

# TDDD86 – Inlämningsuppgift

9 oktober 2018

Skriv en rapport om ETT av ämnena nedan. Ämnena har inte behandlats specifikt i kursen. Detta är en övning i att utforska egenskaper hos programmeringsspråk, använda kunskaperna du har om programmeringsspråk och paradig generellt och att skriva sammanhållet om dem.

Du uppmanas att använda flera referenser när du skriver din rapport. De givna länkarna ska ses som startpunkter för en informationssökning. Inom datavetenskap är Wikipedia ofta en pålitlig källa, men de bästa rapporterna använder information från ett flertal källor.

Var mycket noggrann med att uttrycka det du lärt dig med dina egna ord. **Plagiering är inte acceptabelt.** Vi använder plagiatkontroll verktyget Urkund för inlämningen. Mer information om inlämningen finns under uppgiftenshemsida <https://www.ida.liu.se/~TDDD86/info/inl.sv.shtml>.

Din rapport bör vara strukturerad kring ståndpunkten (hypotesen) som presenteras, med vilken du kan hålla med eller inte hålla med. I vilket fall som helst ska en bra rapport bestå av ett antal delar med lämpliga rubriker som "Introduktion" (formulerar ståndpunkten och ger sammanhanget för din rapport), en eller flera tekniska delar (ger bakgrund till scenariot i mer detalj, relevanta delar av programspråksdesign för ditt argument, poänger för och emot ståndpunkten och kriterier för att avgöra i frågan), "Slutsatser" (är du för eller emot?) och andra nödvändiga delar. Inkludera gärna figurer och enkla exempelprogram för att förtydliga texten. Rapporten ska vara ungefär 5–6 sidor lång (i till exempel 11 punkters fontstorlek). Det motsvarar ungefär 2500 ord.

## 1 Mobila användare och Molnet

*Ståndpunkt: Inom de närmsta fem åren kommer alla användare att ha skaffat lätta mobila enheter och informations- och beräkningsresurser finns i molnet. Detta kommer att kräva en ny approach till programmering, där man ser en tillbakagång till att betona applikationer som använder begränsade mängder beräkningskraft och minne samt återanvändandet av komponenter.*

För i tiden satt datoranvändare framför dumma terminaler (alltså maskiner utan egna beräkningsresurser) och arbetade med program och data på en fjärrserver. Ofta hade användare inte kunskap om den exakta kapaciteten hos servern eller var servern fanns någonstans. Efter att ha gått genom en fas med datorn på skrivbordet återvänder vi till situationen där resurserna finns någon annanstans och datorn framför oss har relativt begränsade beräknings- och minnesresurser. Skriptning i olika former blir vanligare allt eftersom användare kopplar samman komponenter som finns tillgängliga via fjärrinloggning.

Vilka följder för programspråksdesign har ett återvändande till mer lättviktiga applikationer? Håller du med ståndpunkten ovan?

### Referenser

<http://d1smfj0g31qzek.cloudfront.net/abovetheclouds.pdf>

<http://cloudcomputing.sys-con.com>

<http://en.wikipedia.org/wiki/MapReduce>

<http://boinc.berkeley.edu>

<http://scg.unibe.ch/research/piccola>

<http://searchitchannel.techtarget.com/feature/Cloud-applications-A-programming-languages-background>

## 2 Dynamisk typning är superbra för nybörjare

*Ståndpunkt: Fria från onödig syntax, speciellt variabeldeklarationer, kan språk som Python låta nybörjare inom programmering lära sig snabbare.*

Python är ett dynamiskt typat starkt typat objektorienterat programmeringsspråk med en stadigt växande användarbas. Flera universitet använder Python som första programmeringsspråk. Identifiera Pythons styrkor som undervisningsspråk och jämför dessa med Java och C++. Koncentrera dig särskilt på egenskaper som kan tilltala någon som lär sig programmera för första gången snarare än en erfaren programmerare i färd med att byta språk. Gör dynamisk typning programmering enklare? Säkrare? Är det lättare att lära sig koda? Får man enklare program?

### Referenser

<http://www.python.org>

[http://en.wikipedia.org/wiki/Dynamic\\_typing](http://en.wikipedia.org/wiki/Dynamic_typing)

<http://www.ariel.com.au/a/teaching-programming.html>  
<http://mcs.wartburg.edu/zelle/python/python-first.html>

### 3 Behandling av naturligt språk

*Ståndpunkt: Röstigenkänning har nu blivit så avancerad att alla användare borde kasta bort sina tangentbord och kommunicera med sina datorer enbart via rösten.*

Typisk science fiction inkluderar datorer som kommunicerar med människor på ett naturligt sätt, med talat in- och utdata, efterliknande vardagligt tal. Intåget av sofistikerade röststyrda applikationer (till exempel Siri, Dragon, NaturallySpeaking) antyder att framtiden redan är verklighet.

