

Eksamensopgave

Objektorienteret programmering i C++ og Softwareudvikling

RB-PSW2 – T540003101

3. juni 2020

Alle hjælpemidler er tilladt. Husk at de generelle regler om snyd stadig gælder, og at snyd medfører bortvisning.

Aflevering

Der skal afleveres én samlet zip-fil med al jeres kildekode fra C++, inklusive testdriver(e) og én pdf-fil med besvarelserne til softwareudviklingsopgaverne. Altså kildekode og pdf i én zip-fil. Der kan kun afleveres én fil.

Opgavesættet indeholder seks sider inklusive forside.

Opgave 1 – programmering i C++ (anslået tidsforbrug 80 minutter)

Denne opgave omhandler et antal samarbejdende klasser til håndtering af informationer til et bilværksted.

Du skal implementere fire klasser:

`Vaerksted` – overordnet klasse, som bl.a. indeholder containere af de øvrige klasser.

`Bil` – med informationer om kundernes biler.

`Mekaniker` – med informationer om værkstedets mekanikere.

`ArbejdsSeddel` – med data om arbejde, som mekanikerne har udført på bilerne.

Klassen `Vaerksted`

Nedenfor er vist et uddrag af et forslag til headerfil:

```
class Vaerksted
{
public:
    Vaerksted();
    Vaerksted(string adresse, string ejer);
    string getAdresse();
    string getEjer();
    void addBil(Bil &enB);
    void addMekaniker(Mekaniker &enM);
    void addArbejdsSeddel(ArbejdsSeddel &enA);
    ~Vaerksted();

private:
    string adressen;
    string ejeren;
    // Her tilføjes containere og evt. attributter til at holde styr
    // på antallet af elementer i containerne
    // biler – fx en vector eller et array med objekter af klassen Bil
    // mekanikere – fx en vector eller et array med objekter af klassen Mekaniker
    // arbejdsSedler – fx en vector eller et array med objekter af klassen
    // ArbejdsSeddel
};
```

Klassen `Bil`

indeholder attributterne `regNr` og `ejer`, begge af typen `string`.

Klassen `Mekaniker`

indeholder attributten `navn` af typen `string`.

Klassen `ArbejdsSeddel`

indeholder referencer til et objekt af klassen `Bil` og til et objekt af klassen `Mekaniker`. Derudover attributterne `datoen` og `timerne`, som angiver datoen for arbejdets udførelse og antal anvendte timer. Begge er af typen `int`.

Når du har implementeret klasserne, skal nedenstående testdriver kunne køres uden fejl.

```
int main()
{
    Vaerksted vS("Nymarken 104, 5330 Munkebo", "Lars Peter Hansen");

    Bil b1("MA39604", "Knud Pedersen");
    Bil b2("MH40136", "Lis Fugl");
    Bil b3("MA45647", "Herluf Jensen");
    vS.addBil(b1);
    vS.addBil(b2);
    vS.addBil(b3);

    Mekaniker m1("Poul");
    Mekaniker m2("Per");
    vS.addMekaniker(m1);
    vS.addMekaniker(m2);

    ArbejdsSeddel a1(b1, m2, 7, 20200503);
    ArbejdsSeddel a2(b3, m1, 4, 20200512);
    ArbejdsSeddel a3(b3, m2, 3, 20200514);
    ArbejdsSeddel a4(b1, m1, 5, 20200516);
    ArbejdsSeddel a5(b3, m2, 2, 20200518);
    vS.addArbejdsSeddel(a1); // Det antages, at arbejdssedler
    vS.addArbejdsSeddel(a2); // registreres i kronologisk orden
    vS.addArbejdsSeddel(a3);
    vS.addArbejdsSeddel(a4);
    vS.addArbejdsSeddel(a5);
}
```

Du skal nu implementere følgende tre metoder i Vaerksted-klassen

1

```
void Vaerksted::ingenRegning()
```

Metoden skriver (på `cout`) navnene på de bilejere, som ingen regning skal have på baggrund af de udfyldte arbejdssedler. I dette tilfælde *Lis Fugl*.

