# Inledning:

Under åren jag spenderat på gymnasiet har jag bland annat lärt mig grunderna i webbutveckling och programmering. Jag har funnit dessa ämnen intressanta nog att jag utanför skoltid har fördjupat mig i dem för att lära mig mer. Ett möjligt sätt att vidareutveckla mina kunskaper på dessa områden är att konstruera egna program och webbsidor.

Ett annat intresse för mig är att se på skidskytte, en vintersport som kombinerar längdskidåkning med skytte. Mitt intresse uttrycks främst genom att jag tittar på skidskyttetävlingar på elitnivå, vilket jag anser vara mycket god underhållning. Dock varar tävlingssäsongen endast ca 4 månader per år.

Från denna brist på skidskytte under den större delen av året uppstod en idé: Skulle jag, med hjälp av mina programmeringskunskaper, kunna konstruera ett program som simulerar skidskyttetävlingar? Mer specifikt ville jag skapa ett program som genererar inte bara ett slumpmässigt resultat utan resultat som var någorlunda rimliga baserat på åkarnas verkliga prestationer. På detta sätt fick jag idén till detta projekt.

Det jag vill komma fram till med detta arbete är en webbsida med en funktionell simulator av skidskyttetävlingar som genererar något rimliga resultat. Syftet är att programmet ska ha ett underhållningsvärde för skidskytteintresserade som tycker att det vore roligt att kunna simulera tävlingar särskilt utanför tävlingssäsongen. Personligen var idén även ett sätt för mig att träna på webbutveckling och programmering genom att konstruera en webbsida med ett program och valde detta ämne eftersom jag själv har ett intresse av skidskytte. Jag tror att något som har att göra med webbutveckling och programmering skulle kunna bli en framtida karriär och därmed tänkte jag att det fanns ett värde i att öva på att konstruera en webbsida med program.

# Genomförande:

Inför själva genomförandet var alltså mitt mål att konstruera ett program som simulerar skidskyttetävlingar med rimliga resultat, till nytta för skidskytteintresserade som finner ett underhållningsvärde i det. För att uppnå detta planerade jag inte att använda något annat fysiskt material än min dator.

Den första frågan jag tvingades ställa mig var vilket programspråk jag skulle använda. Eftersom jag inte ville lära mig ett nytt språk stod valet mellan de två jag har erfarenhet av: Java och JavaScript. Jag valde JavaScript av flera anledningar. För det första så höll jag redan på att programmera andra saker i Java vid samma tid, vilket för mig gjorde det naturligt att skriva detta program i JavaScript för att öva även på det språket. För det andra kunde jag med JavaScript även inkorporera programmet i en webbsida och därmed även träna på webbutveckling samtidigt som programmering. Dessutom ville jag även lära mig mer om hur man använder JavaScript för att styra webbsidors utseende. För att underlätta programmeringen bestämde jag mig för att använda jag mjukvaran WebStorm, en så kallad integrerad utvecklingsmiljö (IDE). Att jag valde just Webstorm beror på att det är den IDE jag tidigare har använt mig av för programmering i JavaScript.

Jag förväntade mig att jag skulle behöva hämta in information för att kunna göra klart programmet. Den enda egentliga källan som jag använt tidigare för inlärning av JavaScript är det som finns på webbsidan <https://developer.mozilla.org/en-US/docs/Web/JavaScript>. En av mina tankar inför programmeringen var att jag skulle lära mig hur och var jag kan inhämta information jag behöver för att konstruera olika program.

Redan innan jag börjat skriva någon kod visste jag att jag skulle behöva tillgång till statistik baserad på verkliga tävlingar som beskriver de olika elitåkarnas egenskaper för att skapa någorlunda rimliga resultat. Åtminstone behövde jag känna till samtliga åkares genomsnittliga åkhastighet och skytteprocent. Jag visste dock sedan tidigare att en mängd detaljerad information angående skidskytte på elitnivå fanns på webbsidan <https://www.realbiathlon.com/>, inklusive det jag var i behov av.

## Grunder:

Jag började kodandet med att konstruera två funktioner som omvandlar tid. Till beräkningar i själva koden visste jag att det skulle vara enklast att mäta tid i antal sekunder, men i den slutgiltiga webbsidan ville jag att tid skulle beskrivas på ett mer lättläst sätt, t.ex.’32.58.7’. Jag konstruerade därför två funktioner som omvandlade tid mellan de olika formaten.

### Data om åkare:

Information om olika åkares egenskaper inhämtades från <https://www.realbiathlon.com/>. Dessa innefattade åkarens namn, nationalitet, genomsnittlig åkhastighet jämfört med medianåkaren samt genomsnittlig skytteprocent i liggande och stående skytte. Senare lade jag även till genomsnittlig tid på skjutvallen i liggande respektive stående skytte.