Har vi verkligen nått höjden av naturlig språkbehandling, eller finns det mer arbete att göra? Vilka problem kvarstår att lösa? Varför är naturligt språk så mycket svårare att behandla än datorspråk? Vilken sorts programmeringsspråk kan vi använda för att behandla naturligt språk? Traditionellt har språk som Lisp och Prolog använts, men skriptspråk är också populära.

#### Referenser

<http://www.ida.liu.se/labs/nlplab/>  
<http://research.microsoft.com/nlp/>  
[http://en.wikipedia.org/wiki/Natural\\_language\\_processing](http://en.wikipedia.org/wiki/Natural_language_processing)  
<http://www.alicebot.org/>

### 4 Syntetisk biologi är framtiden för datavetenskap

*Ståndpunkt: Nästa stora utmaning inom datavetenskap är att programmera livet.*

Genom att använda datorer för att förstå komplexa biologiska system står vi nu redo att börja programmera dessa system. Det här är en aspekt av *syntetisk biologi*. Om vi nu programmerar i DNA snarare än på en von Neumannarkitektur, hur påverkar det programmeringsspråket? Är det ens vettigt att tänka sig traditionella språk för att programmera levande organismer? Hur kommer ändringar i den underliggande arkitekturen att förändra sättet vi skriver program? Behöver vi nya programmeringsspråk? Hur kommer de att se ut? Eller finns det något fundamentalt i sättet vi beskriver algoritmer steg-för-steg i ett imperativt språk vi aldrig kommer undslippa?

#### Referenser

[https://en.wikipedia.org/wiki/Biological\\_computing](https://en.wikipedia.org/wiki/Biological_computing) <http://computer.howstuffworks.com/dna-computer.htm>  
<http://techcrunch.com/2014/03/10/teselagen-is-building-a-platform-for-rapid-prototyping-in-synthetic-biology>  
<http://www.organic-computing.org>

### 5 Visuell programmering

*Ståndpunkt: Nu när pekskärmar blir så vanligt kommer vi snart rutinmässigt skapa program visuellt.*

Program består normalt av ASCII-text men programmerare har länge försökt skapa mer visuella sätt att programmera, ofta med fokus på språk som använder diagram i stället för text. Till exempel låter Alice från Carnegie Mellon studenter programmera en 3D-värld genom att klicka och dra. Blockly är ett liknande projekt.

Skissa de viktigaste egenskaperna hos visuella programmeringsspråk och diskutera deras för- och nackdelar för programmeraren. Vilket syfte finns det med att göra ett språk visuellt? Är det möjligt att skriva stora och komplexa program på detta vis? Eller är visuell programmering bara för nybörjare och därför lite av en modefluga?

#### Referenser

<http://blog.interfacevision.com/design/design-visual-programming-languages-snapshots>  
[http://en.wikipedia.org/wiki/Visual\\_programming\\_language](http://en.wikipedia.org/wiki/Visual_programming_language)  
<http://www.alice.org>  
<http://code.google.com/p/blockly>  
<http://www.tersus.com/>

### 6 Big Data-programmering

*Ståndpunkt: Att hantera mycket data är samma sak som att hantera lite data; det finns bara mer av det.*

“Data Science” är ett snabbt växande område inom datavetenskap. Vi har nu ett stort antal sensorer som producerar ofantliga datamängder hela tiden. Vi skapar också själva mycket data via sociala medier. Vad gör vi med all denna data? Hur kan vi använda den på bästa sätt?

Skissa de viktigaste egenskaperna hos behandling av stora datamängder och diskutera speciella krav det kan ställa på programmeringsspråken. Går det bra att använda traditionella språk rakt av? Handlar det om att återanvända och kombinera komponenter? Eller MapReduce? Behöver vi ett nytt paradig för att kunna uttrycka oss elegant och effektivt om behandling av stora datamängder?

#### **Referenser**

<http://www.datascientists.net/what-is-data-science>

<http://www.ibmbigdatahub.com/>

<http://readwrite.com/2013/11/25/python-displacing-r-as-the-programming-language-for-data-science>

[http://en.wikipedia.org/wiki/Programming\\_with\\_Big\\_Data\\_in\\_R](http://en.wikipedia.org/wiki/Programming_with_Big_Data_in_R)