Eksamen A

I fagene

Programmering og Teknik

2023, juni, kl. 9.00 – 13.00 (14.30)

Alle skriftlige materialer, pc'er, bærbare computere og internet ressourcer er tilladt til eksamen.

Mobiltelefoner og kommunikation med andre personer (inkl. chatbots), bortset fra kommunikation med eksaminator og censor, er forbudt. Dog er det tilladt at bruge GitHub Copilot

Du må ikke gemme dine løsninger på eksterne netværksdrev/værter som GitHub, Facebook o. lign. Overtrædelse af dette vil medføre bortvisning fra eksamen og passende sanktioner vil senere blive tildelt både til afsender/uploader og modtager.

Ved eksamens afslutning skal du lægge din løsning op i Wiseflow.

Eksamen varer 4 timer og efterfølges af 1 times evaluering. I disse tidsrum kan du kun forlade eksamenslokalet med henblik på toilet besøg. Rygning er ikke tilladt.

Ud over disse opgaver, kan du kan blive bedt at besvare nogle ekstra spørgsmål omkring dine svar og eventuelle andre valg.

Kontroller at dette sæt indeholder 3 sider inklusive forsiden.

# Introduktion

I denne opgave skal du arbejde med validerings-metoder, unit test, REST API og web-applikation.

Du skal lave en applikation, der kan holde styr på film, jeg vil se i fremtiden.

|  |  |  |
| --- | --- | --- |
| **Id** | **Title** | **ReleaseYear** |
| 1 | Snehvide | 1937 |
| 2 | Citizen Kane | 1941 |
| 3 | Ternet Ninja | 2018 |
| 4 | Italiensk for begyndere | 2000 |
| 5 | Journal 18 | 2018 |

# Opgave 1 Model-klasse

Du skal lave et projekt af typen ”Class Library”. Navnet på projektet skal være MovieLib.

I projektet skal der være en klasse med navnet Movie.

Movie klassen skal have følgende properties (bemærk *constraints*)

* Id, et tal
* Title, tekst-streng, mindst 1 tegn
* ReleaseYear, et tal, ReleaseYear >= 1895

Movie klassen skal override ToString metoden.

Der skal være valideringsmetoder til at checke ovenstående *constraints*.   
Valideringsmetoderne skal kaste passende *exceptions* hvis *constraints* ikke overholdes.

Husk at *builde* dit projekt.

Opgave 2 Unit test

Du skal lave en unit test af din Movie klasse.   
Du skal teste dine valideringsmetoder.

Testen skal have et godt ”Code Coverage”

# Opgave 3 Repository-klasse

1. Opret en repository klasse. Repository klassen skal bruge en liste til opbevare Movie objekter.  
   Du skal indsætte mindst 5 forskellige Movie objekter i listen.  
   Movie.id skal tildeles af repository klassen.
2. Repository klassen skal have følgende metoder  
   - GetAll(), returnerer alle Movie objekterne  
   - GetById(id), returnerer Movie objektet med det angivne id  
   - Add(movie), tilføjer et nyt Movie objekt til listen  
   - Remove(id), fjerner Movie objekt med det angivne id fra listen

# Opgave 4 REST API

Du skal lavet et REST API, der kan håndtere Movie objekter.

1. Lav et nyt projekt af typen ”ASP.NET Core Web API”.  
   Husk at vælge API og ikke Web Application, når du laver projektet.  
   Sørg for at “Configure for HTTPS” er *unchecked* (du skal ikke bruge HTTPS)  
   Tilføj en reference til dit *class library*.
2. Lav en controller klasse, MoviesController.  
   Metoderne i controller klassen skal kalde metoderne i repository klassen.
3. Controller klassens metoder skal bruge relevante HTTP metoder og status-koder.
4. Afprøv dit REST API med Swagger

# Opgave 5 CORS og Azure

1. Du skal tilføje CORS til dit REST API.
2. Forklar hvilke dele af en applikation, der har brug for CORS
3. Du skal publicere din REST API til Azure.
4. Forklar om fordele ved at have REST API i Azure i forhold til at have det på *localhost*.

# Opgave 5 Test af REST API

I Postman skal du lave en test af dit REST API.

Du behøver kun teste en af metoderne i dit REST API.

# Opgave 6 Web-applikation

Du skal lave en web-applikation med HTML, JavaScript, Vue.js, Axios og Bootstrap.

Din web-applikation skal kalde metoderne fra dit REST API.

Brugeren af web-applikationen skal kunne

1. Se en tabel med alle Movie objekterne
2. Vise et Movie objekt med et angivet Id
3. Tilføje et nyt Movie objekt
4. Slette et Movie objekt

Brug Bootstrap til at *style* din web-applikation

# Opgave 7 Web-applikation, flere features

Brugeren af din web-applikation skal have mulighed for at sortere og filtrere Movie objekter.

Nævn fordele og ulemper ved at lave sortering og filtrering foregå på front-end henholdsvis back-end.

# Opgave 8 REST repository med database

Lav en database-tabel med kolonner svarende til properties i din model-klasse.

Du skal lave en ny repository klasse, der bruger din database-tabel til at opbevare data.