RB-PSW2

Eksamen i Softwareudvikling og Objektorienteret programmering

Eksamensdato: 1. juni 2021

Kursusansvarlige: Ole Dolriis og Thorbjørn Mosekjær Iversen

Vigtig information vedrørende eksamen:

Alle hjælpemidler er tilladt. Dette inkluderer brug af internettet. Det er dog ikke tilladt at kommunikere med andre under eksamen. Husk at de generelle regler om snyd stadig gælder.

Aflevering

- Der skal til Softwareudviklingsdelen afleveres én pdf-fil med besvarelsen. Husk: *ingen* screenshots til SQL-delen.
- Der skal til C++-delen afleveres én zip-fil indeholdende:
    - alle header- og cpp-filer
    - en readme.txt der angiver hvilke filer der hører til hvilke opgaver. Hvis I har nogen generelle kommentarer til jeres besvarelse, kan dette også skrives i readme.txt filen.
    - screenshots der viser at I kan compilere hver af de 3 opgaver samt outputtet af disse. Hvis ikke jeres kode compiler tager I i stedet et screenshot af fejlmeddelelsen fra compileren

Generelt vurderes jeres besvarelse til C++ delen på, I hvilket omfang I er i stand til at løse opgaverne. Enhver form for kopier/indsæt (copy/paste) fra tidligere opgaver eller andre kilder anses som eksamenssnyd. Kode fra denne eksamensopgave må dog gerne kopieres ind.

Opgavesættet består af 5 sider, 1 forside, 2 sider med C++ opgaver og 2 sider med softwareudvikling

Opgave 1 - programmering i C++ (anslået tidsforbrug 60 minutter)

Lone Hansen er globetrotter og kunstentusiast og rejser verden rundt for at købe og sælge kunstværker. Hun ender derfor med en masse kunst og pengebundter med fremmed valuta. Hun ønsker at få udviklet et program der kan udregne den samlede værdi af sine kunstværker og sin valuta udregnet til danske kroner. I denne opgave skal du programmere dette system.

Systemet skal indeholde nedenstående main-fil og følgende klasser: `AllValuables`, `Valuable`, `StackOfMoney`, `ExchangeRate`. Klasserne er beskrevet herunder. Du skal som minimum implementere de nævnte metoder og funktionalitet. Desuden skal main metoden kunne køres så den giver det output der er angivet i slutningen af opgaven. Du må ikke tilføje flere klasser end de ovennævnte. Det er dog tilladt at tilføje yderligere hjælpefunktioner og variable.

Valuable er en klasse der indeholder metoderne `getValue()` og `getName()`. Den repræsenterer både et kunstværk (F.eks. name="Monika Lisa", value=30000) og er superklasse til StackOfMoney (F.eks. name="From mom", value=1000)

ExchangeRate indholder navnet på en type valuta og kursen, dvs. prisen i dkk svarende til 100 af den fremmede valuta (F.eks. er 100USD=631kr). Det skal desuden være muligt at ændre kursen med et kald til `void setExchangeRate(double rate)`.

StackOfMoney repræsenterer et bundt pengesedler i en fremmed valuta. StackOfMoney nedarver fra Valuable og overskriver metoden getValue() således at der returneres værdien af pengesedlerne i danske kroner. For at kunne omregne fra den fremmede valuta til danske kroner afhænger klassen af en instans af ExchangeRate.

AllValuables indholder en std::vector<Valuable*> som er en vector af pointers til alle kunstværker og pengeseddelsbundter. Det er AllValuables som står for at summere over værdien af alle kunstværker og pengebundter. Der skal desuden kunne printes en liste over alle kunstværker og pengebundter og værdien af disse (Se outputet fra main filen i slutningen af opgaven). Vær opmærksom på at AllValuables kun må kende til Valuable klassen og altså *ikke* hverken ExchangeRate eller StackOfMoney klasserne.

Du skal desuden være opmærksom på at systemet skal programmeres så AllValuables udskriver den rigtige værdi selvom kursen ændrer sig efter alle kunstværker og pengebundter er tilføjet (Se testkode i slutningen af opgaven).