2

```
void Vaerksted::mekanikerTimer()
```

Metoden udskriver en liste på `cout` med navnene på mekanikerne og det samlede antal timer, de hver især har arbejdet ifølge arbejdssedlerne. En linje per mekaniker.

3

```
string Vaerksted::senesteArbejde()
```

Metoden returnerer en `string` indeholdende datoen for det senest udførte arbejde på formen DD/MM/YYYY. I dette tilfælde "18/05/2020".

Opgave 2 – programmering i C++ (anslået tidsforbrug 20 minutter)

Skriv en metode med følgende signatur:

```
bool ordITekst(string tekst, string ord)
```

Metoden returnerer `true`, hvis parameteren `ord` er indeholdt i parameteren `tekst`. Hvis ikke, returneres `false`.

Opgave 3 – programmering i C++ (anslået tidsforbrug 20 minutter)

Skriv en metode med følgende signatur:

```
int largestSumOfTwo(int ar1[], int ar2[], int size)
```

Metoden skal returnere summen af de to største tal i begge arrays. Kaldt med de to nedenstående arrays og med `size = 5`, returneres 110.

```
int ar1[] = { 17,33,44,11,9 };  
int ar2[] = { 22,66,1,35,22 };
```

Opgave 4 - Softwareudvikling (anslået tidsforbrug 60 minutter)

Denne opgave går ud på at udarbejde en **domænemodel** til en 1. iteration af et it-system, som kan bruges til styring af softwareudviklingsprojekter, hvor der anvendes *Unified Process* og *Scrum*.

Et projekt består således af fire faser (inception, elaboration, konstruktion og deployment) og hver fase består af et antal iterationer. Der opereres med to typer af backlogs, den store, som omfatter alle projektets aktiviteter, og sprint backlogs.

Systemet skal kunne opfylde følgende krav:

- Oprettelse af ovennævnte infrastruktur samt projektmedarbejdere og projektleder.
- Registrering af tidsforbrug per medarbejder per aktivitet. Medarbejderne skal således lave almindelig tidsregistrering, hvor de ved dagens slutning indrapporterer hvor mange timer de har arbejdet på hvilke aktiviteter i dagens løb.
- Opfølgning på projektplan. Det er ambitionen, at systemet automatisk skal kunne generere en *sprint burndown* på daglig basis.

