, Blue-Bot och Scratch ge barnen kod som innehåller fel, och till exempel, fråga dem varför dansar inte scratch som jag vill, eller varför når inte Blue-Bot till målet? Och sedan be dem hjälpa till att felsöka och rätta till koden genom att lägga till kommandon som saknas och ändra eller ta bort felaktiga kommandon.

En situation som jag stöter på vardagligen som innefattar felsökning är när de yngsta barnen i förskolan ska klä av eller på sig själva. Ibland försöker barnen att ta

av sig sin fleece, och börjar dra ned dragkedjan innan de tagit av sig sin jacka eller hängselbyxor vilket inte går då det är omöjligt för dem att dra ned dragkedjan helt och ta av fleecen innan de tagit av både jackan och hängselbyxorna. ISTE och CSTA (2011) skriver att datalogiskt tänkande tränar kommunikations- och samarbetsförmåga. När man jobbar med felsökning kan det vara bra att utgå ifrån Vygotskijs sociokulturellt perspektiv på lärande eftersom det erbjuds möjligheter till att lära sig i olika sociala sammanhang. Med stöd från andra kan barnen lösa problem och lära sig att ta av sig jackan först, sedan sina hängselbyxor och först efter det ta av sig fleecetröjan.

2 Structured problem decomposition (modularizing)

British Broadcasting Corporation (BBC, 2019) och Harvard Graduate School of Education (2017) beskriver att dekomposition innebär att man bryter ned ett stort problem eller system till mindre, mer lätthanterliga delar, samt att man utforskar sambandet mellan helheten och dess delar.

Det kan underlätta särskilt för de yngre barnen i förskolan samt barn i behov av särskilt stöd om de har tillgång till ett dagligt schema som visuellt visar hur deras dag ska ser ut. Med hjälp av bildstöd kan pedagogerna bryta ned barnens vardag i mindre med hanterliga delar så att barnen kan förstår hur deras dag eller vecka ser ut. Det kan också göra så att barnen får mer inflytande och delaktighet i sin vardag genom att barnen kan väljer mellan olika aktiviteter. Det kan underlätta barnens vardag och leda till en bra övergång mellan två aktiviteter men också tydliggöra struktur, planering och organisation för båda barnen och pedagogerna.

Det finns många olika sätt att arbeta med modularisering inom skapande. Det är viktigt att man först identifierar alla de olika komponenterna i en bild för att sedan kunna rita av den, till exempel så kan barnen bryta ned en enkel teckning av en katt genom att rita en stor cirkel för kattens huvud, en triangel för dess öronen och två små cirklar för ögonen. Barnen kan också använda sig av ett rutnät för att bryta ned en bild i mindre komponenter för att kopiera eller fortsätta rita på en bild med hjälp av symmetri.

3 Symbol systems and representations

Goldin och Shteingold (citerad i Kjällander och Riddersporre, 2019) skriver att ”en symbol är en representation- till exempel ett tecken- som står för (representerar) något annat än sig själv” (s. 128).

Nouri et al. (2019) föreslår att barnen kan skapa sina egna symboler för att representera olika kommandon i deras kod när de göra analogprogrammering. Sedan kan barnen använda sig av mer avancerade symboler som med Bee-Bot, Scratch och Microbit i framtiden.

I förskolans läroplan (Skolverket, 2018) står det att barnen ska utveckla ”intresse för skriftspråk samt förståelse för symboler och hur de används för att förmedla

budskap" (s. 14). Kjällander och Riddersporre (2019) skriver att barnen vardagligen möter många symboler i förskolan; bokstäver i sitt namn, siffror som visar hur gamla barnen är, bilder som representerar olika saker eller en pil som barnen möter när de programmerar analogt eller digitalt med hjälp av en robot.

Barnen skulle till exempel kunna rita sina egna symboler för att representera hur de dansar till musik eller gör olika rörelser för att härma olika djur. Symbolerna skulle representera hur barnen utforskar sina kroppar och spatiala tänkande med olika rörelser och vart i rummet de ska ske. Barnen måste fundera över hur de ska rita sina olika symboler så att även andra kan läsa och förstå dem och därmed kunna röra sig till symbolerna.

4 Algorithmic notions of flow of control

En algoritm är en väldefinierad serie steg av instruktioner eller sekvenser som är en lösning på ett problem för att uppnå ett önskat resultat (The Chartered Institute for IT, BCS, 2019). Den är beroende på ordningen mellan instruktionerna eller reglarna för att lösa problemet.

Pedagoger i förskolan kan jobba med algoritmer med de yngsta barn när de till exempel följer ett recept för att göra play-dough. När de bakar finns det också möjligheten för barnen att träna på abstraktion (se nedan). När barnen följer receptet kan de med hjälp av abstraktion väljer ut de viktigaste delarna av informationen som behövs för att göra sin play-dough. I receptet kanske det finns onödig och oviktig information som inte har någon betydelse för hur slutresultatet; play-dough blir.