Main-fil

```
#include <iostream>
#include "valuable.h"
#include "exchangeRate.h"
#include "stackOfMoney.h"
#include "allValuables.h"
```

```
int main(int argc, char** argv){
    Valuable mona("Monika Lisa", 30000);
    Valuable venus("Venus de Milano", 20000);
    Valuable scream("The Screamer", 23000);
    ExchangeRate dollar("USD", 631);
    ExchangeRate euro("EUR", 768);
    StackOfMoney whistler("Payment Whistler's Father", 5000, dollar);
    StackOfMoney guernica("Payment Guernicalala", 3000, euro);
    StackOfMoney mom("From mom", 1000, euro);
    AllValuables all;
    all.add(mona);
    all.add(venus);
    all.add(scream);
    all.add(whistler);all.add(guernica);
    all.add(mom);
    std::cout << all.getValue() << std::endl; //135270 (original Euro kurs)
    euro.setExchangeRate(759);
    std::cout << all.getValue() << std::endl; //134910 (ny Euro kurs)
    all.printAll();
}
```

output:
135270
134910
Monika Lisa, 30000
Venus de Milano, 20000
The Screamer, 23000
Payment Whistler's Father, 31550
Payment Guernicalala, 22770
From mom, 7590
Total: 134910

Opgave 2 - programmering i C++ (anslået tidsforbrug 30 minutter)

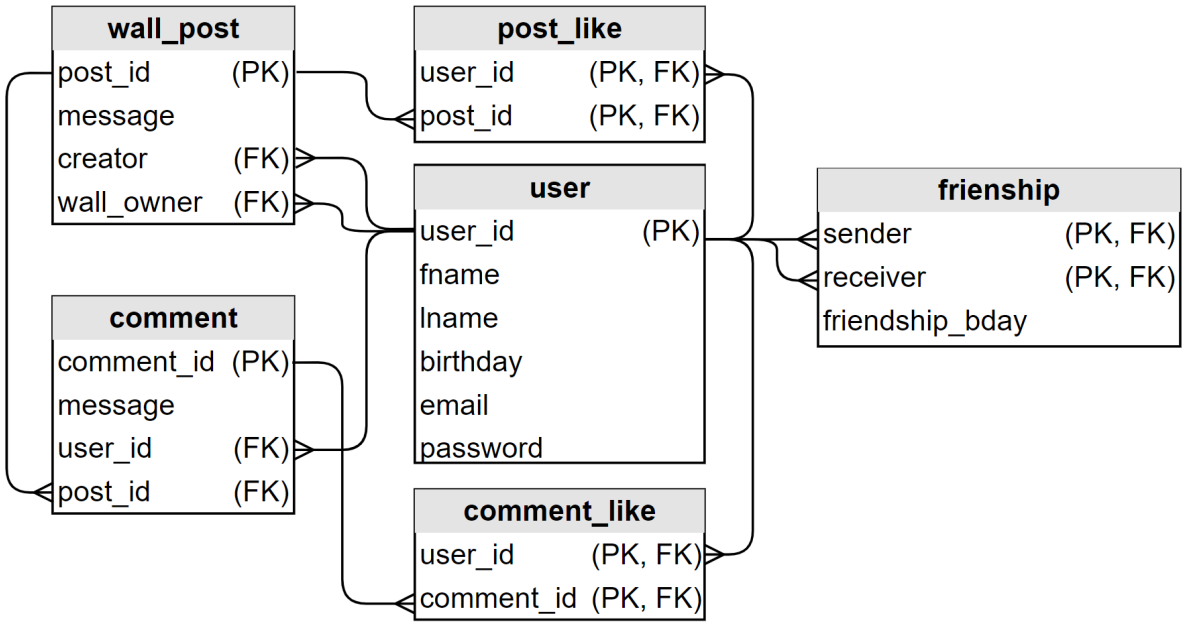
En Pythagoras-triplet er tre heltal a, b og c for hvilket det gælder at $a^2 + b^2 = c^2$. Skriv en funktion der tager en std::vector<int> som input og returnerer en std::vector<int> hvor alle Pythagoras-triplets er fjernet. Fx. vil et input bestående af {6, 25, 5, 3, 7, 24, 4, 23} returnere {6, 23} fordi $3^2 + 4^2 = 5^2$ og $7^2 + 24^2 = 25^2$ er Pyhagoras-triplets.

