

Projektuppgift - Objektorienterad design och analys



Översikt & Generella riktlinjer

Det här är den projektuppgiften i kursen *Objektorienterad design och analys*. Inlämning sker på Omniway, genom att en länk till antingen ett Gitlab- eller ett Github-repo med projektkoden klistras in i inlämningsrutan för uppgiften. Om repot är privat behöver kontot @viktorsvensson (Gitlab) alt. @viktorsvensson502 (Github) ges tillgång till projektet.

Koden som lämnas in skall kunna kompileras och innehålla sådan funktionalitet som anges i det här dokumentet. Det krävs inte att programmet kan (fel)hantera ogiltig användarinput eller andra edge cases som avviker från det förväntade flödet, annat än där så anges.

Uppgiften är uppdelade i en G-del och en VG-del. G-delen genomförs enskilt eller i grupp om upp till 4 personer. VG-delen görs enskilt eller i grupp om upp till 2 personer. En startgenomgång av uppgiften för att reda ut ev. oklarheter eller frågor kommer hållas tisdagslektionen den 26/9.

Senaste inlämningsdatum är söndagen den 8:e oktober, kl 23.59

Uppgift att genomföra

G-del

Ert uppdrag är att utveckla ett terminalbaserat sten-sax-påse-spel för 1 spelare.

- En spelare skall kunna välja sitt namn, starta ett nytt spel och avsluta systemet genom någon form av terminalbaserat meny-system. (Den exakta strukturen med menyer, ev. submenyer etc. är valfri)
- Ett nytt spel skall inte kunna skapas om användarens namn inte är valt. När ett nytt spel startas skall ett nummer för "först till" kunna anges - dvs en användare skall kunna välja t.ex. 3, och då spela så många rundor mot datorn som krävs tills någon har vunnit 3 rundor. I slutet skall slutställningen visas, t.ex. *3-1 till datorn*.
- Startas spelet skall en omgång av sten-sax-påse genomföras mot datorn.
- Systemet skall vara uppbyggt med hjälp av *minst* ett etablerat design pattern.
- Ett textdokument skall bifogas där ni söker identifiera och relatera er kodbas till objektorienterad design och objektorienterade principer. T.ex. - vilka strukturer ser ni i er kod som implementerar objektorienterade koncept? Vilka för- och nackdelar kan en sådan koncept haft för att lösa olika typer av problem relaterade till ert projekt?
- En avslutande muntlig genomgång bokas in med läraren under någon av de avslutande lektionerna, för att gå igenom vad ni har gjort, ev. hinder ni har stött på under projektets gång och hur ni har tagit er an dem, etc.

VG-del

Utöver de krav som anges för VG-delen skall systemet vidareutvecklas så att

- Spelaren skall kunna se en historik - varje avslutad match mot datorn bör avslutas med att resultatet från varje runda sparas i en historik, där man kan se vilket drag respektive person gjorde i varje rund och vad den totala slutställningen sist blev.
- Spelaren skall kunna möta 3 olika sorters motståndare - "Slumpis" som slumpar sitt drag, "Klockis" som baserar sitt drag på vad klockslaget är samt "Namnis" som baserar sitt drag på vad spelaren heter.
- Spelaren skall kunna se statistik - hur många vinster har användaren totalt, hur stor andel av alla matcher har användaren vunnit samt hur stor andel av matcherna har spelaren vunnit av sina matcher mot respektive typ beskriven ovan (Slumpis, klockis, namnis)
- Minst ett design pattern (som inte är Singleton eller Builder) behöver finnas i systemet, och minst en tillämpning av någon av principerna i SOLID behöver ni kunna peka på i ert system.
- Systemet nyttjar Streams *minst* en gång någonstans i programmet där så kan vara lämpligt.
- Likt ovan - ett dokument som reflekterar kring hur ert slutresultat relaterar till objektorienterade koncept.

Bonus om man önskar ytterliggare utmaning - lägg till en möjlighet att spela en slutspelsturnering mot datormotståndarna, se om du kan implementera den med m.h.a en datastruktur som kallas för binärträd / binary trees (Du kommer behöva söka upp kunskap på egen hand, men ett starttips:

http://www.it.uc3m.es/java/2011-12/projects/game-modeling/3/guide_en.html)