

Eksamenssæt HA(it.) - 1. år

Programmering og udvikling af små systemer samt databaser

Skriftlig eksamen: Efterår 2020

Varighed: 3 timer

Hjælpemidler:

- Alle hjælpemidler er tilladt

Aflevering:

- Hver af programmeringsopgaverne skal besvares i en tekstfil, hvori I vedlægger jeres kode samt evt. kommentarer.
- Alle tre filer samles til en PDF-fil og afleveres samlet. Vejledning til oprettelse af samlet pdf- fil er vedlagt som bilag i Digital Eksamen.

Indhold:

1. Programmeringsopgave 1 (besvares i tekstfil). Vægter 30%.
2. Programmeringsopgave 2 (besvares i tekstfil). Vægter 30%.
3. Programmeringsopgave 3 (besvares i tekstfil). Vægter 40%.

Programmeringsopgave 1 (30%)

Du skal skrive et program, som kan kontrollere om en streng er et palindrom. Programmet skal indeholde en **rekursiv** funktion, som løber selve strengen igennem, men du må gerne have flere funktioner i programmet.

Et palindrom er et ord, som skrives på samme måde både forfra og bagfra. Eksempler herpå kunne være:

- Regninger
- Kajak
- Rejer
- Bøvs i svøb

Programmet skal være skrevet i Node.JS og kunne kaldes fra command-line. Det skal herudover kunne initieres med en parameter, som er det ord vi skal kontrollere. Såfremt ordet er et palindrom skal funktionen returnere true, i alle andre tilfælde hvor ordet ikke er et palindrom skal funktionen returnere false.

Du forventes at have overholdt god kodeskik igennem opgaven, hvorfor du bedes indsætte kommentarer i din kode og generelt overholde en god struktur i din kode.

Du vil blive bedømt på om

- Du har løst opgaven funktionelt
- Overholdt god kodestruktur
- Kommenteret din kode
- Din kode kan køres og giver det rigtige resultat
- Performance af din løsning

Programmeringsopgave 2 (30%)

Du skal udarbejde et program, som kan fjerne dubletter og samtidig sortere et array bestående af en række tal-elementer. Du skal antage, at array'et kan indeholde flere millioner elementer, hvorfor performance er vigtigt. Du skal tage udgangspunkt i nedenstående array, men det er vigtigt, at du holder performance for øje såfremt vi tilføjede flere elementer. Dit program skal derfor virke uanset hvilket array det modtager som start.

```
var array = [1180, 3, 5, 9, 4, 5, 11, 9, 3, 1, 1, 3, 120, 4, 5, 7, ...];
```

Skal blive til:

```
var result = [1, 3, 4, 5, 7, 9, 11, 120, 1180];
```

Programmet skal være skrevet i Node.JS og kunne kaldes fra command-line. Du må **IKKE** benytte indbyggede sorteringsfunktioner, men skal skrive jeres egen. Du skal angive køretid givet som O-notation i en docstring i dit program. Programmet skal indeholde en unit test som tester for om arrayet har dubletter.

Du forventes at have overholdt god kodeskik igennem opgaven, hvorfor du bedes indsætte kommentarer i din kode og generelt overholde en god struktur i din kode.

