

# Min Kursrapport

DV1610

Objektorienterade webbt teknologier  
(mvc)

Jon Kullberg

jonkullberg@gmail.com

joku17

v1.0.0

Mars 16, 2021

Läsperiod 4

Våren 2021

Institutionen för datavetenskap (DIDA)

Blekinge Tekniska Högskola (BTH)

Sverige

# Innehåll

<b>1</b>	<b>Objektorientering</b>	<b>3</b>
1.1	Förkunskaper . . . . .	3
1.2	PHPs modell för klasser och objekt . . . . .	3
1.3	Uppgiftens kodbas . . . . .	4
1.4	Mitt spel . . . . .	4
1.5	PHP The Right Way . . . . .	4
1.6	TIL . . . . .	5
<b>2</b>	<b>Controller</b>	<b>6</b>
2.1	Nytta av flödesdiagram och pseudokod . . . . .	6
2.2	Förklaring av några objektorienterade konstruktioner . . . . .	6
2.3	Mitt spel . . . . .	7
2.4	Kodstruktur . . . . .	7
2.5	TIL . . . . .	7
<b>3</b>	<b>Enhetstestning</b>	<b>8</b>
<b>4</b>	<b>Ramverk</b>	<b>9</b>
<b>5</b>	<b>Autentisering</b>	<b>10</b>

<i>INNEHÅLL</i>	2
<b>6 To be defined</b>	<b>11</b>
<b>7 Projekt &amp; Examination</b>	<b>12</b>
7.1 Projektet . . . . .	12
7.2 Avslutningsvis . . . . .	12

# Kapitel 1

## Objektorientering

### 1.1 Förkunskaper

Vi har ju haft en pythonOOP-kurs på den här utbildningen redan, som jag tyvärr inte har avslutat ännu (ska göra det i sommar). Jag har alltså bekantat mig ganska väl med konceptet, men har fortfarande en del kvar att verkligen förstå, och desto mer att bemästra.

Några saker som jag funderade på under detta moment var i vilken utsträckning jag faktiskt nyttjade objektorientering, och ifall det hade hjälpt mig att göra det mer. Jag skapade DiceHand-objekt för spelaren och datorn med antal tärningar och kast, men i slutändan sparade jag nog ändå den mesta av datan i session-variabler, snarare än som objektens properties.

### 1.2 PHPs modell för klasser och objekt

En klass är en samling kod som innefattar funktioner och variabler, som när de används i en klass kallas för metoder och medlemsvariabler.

Ett klassiskt exempel är att man kan likna en klass vid en ritning, som innefattar funktionalitet och egenskaper. Varje objekt, dvs. varje instans av klassen, som konstrueras efter ritningen, får därmed denna funktionalitet och dessa egenskaper.

Vanligtvis döljer man objektets eller klassens inre funktioner och variabler för användaren och yttre miljöer, och tillåter interaktion med dessa endast via specifika metoder som får hämta och modifiera medlemsvariabler på förutbestämda sätt, så kallat API.

### 1.3 Uppgiftens kodbas

Jag undersökte inte riktigt alla beståndsdelar i den tillhandahållna kodbasen, men kikade igenom en hel del. Den framstår som kompetent och ganska lätt att förstå. Kanske att den skulle kunna fyllas på med lite mer dokumentation eller lite mer exempelkod som är relevant för detta kursmoment, så att det går snabbare att förstå flödena. Specifikt hur olika interaktioner som länkar eller formulär flödar genom routern och hur man får med sig data in på rätt plats igen efteråt.

### 1.4 Mitt spel

Jag löste uppgiften genom att göra de efterfrågade klasserna och deras metoder. Sedan en fil där det mesta av logiken ligger, som anropar dessa och sparar resultaten i variabler. Dessa skickas i sin tur till vyn, där saker och ting visas eller döljs beroende på vad variablerna innehåller, dvs. hur långt man kommit i spelet, och dess utfall. Interaktion löste jag genom POST/GET, men mest med olika länkar via routern som anropar funktioner eller sätter variabler.

Min kod är i sitt nuvarande tillstånd ett resultat av trial and error, snarare än god planerad arkitektur. Såvitt jag har kunnat testa så fungerar allt som det ska, men jag skulle egentligen vilja göra om allt från grunden så att logiken och flödena blir klarare. Jag tror och hoppas att man i takt med att erfarenheten växer lär sig att konstruera bättre kod.