All denna data sparade jag i ett fält som ursprungligen kallades athleteInfo. Den första versionen av detta fält var ett tvådimensionellt fält med ett fält i fältet för varje åkare, men senare lagrade jag denna data i ett fält bestående av objekt som alla innefattade en åkares egenskaper. I den nuvarande versionen finns två fält, ett för damerna och ett för herrarna. Det hjälpte när jag senare ville att användaren skulle kunna välja mellan damer och herrar i sin simulation.

### Åktid:

För att simulera åktider skapade jag en funktion. Funktionen skulle läsa in en åkares åkhastighet och returnera med den åkarens totala åktid för tävlingen. Jag hämtade data om de olika åkarnas genomsnittliga åkhastighet jämfört med medianåkaren från <https://www.realbiathlon.com/>. Funktionen jämför åkhastigheten med en variabel som innehåller medianåkarens åktid. Utifrån det beräknas åkarens åktid. Denna version av funktionen fungerade i att den gav en fullständig åktid för varje åkare, men saknade något viktigt. Eftersom all data som användes var konstant gav den exakt samma åktid för samma åkare varje gång programmet kördes. I verkligheten beror åkhastigeten för dagen delvis på dagsform och valla. För att simulera detta valde jag att lägga in en viss slump i funktionen för åktidsberäkning.

När jag senare ville ha med olika mellantider valde jag att ändra funktionen till att ge som resultat åktiden på ett visst varv. I samband med denna ändring experimenterade jag med att använda medianåktiden för de individuella varven som grund för beräkningarna. Men efter att ha inhämtat den informationen från <https://www.biathlonresults.com/#/start> insåg jag att skillnaden i medianåktid för olika varv i samma tävling var så liten att det i praktiken var irrelevant. I den färdiga version används därför som variabel den totala medianåktiden delat med antal varv.

Att beräkna varvtider separat fungerade, men ledde till en sak jag inte ville. Eftersom varje varvtid beräknades separat var det ofta stora skillnader i åkplacering per varv för samma åkare i en tävling. Jag tyckte att det var lite orealistiskt. I verkligheten är oftare de åkare som är starka för dagen snabba på alla varv och de sämre åkarna långsamma på alla. Därför skapade jag en ny funktion som modifierar åkarens genomsnittliga åkhastiget slumpmässigt för att simulera dagsform. Det värdet används sedan för att beräkna åkarens individuella varvtider med en mycket mindre slumpfaktor. Med detta åker varje enskild åkare ungefär lika fort på alla varv.

### Skytteresultat:

För att räkna ut skytteresultat konstruerade jag en funktion som simulerar en skjutserie och returnerar antalet bom i serien. Funktionens parameter är skytteprocenten för den relevanta åkaren i liggande eller stående. I funktionen körs en for-sats som jämför ett slumpmässigt genererat värde med skytteprocenten för att avgöra om ett skott är träff eller bom. Denna sats körs fem gånger innan funktionen returnerar med antal bom under serien.

### Beräkning och utskrift:

Funktionerna jag dittills konstruerat kunde skapa värden för åktider och skytteresultat, men för att simulera en hel tävling behövde jag kunna spara dessa värden någonstans för att senare skriva ut dem. Jag valde att spara all information om själva tävlingen som komponenter i ett tvådimensionellt fält som jag kallade raceData. Liksom athleteInfo bestod fältet av fält som var och en representerade en enskild tävlande. När programmet kördes åberopades funktionerna som räknade ut åktider och skytteresultat för varje åkare, vars returvärden lagrades i raceData. Sedan skulle fältet sorteras utifrån slutgiltig tid så att åkarna hamnade i korrekt resultatordning innan resultatet skrevs ut i en tabell.

För att skriva ut resultatet på själva webbsidan skapade jag en funktion som skulle skriva ut det slutgiltiga resultatet i en lättläslig tabell utifrån information i raceData. Dock behövde raceData sorteras innan tabellen skapades för att åkarna skulle listas i rätt ordning, vilket skapade ett problem. Utskriften krävde att jag använde information från både athleteInfo (t.ex. namn) och raceData (t.ex. antal bom). Jag kunde sortera raceData i fallande placeringsordning men visste inte hur jag skulle göra samma sak i athleteInfo då de relevanta komponenterna för sortering (slutgiltig tid) inte fanns i det fältet. Det medförde att resultaten inte visades med rätt åkare i utskriften.

För att lösa detta problem lade jag till satser som när tävlingen påbörjas tilldelar åkarnas egenskaper från athleteInfo till deras respektive fält i raceData. Då all information nu fanns i samma fält kunde det sorteras och skrivas ut korrekt.

