

Peer-Review 2

Objektorienterad programmering med Java

Emil Björklund - embj3739
emilbjorklund@live.com

Källkod: Daren Domingo

March 18, 2019

Inledning

- IDE: Eclipse IDE v4.5.0
- Java version: JDK 8 update 191
- OS: MacOS Mojave

Review

Programmet var lätt att köra och exekverade utan problem. Följande testfall utfärdades och klarades:

- Normalt testfall: Input gavs efter spelets regler, tills människa vann.
- Onormalt testfall: Ett högt antal sticks att börja med, gick att mata in upp till maxtal för Integer. Input är även skyddat för negativ inmatning.
- Onormalt testfall: Spelaren försöker bryta mot reglerna som att ta bort mer än hälften eller mindre än 1.

Vissa scanner inputs från konsollen är skyddade med try catch funktionalitet som i klassen Player till exempel. I klassen Nm finns det dock scanner inputs som ej är skyddade med try catch något som skulle kunna implementeras för att öka tillförlitligheten hos programmet. Lösningen är relativt lik min egna då jag också använder rand från standardbibloteket för att skapa slumpmässiga drag från datorn. I detta program så skyddas rand med try catch precis som i min implementation. Koden är mycket lättläst med få kommentarer. Kommentarer är dock inte nödvändiga på många ställen då koden är självbeskrivande. Metoderna är döpta efter dess funktionalitet som t.ex. `getSticks()` och `removeSticks()`. Klasserna `computer` och `player` har båda superklassen `common` för dess gemensamma implementation vilket är ett snyggt upplägg.

Delay funktionen för kan tas bort då programmet upplevs långsamt, programmeraren har dock en poäng med att man vill fördröja datorns drag för att skapa en bättre känsla men jag anser att det är onödigt. Rapporten skulle kunna beskriva lite mer om hur man kör programmet som en readme. Vilken `.class` fil innehåller main-konstruktorn t.ex. Rapporten kan även delas upp i underrubriker för lättare läsning.

I sora drag är koden väldigt välskriven och definitivt en inspirationskälla för senare arbete. Korta IF-satser, självbeskrivande variabler samt metoder och lagomt luftig kod anser jag är dom största styrkorna. Upplägget med klasser är mycket likt min egna implementation med en superklass, en klass för människa och dator samt en klass för själva spelet.