

Prøve i programmering

9/12 2019

Dette er en prøve i de ting vi har arbejdet med i programmering indtil nu. Materialerne til prøven er dette dokument, samt de programmer der ligger i portfolio sammen med.

Der er tre opgaver i alt.

Når modulet er slut skal du aflevere din besvarelse i portfolio. Du skal aflevere følgende:

- En samlet .zip-fil, der indeholder:
 - Et .pdf-dokument med forklaringer til opgaverne. Der står tydeligt i opgaveteksten hvad du skal forklare.
 - De python-filer du har rettet i eller skrevet selv, i løbet af prøven.

Du skal forsøge at løse så mange af opgaverne som muligt, men ikke gå i panik, hvis du ikke kan nå at løse dem alle. Jeg har forsøgt at lave opgaver nok til, at alle har noget at lave hele modulet. Hvis du ikke kan løse en opgave, er det også helt fint at aflevere delvise løsninger, kode der ikke virker helt, eller bare en forklaring på hvad du har tænkt. De fleste delspørgsmål kan løses uafhængigt af de andre.

Du bliver bedømt på de elementer du bruger i dine løsninger, fagsproget i dine forklaringer og selvfølgelig hvorvidt du har løst opgaverne.

Du må gerne gå på internettet under prøven, hvis du skal søge dokumentation eller finde en god idé til koden.

Du må ikke kommunikere med andre, eller udveksle filer på nogen som helst måde.

Opgave 1. Denne opgave handler om en hjemmeside til at lave tilfældige tal.

- Start programmet `tilfældighed.py`, og besøg siden <http://127.0.0.1:5000/>. Du skulle gerne se en simpel side med en overskrift og et tilfældigt tal.
- Lige nu producerer siden et tilfældigt tal mellem 0 og 1. Det kan man ikke bruge til så meget. Lav om i koden til programmet, så det tilfældige tal er en simulering af et terningkast med en 6-sidet terning. Det vil sige, at det tilfældige tal skal have en af værdierne 1,2,3,4,5 eller 6, med lige stor sandsynlighed for hver.
- Beskriv din løsning i dokumentet med forklaringer.

Opgave 2. Ejeren af tilfældighedshjemmesiden vil gerne sikre sig, at hans tilfældige tal er tilfældige nok. Han har fået den idé, at han gerne vil gemme alle terningekast i en database. Der er gjort klar til databasen i koden, og din opgave er, at lave løsningen færdig.

- Undersøg funktionen `create_db_table()`, der opretter en tabel i databasen.
- Skriv en funktion der hedder `gem()`, som tager et terningekast (en `int` med værdi mellem 1 og 6) som argument, og indsætter værdien i databasen.
- Skriv en funktion der hedder `antal()`, der returnerer en liste med 6 elementer. Listen skal indeholde antallet af gange hvert terningekast er forekommet.
- Skriv en funktion der hedder `gennemsnit()`, der læser alle terningekast i databasen, og returnerer gennemsnittet af værdierne.
- Udvid hjemmesiden med en sti der hedder <http://127.0.0.1:5000/statistik>, hvor man kan se gennemsnittet af alle terningekastene, samt antallet af gange hvert slag er forekommet.
- Forklar grundigt om din løsning i dokumentet med forklaringer. Brug figurer, diagrammer og kodeeksempler, der hvor det er relevant.

Opgave 3. I filen `yatzy.py` kan du finde en klasse, der hedder `Terning`. Klassen indeholder en funktion der hedder `roll()`, der printer et terningekast, samt en variabel `n`, der afgør hvor mange sider der er på terningen.

- Lav funktionen `roll` om, så den laver et terningekast med `n` sider, og returnerer værdien.
- Udvid klassen med funktionen `rollMany(s)`, der foretager `s` terningekast, og returnerer en liste med de `s` tal.
- Udvid klassen med funktionen `sum(t)`, der slår med terningen `t` gange, og returnerer summen af terningekastene.
- Udvid koden i bunden af filen med et program, der simulerer et spil yatzy.
- Beskriv din løsning grundigt i dokumentet med forklaringer. Brug masser af figurer, diagrammer, kodeeksempler, og demonstrer hvordan programmet fungerer.