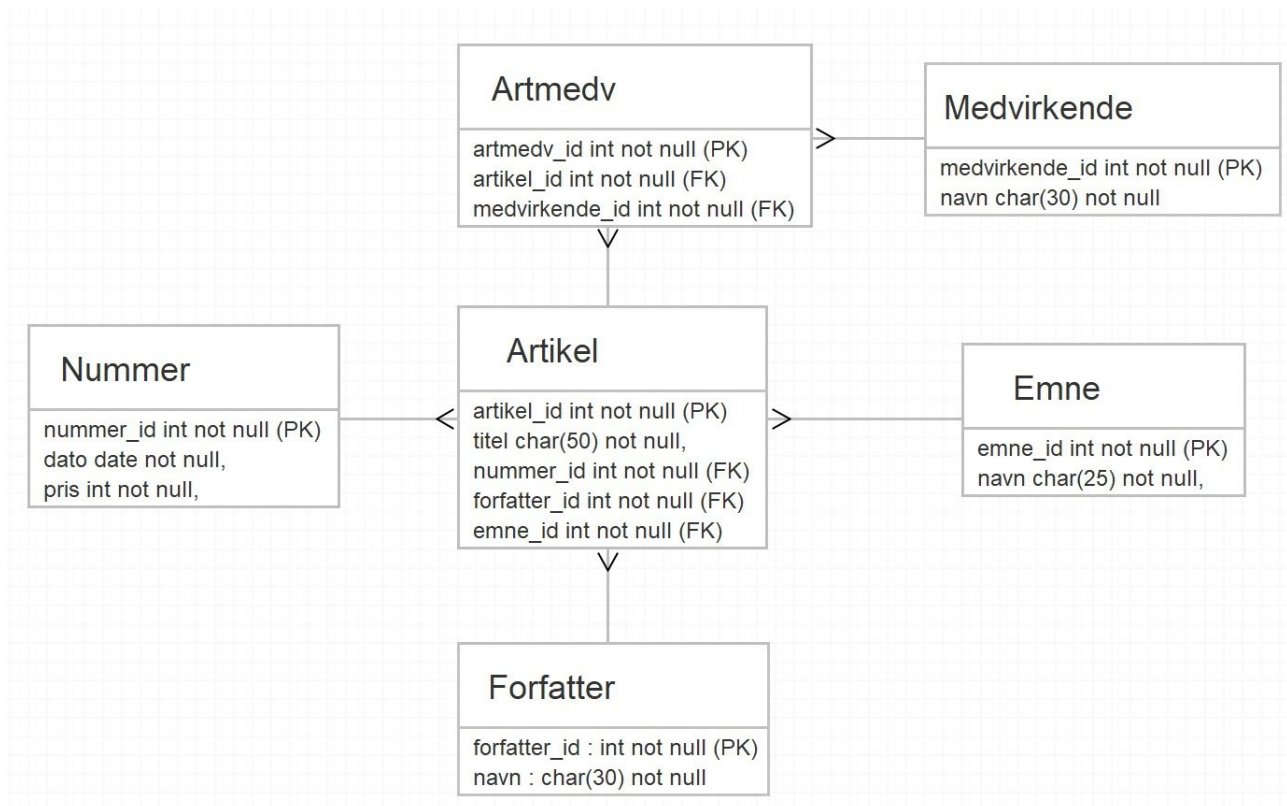
Du bestemmer selv, hvilke 'hjelpeartefakter' (brugsmønstre, sekvensdiagrammer, kontrakter), du vil lave. Husk at et artefakt skal kun udarbejdes, hvis den skaber værdi for systemet.

Det er imidlertid meget vigtigt, at du argumenterer for de trufne valg. Den gode besvarelse indeholder således relevante kommentarer til de indgående elementer (klasser, attributter og relationer mellem klasser).

Ikke alle kommentarer er lige relevante. Relevante kommentarer skal, i lighed med 'hjelpeartefakterne', skabe værdi enten i form af bedre forståelse for rationalet bag de trufne valg eller ved at forøge kendskabet til domænet.

Opgave 5 - Softwareudvikling (anslået tidsforbrug 60 minutter)

Opgaven tager udgangspunkt i nedenstående database¹.



Der er tale om en database til håndtering af artikler i et tidsskrift, hvor hvert nummer i dette eksempel indeholder tre artikler. Artiklerne vedrører hver især et *emne*, de er skrevet af én *forfatter* og har et antal *medvirkende* personer.

Databasen kan etableres i MySQL ved hjælp af tekstfil med navnet *TidsskriftsstyringDDL*

For at opnå maksimumpoints i denne opgave skal dine besvarelser kunne afvikles fejlfrit på denne database; men du skal blot aflevere SQL-kaldene i én pdf-fil sammen med besvarelsen til opgave 1.

Spørgsmål 1

Skriv titel på artikel og navn på emne af artikler forfattet af Michael Christiansen.

Spørgsmål 2

Vis titlerne på de artikler, som ikke har medvirkende.

Spørgsmål 3

Udskriv nummer_id og pris på numre, som endnu ikke er udkomne. Aktuel dato må IKKE hardkodes, dvs. du må ikke skrive fx *dato = '2020-08-01'*.

Spørgsmål 4

Udskriv titlerne på de artikler, som har mere end en medvirkende.

¹ Der er tale om et ER-diagram (entity-relationship), som ikke må forveksles med et UML klassediagram. I et ER-diagram modelleres en 1:M relation (fx ét emne kan indgå i mange artikler), således at forgreningen af den ene ende af forbindelsen mellem tabellerne vender ind imod den tabel, hvor fremmednøglen er.