I och med denna ändring fungerade programmet till den grad att det ursprungliga syftet med projektet i princip var uppnått: Att simulera en skidskyttetävling med någorlunda rimliga resultat. Jag valde dock att fortsätta med programmet då jag såg ett antal möjligheter för förbättringar.

## Förbättringar:

### Mellantider:

En av de självklara förbättringarna var att lägga till mellantider. Med mellantider skulle användaren kunna följa tävlingens gång istället för att direkt se slutresultatet, vilket adderar ett spänningsmoment.

För att uppnå detta ändrade jag först själva beräkningarna så att de räknade ut och lagrade tider och resultat för olika delar av tävlingar, exempel åktid på olika varv. Det medförde att koden för beräkning och utskrift blev mer komplicerad. Jag valde att dela upp den i två funktioner. Under en tävling körs först en funktion som tilldelar all data om olika varvtider, skytteresultat o.s.v. till dess respektive åkare. Sedan körs en funktion som löper igenom tävlingen kronologiskt och skriver ut den ”nuvarande” ställningen i tabeller som mellantider och slutar med att skriva ut slutresultatet.

### Konstanter:

I det tvådimensionella fältet raceData består varje inre fält av komponenter som innehåller all information om en specifik åkares tävling. Ju mer komplicerat jag gjorde programmet, desto fler komponenter använde varje inre fält. Det medförde att det blev i princip omöjligt att minnas vilket numeriskt värde motsvarade vilken information, vilket gjorde det svårt att komma ihåg vilken del av programmet som gjorde vad. Hela programmet blev svårläst.

Lösningen på detta fanns i användning av konstanter. Genom att tilldela indexvärden för olika åktider, skyttetider o.s.v. till konstanter blev programmet mer lättförståeligt och mer flexibelt eftersom jag nu lätt kunde ändra konstanters värde och lägga till nya utan att det påverkade programmets funktionalitet.

### Tid på skyttevallen:

Rangetime (genomsnittlig tid på skjutvallen) lades till för att inkorporera skyttehastighet i simulatorn, något som spelar en viktig roll i dagens tävlingar. Information om genomsnittlig tid på skjutvallen lades till objekten som beskriver åkarnas egenskaper och används tillsammans med en slumpfaktor för att simulera hur lång tid åkaren spenderar på skjutvallen.

### Val av herrar eller damer samt val av disciplin:

En förbättring jag ville få till var att låta användaren välja själv om hen vill simulera en herr- eller damtävling samt i vilken disciplin. I den första versionen simulerades en tävling automatiskt så fort webbsidan laddades, men om användaren själv skulle få välja vilken sorts tävling hen vill simulera måste hen få mata in den informationen först. Därför ändrade jag programmet så att det första som laddas är ett antal knappar som motsvarar olika tävlingsformer när de klickas.

Beroende på vilken knapp som klickas tilldelas ett antal variabler olika värden som bland annat gör att viss kod körs eller skippas beroende på om de är relevanta för tävlingsformen (i en sprinttävling ska exempelvis bara 2 skytten simuleras, inte 4). På det sättet kan programmet simulera både sprint- och distanstävlingar.

I den ursprungliga simulatorn hade jag behövt gå in i själva koden och byta ut alla åkarne för att byta från damer till herrar och vice versa. Men genom att lagra alla damer i ett fält och alla herrar i ett annat och kopiera ett av dessa fält till raceData för att användas i själva tävlingen beroende på vilken knapp som trycks i början är detta inte längre nödvändigt.

# 3. Utvärdering och reflektion

## 3.1 Hur det gick:

### Slutresultat i förhållande till målet:

Det främsta målet med skidskyttesimulatorn var som tidigare konstaterat att skapa ett program som simulerar skidskyttetävlingar med någorlunda rimliga resultat. Det nuvarande programmet uppnår detta mål med råge.

Personligen anser jag att programmet även uppfyller syftet att ge underhållning till skidskytteintresserade, utifrån det faktum att jag är skidskytteintresserad och finner underhållning i att simulera dessa tävlingar. Dock vet jag inte vilken åsikt andra skidskytteintresserade har om saken.

### Såhär gick det att följa planen:

Under konstruktionen följde jag mina planer relativt noga. Jag skrev all kod i WebStorm och använde programmeringsspråket JavaScript för att inkorporera simulatorn på en webbsida. All data om åkarna jag utnyttjade hämtade jag som planerat från realbiathlon.com.

### Användarvänlighet:

Användarvänligheten i programmet är, enligt mig, relativt god. Att simulera en tävling kräver inga särskilda kunskaper i programmering eller något annat, och resultaten skrivs ut i lättläsliga tabeller.

