## Individuell uppgift - Quiz

Din uppgift är att skriva en quiz-applikation. En quiz-applikation är ett frågesport-spel.

* Din applikation ska vara objektorienterad. Du ska visa att du kan skapa lämpliga klasser. Det kommer att vara högre fokus på det i denna uppgift än i gruppuppgiften, eftersom ni kan börja att skriva era applikationer objektorienterat från början nu.
* Quizet ska hålla reda på en spelare. Det ska hantera spelarens namn, spelarens poäng i den aktuella omgången.
* Frågorna ska läsas in från <https://quizapi.io/> som levererar ett resultat i JSON.

Du måste använda ditt omdöme och göra en analys av kraven för att kunna leva upp till dem.

**G-krav**

* Spelet ska innehålla 10 frågor.
* Man ska kunna välja flera svar. Ibland kan ett alternativ vara rätt, ibland flera.
* När omgången är slut ska poängen visas och användaren ska få välja att starta ett nytt spel med nya frågor.
* Skriv minst en klass som innehåller minst en metod och minst en egenskap.
* Du får inte använda inline-css eller inline-event (t ex <p style="color:red" onlick="something();">)
* Du ska använda minst en array-funktion.
* Lämna in projektet som ett git-repo.

**VG-krav:**

* Allt som ingår för G-kraven.
* Låt användaren bestämma hur många frågor som ska visas. (5-10)
* Du ska visa att du kan skapa lämpliga klasser med lämpliga metoder och egenskaper. Du får inte använda globala variabler eller funktioner utanför klasser, förutom anonyma funktioner i event-lyssnare. (De behövs inte där heller, men du får.)
* Du ska skriva en metod i en lämplig klass som heter correct (eller liknande) och som tar emot minst två parametrar:

1. En HTML-collection (eller liknande, array, NodeList etc) som innehåller de svar användaren har kryssat i.
2. En array, ett objekt eller liknande , som innehåller de korrekta svaren.

Metoden ska kontrollera om användaren har svarat rätt på frågan. Om flera alternativ kan vara rätt måste användaren ha kryssat i alla korrekta alternativ för att den ska räknas som rätt.

* Du ska visa en fråga i taget och låta användaren bläddra mellan dem. Det kan t ex ske genom att byta ut elementen som innehåller frågan, eller elementens innehåll.
* Visa vilken fråga användaren är på. (T ex 3 av 10.)
* Du ska använda minst en lambda-funktion.
* Du ska lämna in i tid, dvs om du får en restuppgift kommer du inte att kunna få VG, enbart IG/G.

Redovisning sker genom ett kort, individuellt möte den 6/7 oktober, troligen på Zoom. Återkommer senare med ett schema där man får boka sin tid. Du ska också lämna in ett publikt github-repo.  
  
För att alla ska få lika mycket tid på sig oavsett när man redovisar muntligt kommer ni att få lov att lämna in ert repo fram till onsdag den 7/10. Ni måste dock kunna visa upp ett någorlunda fungerande spel på den muntliga redovisningen, tiden efter redovisningen är bara tänkt att användas för finjusteringar, mindre buggfixar osv.

### Status för inlämning

|  |  |
| --- | --- |
| **Status för inlämning** | Inga försök |
| **Betygssättningsstatus** | Ej betygssatt |
| **Stoppdatum/tid** | onsdag, 7 oktober 2020, 23:55 |
| **Återstående tid** | 6 dagar 8 timmar |
| **Senast ändrad** | - |
| **Inlämningskommentarer** | [Kommentarer (0)](https://studentportal.nackademin.se/mod/assign/view.php?id=94260) |