Re-Eksamen

I fagene

Programmering og Teknik

2023, august, kl. 9.00 – 13.00 (14.00)

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 klassebibliotek, Unit-test, REST API, web-applikation og Python.

Du skal lave et system der kan holde styr på forskellige slags terninger, der kan have få eller mange sider.

# Opgave 1 Model-klasse

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

I projektet skal der være en klasse med navnet Dice.

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

* Id, et tal
* SideAmount, et tal, mellem 4 og 20
* Color, en string, minimum 2 tegn

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 Unit-tests af din Dice klasse.

Du skal teste dine valideringsmetoder.

Testen skal have et godt ”Code Coverage”

# Opgav 3 Repository-klasse

Tilføj til dit class library

1. Opret en *repository*-klasse. *Repository*-klassen skal bruge en liste til opbevare Dice objekter.   
   Du skal indsætte mindst 5 forskellige Dice objekter i listen.   
   Dice.id skal tildeles af *manager*-klassen.
2. Repository-klassen skal have følgende metoder   
   - GetAll(), returnerer alle Dice objekterne   
   - GetById(id), returnerer Dice objektet med det angivne id   
   - Add(dice), tilføjer et nyt Dice objekt til listen   
   - Update(id, dice), opdaterer et Dice objekt med det angivne id, med parameter Dice objektets værdier

# Opgave 4 REST API

Du skal lavet et REST API, der kan håndtere Dice 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, DiceController.   
   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 Teori Security

I opgave 3 blev du bedt om at fravælge HTTPS. Du skal besvare følgende spørgsmål, og du skal skrive nogle noter omkring svarene:

1. Hvad betyder HTTPS?
2. Hvad er forskellen på symmetrisk og asymmetrisk kryptering?
3. Bruger HTTPS symmetrisk eller asymmetrisk kryptering (eller ingen eller begge)?

# Opgave 6 CORS

1. Du skal tilføje CORS til din REST service. Dette skal gøres lokalt (ikke i Azure)
2. Hvad betyder CORS? Hvordan kan du se om CORS er tilføjet?

# Opgave 7 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 Dice objekterne
2. Vise et Dice objekt med et angivet Id
3. Tilføje et nyt Dice objekt
4. Opdatere et Dice objekt

Brug Bootstrap til at *style* din web-applikation

# Opgave 8 Python Sense HAT emulator

I denne opgave skal du lave et Python program, der skal simulere, at du slår med 2 almindelige 6-sidede terninger.

Programmet skal køre indtil de 2 terninger giver det samme resultat, det vil sige at programmet skal blive ved med at ”slå med terningerne” indtil de er ens.

Det skal laves i Sense HAT emulatoren på følgende link:

<https://trinket.io/sense-hat>

Sense HAT modulet skal bruges til at vise hvor mange slag du har brugt, og hvilket tal de 2 terninger viser.