## 3.2 Reflektion

Under min tid på gymnasiet har jag lärt mig många saker. Bland dessa tillhör webbutveckling och programmering både de jag är mest intresserad av och ser mest möjlighet att ha nytta av i framtiden. Genom att jobba med detta arbete upplever jag att jag har fördjupat mina kunskaper i webbutveckling och programmering, främst genom att jag har fått mer erfarenhet av att använda mina kunskaper för att konstruera större program.

När det kommer till hur jag arbetade med projektet fanns det alternativa metoder jag kunde ha använt, som att utnyttja något annat programspråk för att skapa en skidskyttesimulator som uppfyller i princip samma syfte, t.ex. Java. Det finns även motsvarigheter till WebStorm som kunde ha använts som IDE för att underlätta kodandet, t.ex. Visual Studio Code.

Det finns ett antal idéer jag har för att vidareutveckla detta projekt någon gång i framtiden. En sak jag överväger är att göra webbsidan jag skapat tillgänglig för allmänheten, främst i syfte att lära mig själv hur jag ska gå tillväga för att uppnå detta. Även själva programmet skulle också kunna förbättras eller utökas med fler funktioner.

Ursprungligen fungerade enbart distanstävlingar, men senare lade jag till sprint till programmet. Det vore möjligt att lägga till fler discipliner, till exempel jaktstart och masstart. En anledning att jag inte ännu gjort det är dock ett simuleringen för tävlingar med gemensam start oundvikligen blir mindre realistisk än för tävlingar med individuell start. I exempelvis masstart påverkas åktiderna av att åkare hamnar i klungor och tar hjälp av varandra, vilket jag tror är någonting som vore ytterst svårt att återskapa någorlunda verklighetstroget i simulatorn.

Att förbättra beräkningen av åktider är svårt, eftersom det inte finns något riktigt objektivt sätt för mig att veta till exempel exakt hur stor slumpfaktorn som avgör dagsform skall vara. Inte heller vet jag hur jag skulle kunna inkorporera olika åkares profiler på olika varv, alltså om en åkare generellt sätt börjar eller avslutar starkare i spåret.

I verkligheten påverkar ett antal faktorer, bland annat vind och dimma, hur svårt det är att skjuta. Det vore möjligt att simulera detta genom att lägga till en slumpfaktor som påverkar träffprocenten för samtliga åkare under en viss tävling. Men då programmet redan ger en viss variation i exempelvis antal nollskyttar från tävling till tävling bara genom den inbyggda slumpfaktorn i beräkningen av skytteresultaten, vilket ger ett relativt realistiskt intryck. Därför är denna ändring ingen prioritet.

En idé, som jag inte tror har några stora nackdelar och som dessutom skulle förbättra underhållningsvärdet, är att låta användaren anpassa startfältet. Det skulle innebära att användaren får välja att köra tävlingen med standardstartfältet eller själv välja vilka åkare som ingår i startfältet. Just nu simulerar programmet en fiktiv tävling från nästa säsong. Därför består startfälten i programmen nu av de bästa skidskyttarna som kommer att tävla nästa säsong och deras egenskaper baseras helt på prestationer från den senaste säsongens tävlingar. Syftet med att låta användaren välja vore inte bara att låta hen ta bort vilka åkare hen vill, utan även att låta användaren lägga till åkare som inte kommer tävla nästa säsong till startfältet, exempelvis åkare som har lagt av. Som skidskytteintresserad tycker jag att det vore spännande att exempelvis låta Martin Fourcade tävla mot Johannes Thingnes Bö eller se hur Magdalena Forsberg står sig mot dagens damelit.

Det finns även sätt att ytterligare förbättra användarvänligheten. En tanke som slagit mig är att skriva ut tabellerna horisontellt. Det skulle göra det lättare att med blotta ögat jämföra resultatlistor från mellantid till mellantid.

# Källförteckning:

Real biathlon, <https://www.realbiathlon.com/>, Real biathlon, hämtad 2023-11-08

IBU, <https://www.biathlonresults.com/#/start>, International Biathlon Union (IBU), hämtad 2023-11-08

MDN Web Docs, JavaScript, <https://developer.mozilla.org/en-US/docs/Web/JavaScript>, Mozilla, uppdaterad 2023-04-12, hämtad 2023-11-10

W3Schools, JavaScript Tutorial, <https://www.w3schools.com/js/default.asp>, W3Schools, hämtad 2023-11-10

Joel Olawanle, JavaScript 2D Array – Two Dimensional Arrays in JS, <https://www.freecodecamp.org/news/javascript-2d-arrays/>, freeCodeCamp, hämtad 2023-11-08

# Bilagor