

Examinations- and kodningsguide

När vi examinerar projekt i kursen TSRT04 Introduktionskurs i Matlab så granskar vi både funktionalitet och kodningsstil. För att bli godkänd så räcker det inte att uppfylla projektets funktionalitetskrav. Det krävs också att koden är enkel att förstå och underhålla.

Kommentarer

Vi förväntar oss att koden kommenteras på ett rimligt sätt. Det ska vara möjligt att förstå vad ett kodsegment gör genom att läsa kommentarerna – man ska aldrig behöva fråga programmeraren. Eftersom lärarna som sköter examinationen kan komma från andra länder så måste ni koda och skriva kodkommentarer på engelska (som man brukar göra på företag och universitet).

Modularisering

Konstruera en kod som innehåller flera funktioner och skriv ett kort exemplskript som använder funktionerna för att lösa projektets problemställning. Detta skript ska ses som *ett* sätt att använda funktionerna och lektionsassistenterna kan be er att ändra på inparametrarna. Funktionerna ska skriva ut specifika felmeddelanden eller varningar om inparametrarna är orimliga eller uppenbart felaktiga (t.ex. negativa värden istället för positiva, eller felaktiga matrisdimensioner).

Namn på funktioner och variabler

Använd beskrivande namn som `months` och `yearlysalary`, istället för generiska bokstäver som `x` och `y`. Genom att läsa namnet på en variabel eller funktion så ska det vara möjligt att förstå dess innebörd. Om du använder MATLAB-editorn för att skriva kod så kan du ändra ett variabelnamn och trycka Shift+Enter för att ändra namnet överallt där variabeln används i samma m-fil. Detta kan användas för att förbättra namnen på existerande variabler.

Undvik att repeterar dig

Kopiera inte samma kodsegment om och om igen för att ändra detaljer i det. Försök att istället skapa en funktion som tar hand om funktionaliteten (för olika inparametrar) eller använd en loop.

Förstå när *for* och när *while* ska användas

for-loopar används när du vet *i förväg* hur många iterationer (upprepningar) som ska utföras. *while*-loopar används när antalet iterationer beror på beräkningarna som utförs inuti loopen.

Effektivitet

Vi förväntar oss inte att du ska skriva en väldigt beräkningsmässigt optimerad kod – faktum är att det ofta är bättre att göra koden lättläst än att spara några millisekunder i körtid. Däremot är det problematiskt om koden är så långsam att den tar flera minuter att köra. För att förstå varför din kod är långsam så kan du använda "profiler" (t.ex. genom att trycka på "Run and Time" i MATLAB-editorn). En vanlig anledning är att beräkningar utförs inuti i loopar (trots att de kunde utföras utanför loopen) eller att matris-dimensioner förändras inuti en loop (det är bättre att definiera matrisen med `zeros` innan loopen körs).