

# TDDD78 projekt: Tiny BASIC-tolk

## 1 Introduktion

Programmeringsspråk är något som har utvecklats i takt med att datorer har utvecklats. I början användes hålkort för att skriva program. Därefter började språken komma i skriftlig form och i dag finns det även grafiska programmeringsspråk som oftast går ut på att man kopplar ihop olika block i ett grafiskt fönster. Hur språken har sett ut har varierat beroende på vilka begränsningar datorn som det ska köras på har. Ett språk som utvecklades tidigt, då det fortfarande var ont om hårddiskutrymme, var Tiny BASIC. Det är en dialekt på programmeringsspråket BASIC för vilken tolken kunde implementeras med väldigt lite kod. Det gjorde att det fanns mer utrymme kvar för användarens egna kod.

Om du väljer detta projekt bör du vara intresserad av programmeringsspråk och ha en vilja att (delvis på egen hand) utforska hur en tolk (interpretator) kan skrivas. Detta är annars något som kommer att diskuteras mer under senare delar av utbildningen.

## 2 Bakgrund

<http://www.ittybittycomputers.com/IttyBitty/TinyBasic/TBuserMan.txt> är en sida som utger sig för att vara en kopia av originalbeskrivningen av Tiny BASIC. Oavsett om det stämmer eller ej är det en bra beskrivning av språket.

[https://en.wikipedia.org/wiki/Tiny\\_BASIC](https://en.wikipedia.org/wiki/Tiny_BASIC) är en Wikipedia-artikel som ger en liten mer lättsmält beskrivning av språket (och även hur man läser grammatiken) men den presenterade grammatiken som är inte helt korrekt (23 oktober 2017).

## 3 Milstolpar

De flesta tolkar är väldigt lika varandra. De har en prompt som låter användaren skriva in kod som körs. Eventuellt skrivs även någon text ut eller resultatet av beräkningen. Den mest grundläggande funktionaliteten finns beskriven i de översta noderna, fram till och med fungerande tolk, i figur 1. Resterande noder i figur 1 samt de i figur 2 är ytterligare funktionalitet i språket samt några utökningar till språket.

## 4 Milstolpe beskrivningar

Följande är en lite djupare beskrivning av milstolparna i projektet. Generellt ges en överblick här och det är upp till er att söka den behövda informationen i dokumentationen av språket.

- **Studsande prompt:** Målet med den här milstolpen är att bygga upp den behövda funktionaliteten i språket. Den går ut på att användaren ska visas en prompt

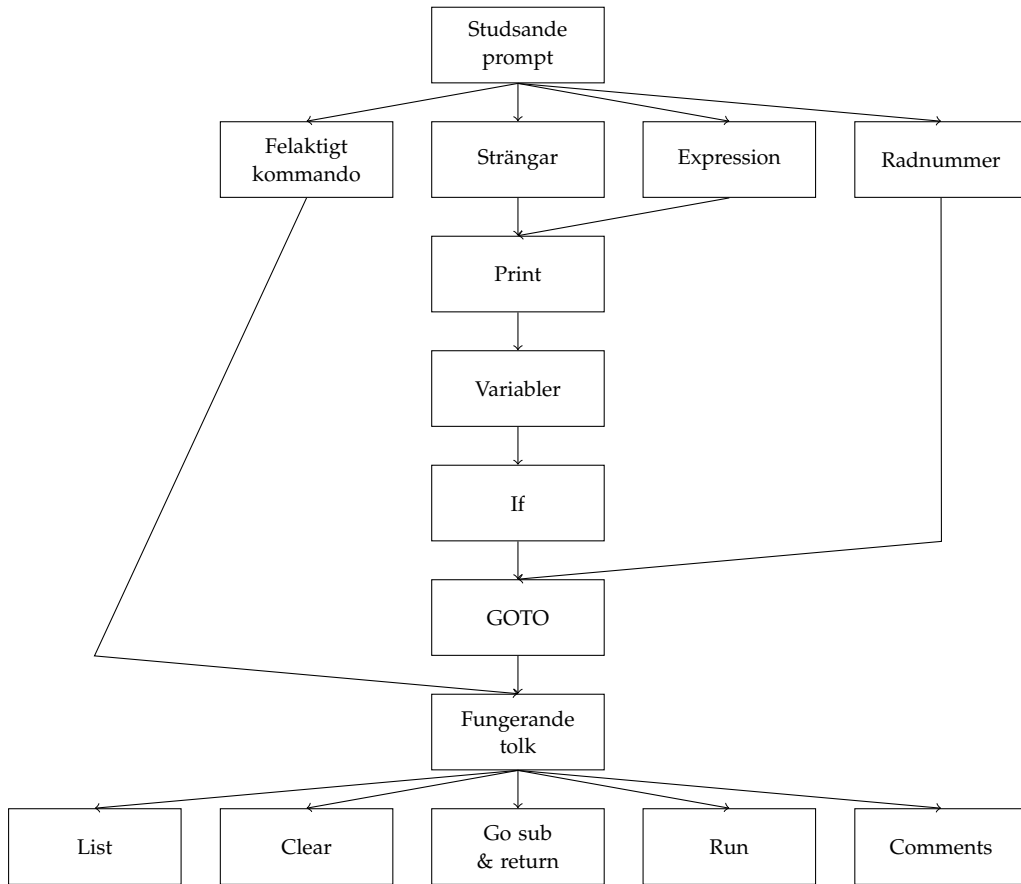


Fig. 1: Milstolpar för en Tiny BASIC-tolk.

