

# Hänga gubbe i Java

January 10, 2020

## 1 Uppgift

Er uppgift är att skapa en implementation i java av spelet hänga gubbe.

## 2 Mål och syfte

Syftet med uppgiften är att ni ska lära er hur mjukvaruutvecklingsprocessen fungerar för ett lite större projekt. Ni ska också öva alla programmeringsfärdigheter som vi arbetat med under hösten. Målet är att genomföra en systematisk utvecklingsprocess som resulterar i ett fungerande program.

## 3 Arbetsgång

Det första steget i en utvecklingsprocess är att förstå problemet. I det här fallet innebär det att ni måste få klart för er vad hänga gubbe är och vad ert program ska kunna göra. Jag hjälper er genom att bestämma att ni ska utgå från reglerna som definieras på spelets engelska wikipediasida. Jag bestämmer också att spelet antingen ska gå att spela mot en dator eller mellan två människor. Med det första alternativet slumpar datorn ett ord från en lista och spelaren gissar bokstäver och ord, det andra alternativet ger en spelare möjlighet att välja ord och den andre spelaren gissar bokstäver och ord. Ett tredje alternativ med flera spelare som gissar är möjligt men frivilligt.

När detta steg är genomfört ska problemet finnas nedskrivet i projektets `readme.md`. Nästa steg är att analysera vilka olika delar problemet består av. Det finns olika sätt att göra detta. Skapa en algoritmbeskrivning eller ett flödesdiagram (eller både och). Pseudokod är också ett alternativ om ni vill testa det, men det brukar fungera sämre i detta skede. Resultatet av

ert arbete ska sparas i projektet, antingen som ett markdowndokument, ett flödesdiagram eller i projektets wiki på github.

Flödesdiagrammet eller algoritmbeskrivningen blir grunden för hur ni sedan strukturerar er lösning. De olika delarna i algoritmbeskrivningen/flödesdiagramet bör bli egna metoder/funktioner i ert program. Tänk då på att varje metod bara kan returnera ett resultat så en metod ska inte göra flera saker. Skriv ner de olika delarna (metoderna) i en todo-list i **readme.md**.

Nu kan du börja välja vilken del av programmet du vill börja med. Lös den, gärna i en fristående fil, testa den och när du är klar välj en ny del av programmet. Var uppmärksam på att det kan dyka upp nya problem längs vägen. Då måste du undersöka dem och se om de påverkar ditt arbete så långt. Uppdatera alla dokument med ny information.

### 3.1 Arbetsgång

1. Förstå problemet
2. Dela upp och klargör problemet
3. Strukturera delarna
4. Välj vilken del du ska börja med, gör den
5. Undersök om det dykt upp nya problem som du måste lösa. I så fall gå till början av listan. Annars gå till steg 4.

## 4 Inlämning

Lämna in länken till ditt repo i classroom.