











 chasacademy-sandra-larsson / js--oop-poker









[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Security](#) [Insights](#)





 1 star  0 forks  1 watching  1 Branch  0 Tags [Activity](#)


 Public repository




 main  1 Branch  0 Tags  











[Code](#) 

 chasacademy-sandra-larsson Update README.md 2 weeks ago  

 README.md Update README.md 2 weeks ago

 README  

## Skapa lämpliga klasser för Card, Player och Deck. Använd dig av ES6 Classes

### Del 1

Skapa en datastruktur med en "normal/poker" kortlek med 52 kort, inga jokrar. Skriv ut kortleken i konsollen, varje kort ska innehålla sin färg, sitt namn och sitt värde (så att det går att räkna med korten). Blanda kortleken, skriv ut den blandade kortleken i konsollen.

### Del 2

Skapa två spelare "Slim" och "Luke", de ska kunna hålla varsin hand med kort. Skriv ut spelarna i konsollen. Dela ut 5 kort vardera till spelarna. Skriv ut kortleken (nu med 42 kort kvar) och spelarna (nu med 5 kort var) i konsollen. Skriv också ut spelarnas händers sammanlagda numeriska värde.

### Del 3

Låt spelarna lägga slänga 2 kort var i en kashög. Låt spelarna dra 2 nya kort var ur kortleken. Skriv ut kortleken (nu med 38 kort kvar) och spelarna (med 5 kort var) i konsollen. Skriv också ut spelarnas händers sammanlagda numeriska värde.

### Del 4

Låt spelarna slänga alla sina kort i kashögen. Flytta alla kort från kashögen till kortleken. Nu ska spelarna och kashögen ha 0 kort och kortleken ska ha 52 kort. Blanda kortleken igen och skriv ut den i konsollen.

## VG - nivå

Steg 5 - 8. Skapa lämpliga klasser Dealer, Game etc



## Del 5

Skapa en ny klass som heter Dealer som hanterar allt som har med kortleken att göra. Dealern ska bland annat kunna: ...”äga” en instans av klassen Deck. ...blanda kortleken ...dela ut kort ifrån kortleken till spelarna. Låt delarn blanda kortleken och dela ut en varsin hand till de två spelarna ”Slim” och ”Luke” via dealern. Skriv ut spelarnas händer.

## Del 6

Skapa en ny klass som hanterar valideringen av spelarnas händer. Klassen ska ha ”statiska” metoder, alltså de anropas genom klassnamnet, inte genom en instans. Skapa en metod som tar in en lista med spelare och utvärdera deras händer utifrån kortens numeriska värde, och returnerar resultatet på ett snyggt sätt. Dela ut händer till de två spelarna och låt den nya klassen validera deras händer.

## Del 7

Skapa en Game-klass som sköter hela pokerspelet. Det enda main ska göra är att skapa en instans av klassen Game och anropa dess metod startGame. Game-klassen ska ha en lista med spelare och en instans av Dealer. Kom ihåg att dealern äger instansen av Deck. Skapa en metod som heter addPlayers som uppmanar användaren att skriva in antalet spelare (minst två) och sen respektive namn på de spelarna. Utifrån det så ska så många instanser av klassen Player skapas och läggas in i listan med spelare. Skapa metoden startGame och skriv in logik där ett par spelare skapas, de får varsin hand, korten valideras och skriv ut vinnaren på ett snyggt sätt.

## Del 8

Skapa en game-loop i startGame-metoden som kan hantera att man kör flera rundor. Spelare skapas, de får varsin hand. Man har en slängningsrunda, man får nya kort och sen valideras det och en vinnare utses. Det är en runda. Man kan utöka till fler rundor genom att öka antalet iterationer som loopen går. I varje runda ska de olika spelare kunna välja vilka kort de ska slänga. Alltså bör de få se sin hand varje runda, och en metod för att slänga ett kort på en specifik indexplats bör skapas i spelarens klass eller något liknande.

## Frivilligt

• Försök att skapa en mer verklig validering av korten. Bar två är trice och så vidare istället för kortens numeriska

### Releases

No releases published

### Packages

No packages published