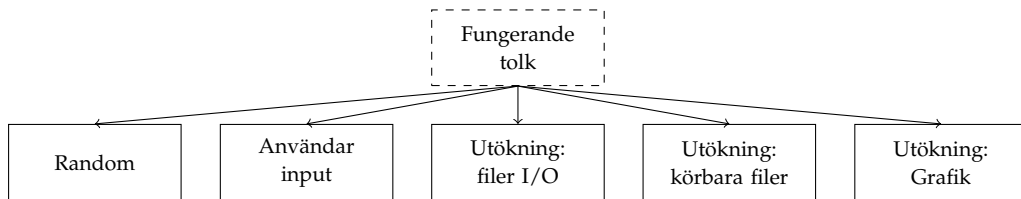


Fig. 2: Milstolpar för en Tiny BASIC-tolk.

samt att allt användaren har skrivit in när hen trycker på enter ska skrivas ut igen.

- **Felaktigt kommando:** När en användare skriver in något i prompten ska det vara giltig indata och om det inte är det behöver användaren få reda på vad hen skrev in, vad som är fel och vad som förväntades. Det som implementeras i den här milstolpen kommer behöva att utökas när andra milstolpar implementeras i och med att mer funktionalitet blir implementerad och därmed giltig.
- **Strängar:** En väldigt viktig sak i ett programmeringsspråk (som ska interagera med en användare) är strängar. Därför behöver er tolk kunna hantera dem.
- **Expression:** Expressions är mer eller mindre matematiska uttryck i Tiny BASIC. Utan dem blir det väldigt jobbigt att skriva intressanta program.
- **Radnummer:** Tiny BASIC har en väldigt intressant tolkning av radnummer som är starkt kopplat till vad som sparas i minnet. Se bakgrunden för mer information. Det här kan vara en av de klurigare sakerna att implementera. Ett tips är att tänka på att Java är objektorienterat.
- **Print:** Utan en print-sats är det omöjligt för användaren att skriva ett command line interface (CLI) program som kommunicerar med användare. Milstolpen går ut på att implementera de print-satserna som finns i språket.
- **Variabler:** Likt de tidigare milstolparna är variabler en av de grundläggande blocken för ett programmeringsspråk (som inte manipulerar minnet direkt).
- **If:** Tiny BASIC har två olika varianter av if-satser vilka båda ska implementeras i den här milstolpen.
- **GOTO:** GOTO är en väldigt grundläggande sats som finns i en del språk. I grund och botten säger den att man ska gå till en rad i koden. I moderna språk som kan använda dem anses det oftast vara väldigt fult att använda dem då de säker läsbarheten men i Tiny BASIC är det den enda möjligheten att iterera.
- **Fungerande tolk:** Vid det här laget bör ni ha en fungerande tolk som nästan är Turingkomplett (varför är den inte det?). Ta en steg tillbaka och se till att er kod inte har några buggar och att den är väl dokumenterad innan ni fortsätter med resterande milstolpar.
- **List:** List är en intressant sats i Tiny BASIC som absolut inte ska blandas ihop med datastrukturen listor. Satsen skriver ut de rader kod som finns registrerade i programmet, se dokumentationen för en mer exakt beskrivning.
- **Clear:** Eftersom koden som skrivs in i tolken sparas dynamiskt är det intressant att ha möjligheten att rensa ut all den sparade koden. Det är vad clear-satsen gör.

- **GoSub & return:** GoSub och return är två satser som används likt GOTO men ger möjligheten att gå tillbaka dit GoSub satsen var med return. Tänk på att tolken ska klara av att hantera nästlade GoSub anrop.
- **Run:** Den här satsen kan användas för att köra program som har skrivits in i tolken med hjälp av radnummer. Alternativt kan den användas som GOTO när satsen själv har ett radnummer.
- **Comments:** För att göra kod mer läsbar kan det vara värt att kunna skriva in kommentarer i programmet. I Tiny BASIC, till skillnad från de flesta programmeringsspråk är kommentarer en sats som inte gör något (ibland kallat "no-op" för "no operation").
- **Random:** Det här är den enda inbyggda funktionen som tolken kan använda sig av (vi bortser från funktionen som direkt arbetar med minnet). Den ska generera ett pseudoslumpartat tal.
- **Användar input:** Många program (men inte alla) kräver att de får någon typ av input från användaren. I det här fallet är det att användaren skriver in något från terminalen.
- **Utökning: filer I/O:** Vill man ha någon typ av sparad data är det intressant att kunna läsa och skriva till filer. Det här är något som inte finns i språket Tiny BASIC men milstolpen syftar till att ni utökar språket med den här funktionaliteten på ett lämpligt sätt.
- **Utökning: körbara filer:** Det är sällan roligt att behöva skriva in sin kod varje gång som man vill göra något. Det kan istället vara bättre att skriva sitt program i en fil som man sedan kör. Hur ni utformar den här funktionaliteten är upp till er (ett exempel är ett argument till tolken, likt python, och ett annat är en ny sats i språket som kör filen).
- **Utökning: Grafik:** Många program blir betydligt roligare om man har grafik. Det är även något väldigt komplext så den här milstolpen bör vara en av de sista som ni böjrar arbeta med.

## 5 Designbeslut att tänka på

I dagsläget finns det inga viktiga designbeslut som har uppmärksammats.