Jag ser stor förbättringspotential, främst genom att kapsla in så mycket mer i funktioner, och konstruera dessa så att de går att återanvända till många olika saker. Såhär i efterhand när allt är klart så känns det som en ganska lagom svårighetsgrad, men jag känner mig för långsam som behövde så mycket tid på mig med det. Börjar bli riktigt trångt i hjärnan nu när man ständigt pumpar den med ny information och aldrig riktigt får landa i en sak och bemästra den. Att stylea spelet vore så klart en rätt god idé också.

### 1.5 PHP The Right Way

Det jag hittills haft mest glädje av i denna artikel är avdelningen Coding Practices, The Basics. Bra att fräscha upp grundreglerna lite när man bollar så många olika språk samtidigt hela tiden.

## **1.6 TIL**

Inte helt lätt att smala ner det till en enda sak. Jag skulle nog säga att en lärdom var att jag framöver faktiskt bör börja skissa upp ett flöde och en plan innan jag börjar koda något såhär stort.

## Kapitel 2

# Controller

### 2.1 Nyttan av flödesdiagram och pseudokod

Det var intressant att testa på detta. Min kod i kmom01 är ostrukturerad och ogenomtänkt, och det hjälpte så klart en hel del i kmom02 att först designa lösningen innan den implementeras. Med det sagt så frångick jag ändå designen när jag skrev koden, då jag märkte att det gick att nå samma resultat med en betydligt enklare struktur.

Jag skulle nog säga att jag använde mig av top-down för planeringen.

### 2.2 Förklaring av några objektorienterade konstruktioner

Arv är ett koncept som beskriver hur en klass kan ärva alla egenskaper från en annan klass, men dessutom specialiserar eller utökar funktionaliteten.

Komposition är när ett objekt är uppbyggt av, eller innehåller, andra objekt. En DiceHand i form av en array som består av ett antal Dice t ex.

Interface är kod som går ut på att snabbt och överskådligt visa vad man kan göra med klassen. Ungefär som när man sätter sig i en bil och ser en ratt, ett antal pedaler, och någon sorts växelspak. Det är inte så viktigt exakt hur de är konstruerade, utan poängen är att man vet ungefär vad de kan göra ändå.

Trait är ungefär som att importera en extern funktion in i en klass. Detta underlättar när flera klasser använder liknande funktioner, vilka man då kan ha i en enda fil istället för att upprepa överallt.

## 2.3 Mitt spel

Jag löste uppgiften genom att modifiera DiceHand och Dice så att man kan spara en enskild tärning, som då inte rullas nästa gång man rullar DiceHand. Därefter är det mer eller mindre bara att rulla DiceHand[round] där round är 1-6, och sedan räkna antalet Dice som överensstämmer med round, som sedan skrivs ut och summeras. Dvs. hur många Dice med resultatet 1 som vi har i vår DiceHand[1]. Resultatet sammanställs i en övergripande array som man slutligen summerar och utvärderar ifall resultatet räcker till för att få bonuspoäng.

Användaren ser detta i form av grafisk rendering av DiceHand, statustext som uppdateras genom spelets gång, samt länkar för att rulla, spara tärningar, gå vidare, och nollställa session för att starta om spelet.

Jag blev mer nöjd med detta kursmoment än det förra, men misstänker att det finns en mängd sätt att förbättra koden på. Exempelvis misstänker jag att jag har lite för mycket logik i min view istället för i kontrollern, vilket kanske ställer till med problem när en eventuell dator ska spela spelet.

## 2.4 Kodstruktur

Jag gillade helt klart dessa nya router/request/controller-strukturer bättre än de första. Det var enkelt att få dem att göra som man ville och att förstå hur de fungerar.

## 2.5 TIL

Jag lärde mig mycket under detta moment. När jag väl hade designat pseudo/flöde, byggt funktionerna i klasserna och lekt lite med dem, så gick det väldigt snabbt att bygga/klistra ihop det slutgiltiga spelet. Jag hoppas att varje sådant här moment gör det initiala förberedande arbetet enklare och snabbare.



## Kapitel 3

# Enhetstestning

Här skriver du din redovisningstext för detta kursmoment.

## Kapitel 4

# Ramverk

Här skriver du din redovisningstext för detta kursmoment.

## Kapitel 5

# Autentisering

Här skriver du din redovisningstext för detta kursmoment.

## Kapitel 6

### To be defined

Här skriver du din redovisningstext för detta kursmoment.

## Kapitel 7

# Projekt & Examination

Här skriver du din redovisningstext för detta avslutande kursmoment.

### 7.1 Projektet

Här skriver du din redovisningstext rörande projektet.

### 7.2 Avslutningsvis

Här skriver du de avslutande orden om kursen.