

## Inlämningsuppgift: OOAD för Bengans bowlinghall

---

### Beskrivning

Uppgiften handlar om att skapa en lösning för Bengans Bowlinghall som behöver ett system för att hantera sin verksamhet. Man har ett stort antal medlemmar/spelare som behöver registreras med kontaktuppgifter. Man har även en anläggning som består av ett antal banor. På dessa banor anordnas tävlingar som består av olika matcher. I varje match deltar två spelare. Varje match spelas över tre serier per spelare. Den spelare som har högst totalpoäng utses till vinnare.

Krav för att lösningen skall bli godkänd:

- Den skall fungera för ovanstående verksamhet och hålla reda på uppgifter, tävlingar, resultat och vinnare i matcher och tävlingar.
- Det skall finnas Use Cases (textuella) och Use Case-diagram som innehåller de viktigaste användningsfallen för systemet.
- Det skall finnas ett Klassdiagram som innehåller klasser i systemet. Klasserna ska vara indelade i en logisk paketstruktur.
- Det ska finnas ett Workflow diagram som visar flödet under en match.
- Sekvensdiagram som beskriver flöden från aktör, via system och de mest relevanta delarna av dess ingående objekt.
- Ett kodprojekt (behövs inget gränssnitt, enbart "backendkod") som skall innehålla minst två designmönster ur **Design Patterns - Elements of reusable Object-Oriented Software** på lämpliga ställen i projektet. *Namngivningen är viktig!*
- Det skall gå att verifiera med tester att det går att generera, lagra och hämta medlemmar och tävlingsinformation. Simulera en match med enhetstester och verifiera att rätt spelare utses till vinnare. Verifiera att den spelare som vunnit högst andel av sina matcher under angivet år returneras korrekt av systemet som det årets champion. (Inget rankingssystem behöver användas).
- Förloppet i systemet ska kunna följas i en systemlog, t.ex genom utskrift till consol.

### För VG krävs även att följande uppfylls:

- ändamålsenlig projektstruktur och ändamålsenlig namngivning av identifierare (paket, klasser, attribut, metoder...)
- interfacebaserad mockning av externa beroenden
- ändamålsenliga enhets- och integrationstester
- loggning sker till fil som är unik för varje körning

## **Betyg**

Det går att få VG, G och IG på uppgiften. Den är obligatorisk för att kunna få minst G på hela kursen.

## **Redovisning**

Redovisning sker genom att du visar läraren den fungerande uppgiften. Skicka INTE in något via mail! Senaste tidpunkt för redovisning är sista veckan på kursen innan avslutande föreläsning.