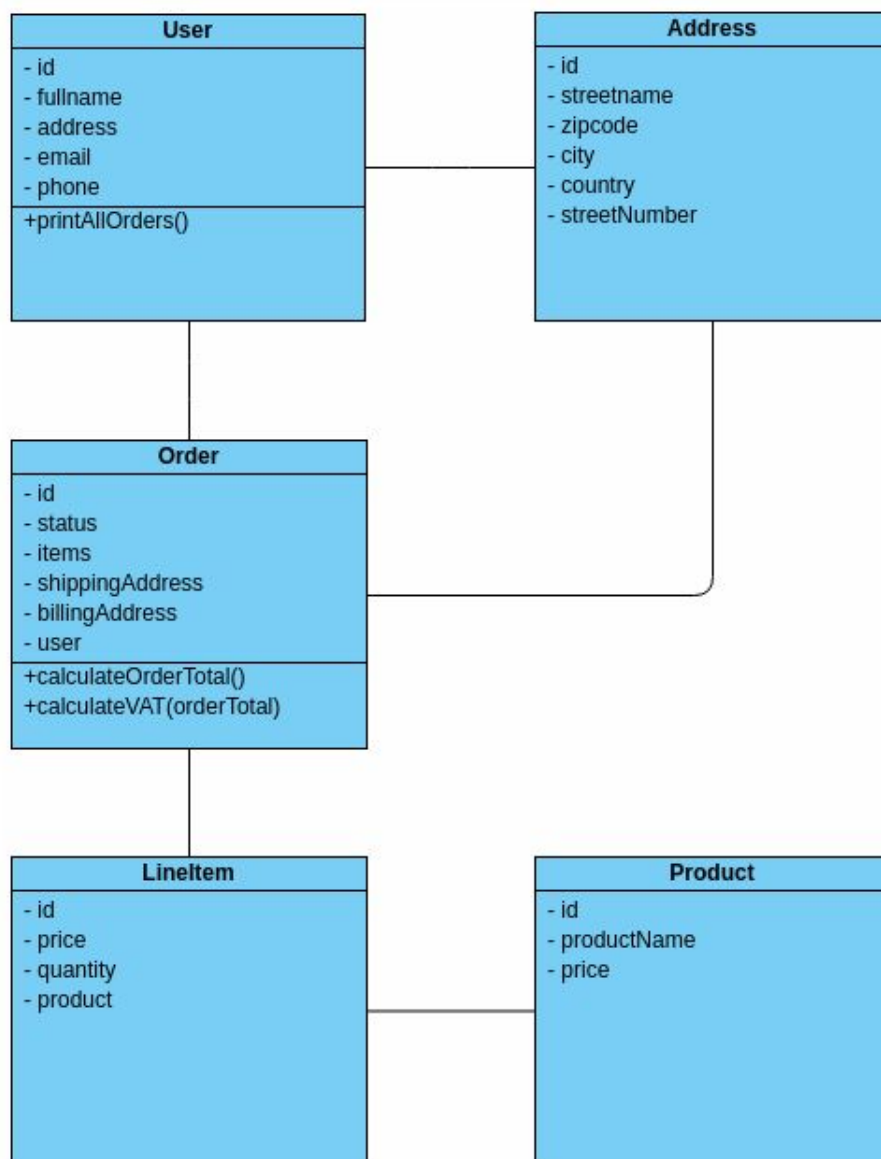
Du vil blive bedømt på om

- Du har løst opgaven funktionelt
- Overholdt god kodestruktur
- Kommenteret din kode
- Din kode kan køres og giver det rigtige resultat
- Performance af din løsning

Programmeringsopgave 3 (40%)

Du skal skrive et program til en webshop, som kan bruges til de mest fundamentale funktioner i en webshop, men uden at have et interface af nogen art. Programmet skal følge programmeringsvejledningen beskrevet sidst i denne opgave, og skal til sidst kunne printe købshistorik og total ordre for specifikke brugere og deres ordre.

Programmet skal indeholde følgende klasser: Product, Address, LineItem, Order, User.



Følgende krav skal opfyldes:

- **Product-klassen** skal indeholde tre attributter;

- id (Number)
- productName (String)
- price (Number)

I constructoren skal alle attributter oprettes.

- **Address-klassen** skal indeholde seks attributter;

- id (Number)
- streetName (String)
- zipCode (String)
- city (String)
- country (String)
- streetNumber (String)

I constructoren skal alle fem attributter oprettes.

- **LineItem-klassen** skal indeholde fire attributter;

- id (Number)
- price (Number)
- quantity (Number)
- product (Product)

I constructoren skal alle attributter oprettes. Attributten product skal være et objekt af klassen Product.

- **Order-klassen** skal indeholde seks attributter:

- id (Number)
- status (String)
- items (Array)
- shippingAddress (Address)
- billingAddress (Address)
- user (User)

I constructoren skal alle attributter oprettes. ShippingAddress og billingAddress skal være af typen Address og derfor være et objekt af klassen Address. User skal være af typen User og derfor være et objekt af klassen User.

Klassen skal indeholde to metoder, som returnere et Number:

- **calculateOrderTotal()** skal gå alle items igennem og summere prisen for disse.
- **calculateVAT(orderTotal)** skal tillægge moms til prisen.. Denne metode tager en parameter orderTotal og tilføjet denne parameter 25% i moms, inden den returnere momsbeløbet.

- **User-klassen** skal indeholde fem attributter;

- id (Number)
- fullname (String)
- address (Address)
- email (String)
- phone (String)

I constructoren skal id, fullname, address og email oprettes. Address skal være af typen Address, hvorfor denne attribut skal være et objekt af klassen Address.

Klassen skal indeholde en metode:

- **printAllOrders()** som skal udskrive alle ordre, som den pågældende bruger har oprettet på et givent tidspunkt. Metoden skal for hver ordre udskrive ID og den totale pris for ordren, inklusiv moms.

Programmeringsvejledning:

Det er vigtigt, at du benytter de samme navne, som er vist i teksten og i klassediagrammet. Husk at indsætte relevante kommentarer i din kode, samt at overholde god kode-skik generelt.

1. **LAV KLASSENE** | Skriv først koden til hver af klasserne og vær særlig opmærksom på klassernes indbyrdes relationer. Du skal ikke lave validering af typer eller værdier i denne opgave.

2. **OPRET PRODUKTER** | Når koden til alle klasserne er lavet opretter du fire produkter;

- iPhone 11 (12.500)
- iPhone X (10.000)
- iPhone 8 (7.500)
- iPhone SE (5.000)

Du kan enten lave fire variable til disse produkter, eller placere objekterne i et array.

3. **OPRET BRUGERE** | Herefter opretter du to brugere med tilhørende adresse.

- a. Anders And, Paradisæblevej 8, 2860 Andeby, Danmark. lequack@disney.com, 4026 39 78.
- b. Joakim Von And, Store Velstands Boulevard 888, 2860 Andeby, Danmark. scrooge@disney.com, 3815 38 15.

4. **FORETAG KØB** | Du kan nu foretage et køb ved at oprette de korrekte objekter af dine klasser.

Anders And skal købe følgende til Rip, Rap og Rup:

- 2 x iPhone 11

Joakim Von And skal købe iPhones til hele firmaet:

- 2 x iPhone X
- 1 x iPhone 8
- 6 x iPhone SE

Anders And glemte at købe en telefon til Rap og laver derfor en order mere, som indeholder:

- 1 x iPhone X

Ved oprettelse af lineltems skal prisen hentes fra produktet.

5. UDSKRIV RESULTAT | Når de korrekte objekter er oprettet udskriver du følgende med brug af `console.log()`:

1. En liste af Anders And's ordre, ved at kalde metode `printAllOrders()` på Anders' objekt.
2. `OrderTotal()` for Joakim Von And, hvilket skal inkludere moms.

Du forventes at have overholdt god kodeskik igennem opgaven, hvorfor du bedes indsætte kommentarer i din kode og generelt overholde en god struktur i din kode.

Du vil blive bedømt på om

- Du har løst opgaven funktionelt
- Overholdt god kodestruktur
- Kommenteret din kode
- Din kode kan køres og giver det rigtige resultat
- Hvor mange af de 5 trin, du har løst korrekt.