Pedagoger använder sig ofta av flanosagor i förskolan där det är viktigt att ha alla huvudkaraktärer och rekvisiter som bilder för att återberätta sagan i rätt följd. Barnen tränar även dessa färdigheter när de sjunger och gör rörelser till sånger som till exempel, huvud, axlar, knä och tån.

5 Abstractions and pattern generalizations (including models and simulations)

Genom att söka efter likheter mellan ett problem man ha nu och med ett tidigare löst problem, kan man göra generaliseringar av mönster som hjälper oss att lösa andra problem (Barefoot, 2019b) och även stora mer komplexa problem mer effektivt (Tiny Thinkers, 2019). Google for Education (2019) skriver att generalisering av mönster är att "skapa modeller, regler, principer eller teorier om observerade mönster för att testa förutsagda resultat". Att abstrahera är möjligheten att dela upp ett problem, extrahera nyckeldelar och skapa en ram för problemlösning genom att ignorera de oviktiga delar.

Barnen kan analogt programmera varandra och bygga med lego med hjälp av olika kommandon som en person ger till den andra. Ett barn bygga ett mönster med klossar av olika form och färg och berätta hur den ser ut och programmerar den andra personen att bygga en likadan. Det kan vara så att barnen använder sig av

olika kommandon som kanske inte ha någon betydelse för hur slutprodukten, bygget ska se ut. Då är det viktigt att den andra person ignorerar sådana onödiga instruktioner.

Att göra ett halsband av pärlor kan vara ett sätt att utforska olika mönster (ABAB). Där kan barnen skapa en kontinuerlig loop av mönster tills tråden tar slut.

Bilagor

Referenser

- Barefoot. (2019a). Debugging. Hämtad 2019-10-04 från https://www.barefootcomputing.org/concepts-and-approaches/debugging
- Barefoot. (2019b). Decomposition. Hämtad 2019-09-26 från https://www.barefootcomputing.org/docs/default-source/resource-downloads/comput-1fc5ffcdbdcfc6c779083ff0100ba3f46.pdf?sfvrsn=d4a390ea_0
- BBC. (2019). Introduction to computational thinking. Hämtad 2019-09-27 från https://www.bbc.co.uk/bitesize/guides/zp92mp3/revision/1
- BCS. (2019). The benefits of computational thinking. Hämtad 2019-10-05 från https://www.bcs.org/content-hub/the-benefits-of-computational-thinking/
- Google for Education. (2019). Hämtad 2019-09-27 från https://edu.google.com/resources/programs/exploring-computational-thinking/#!ct-overview
- Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. Educational researcher, 42(1), 38-43.
- Harvard Graduate School of Education. (2016). Computational thinking with Scratch. Developing fluency with computational concepts, practices, and perspectives. Hämtad 2019-10-04 från http://Harvard Graduate School of Education.gse.harvard.edu/ct/index.html
- International Society for Technology in Education (ISTE) och Computer Science Teachers Association (CSTA). (2011). Operational Definition of Computational Thinking for K–12 Education . Hämtad 2019-09-29 från https://id.iste.org/docs/ct-documents/computational-thinking-operational-definition-flyer.pdf¨
- Kjällander, S. & Riddersporre, B. (red.) (2019). Digitalisering i förskolan: på vetenskaplig grund. (Första utgåvan). [Stockholm]: Natur & Kultur.
- Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., ... & Werner, L. (2011). Computational thinking for youth in practice. Acm Inroads, 2(1), 32-37.
- Mannila, L., Dagiene, V., Demo, B., Grgurina, N., Mirolo, C., Rolandsson, L., & Settle, A. (2014). Computational thinking in K-9 education. In Proceedings of the working group reports of the 2014 on innovation & technology in computer science education conference (pp. 1-29). ACM.

- Nouri, J., Zhang, L., Mannila, L., & Norén, E. (2019). Development of computational thinking, digital competence and 21st century skills when learning programming in K-9. Education Inquiry, 1-17.
- Pea, R.D., & Kurland, D.M. (1984). On the cognitive effects of learning computer programming.
- Harvard Graduate School of Education. (2017). Celebrating Mistakes and Embracing Process. Hämtad 2019-10-04 från https://Harvard Graduate School of Education.gse.harvard.edu/stories/see-inside-classroom-emily-roach.html
- Sverige. Skolverket (2018). Läroplan för förskolan: Lpfö 18. Stockholm: Skolverket.
- Tiny Tinkers. (2019). What's computational thinking? Hämtad 2019-09-29 från https://www.tinythinkers.org/benefits
- Wing, J. M. (2006). Computational thinking. Communications of the ACM, 49(3), 33-35. DOI: https://doi.org/10.1145/1118178.1118215