Opgave 3 - programmering i C++ (anslået tidsforbrug 30 minutter)

Et palindrom-ord er et ord som staves ens forfra og bagfra (fx. racecar og kayak). Skriv en funktion der som input tager en std::string der er en liste af ord sepereret med mellemrum (F.eks. "radar handle machine racecar kayak jetplane") . Funktionen skal returnere en std::vector<std::string> med de ord i input-strengen som er palindrom-ord. For input-strengen angivet ovenfor skal funktionen således returnere en std::vector<std::string> med længden 3 indeholdende ordene radar, racecar, og kayak.

Opgave 4. Databaser og SQL (anslået tidsforbrug 60 minutter)

Opgaven tager udgangspunkt i nedenstående database



Der er tale om en simplificeret udgave af Facebooks database til håndtering af brugere på platformen. En bruger kan have mange venner, lave mange posts og mange kommentarer. Friendship tabellen indeholder ID på afsender og modtager af venneanmodninger. En venneanmodning er accepteret hvis venskabsfødselsdagen har en dato og ikke er NULL. Databasen kan etableres i MySQL ved hjælp af tekstfil med navnet *FacebookDDL*.

Spørgsmål 1

Find alle kommentar tekster til posts som Kim har lavet.

Spørgsmål 2

Hvem har liket Josephines posts? Find fname og lname på alle der har liket en post af Josephine.

Spørgsmål 3

Du ved at én af brugerne har adgangskoden 'gosportgo' men du ved ikke hvem. Skriv et query der kan finde fname og lname for indehaveren af adgangskoden.

HINT: Kig i scriptet (*FacebookDDL*).

Spørgsmål 4

Lav en liste over navn på brugerne og tal på hvor mange kommentarer hver bruger har fået på sine posts, sorter listen efter hvem der har fået flest.

Spørgsmål 5

Du synes ikke det er så kønt med 2 fornavne. Hvis en bruger har 2 fornavne, så opdater brugeren ved at fjerne det sidste fornavn fra fname, og læg det til starten på lname,

HINT: For at finde det første ord i en streng brug REGEXP_SUBSTR(<STRING> , '^[a-z]+')

Opgave i domænemodellering – anslået tidsforbrug 60 minutter

En historiker/forfatter har besluttet sig for at specialisere sig i at skrive kortfattede biografier om afdøde, europæiske monarker (konger/regerende dronninger). Til det brug ønsker han/hun udviklet en it-applikation, som kan opsamle og sammenfatte relevante oplysninger om hovedpersonen.
I den første iteration af systemet skal der holdes styr på monarkens aktiviteter, fx at han/hun på en bestemt dato mødtes med et antal navngivne ministre på et bestemt sted og diskuterede et eller flere emner. Der skal tillige holdes styr på monarkens familieforhold, både hvad angår afdøde og nulevende personer.

Systemet skal som minimum kunne:

- Lave en tidslinje over monarkens aktiviteter og opholdssteder samt hvem der var til stede.
- Liste samvær med navngivne personer, fx for at opnå overblik over hvem der fortrinsvis har påvirket monarken.
- Besvare diverse forespørgsler som fx
    - Hvor længe og hvornår monarken har opholdt sig på et bestemt sted.
    - Angive opholdssted ved input af en dato.
    - Hvornår monarken har diskuteret et bestemt emne med ministre eller rådgivere.

Eksempel: systemet skal kunne registrere følgende (fiktive) aktivitet for en monark. I dette tilfælde den danske konge Frederik 3. (1609-1670).

Den 16. marts 1649 holdt kongen møde med to af sine ministre, finansminister Corfits Ulfeldt og udenrigsminister Hannibal Sehested. Ministrene var begge familiemedlemmer, idet Ulfeldt var gift med kongens halvsøster Leonora Christina og Sehested med en anden af kongens halvsøstre, Christiane. Mødet blev holdt på Københavns Slot og handlede om flådens bemanding.

Den centrale artefakt, som du skal udarbejde, er:

• Domænemodel

Og du bestemmer selv, hvilke 'hjælpeartefakter' (fx brugsmønstre, kontrakter) du vil lave. Husk at en artefakt skal kun udarbejdes, hvis den skaber værdi for systemet.

Det er vigtigt, at du argumenterer for de trufne valg i de tilfælde, hvor domænemodellen måske forekommer utilstrækkelig eller tvetydig. Den gode besvarelse indeholder således kommentarer til de væsentligste valg af de indgående elementer (klasser, attributter og relationer mellem klasser).