

Eksamensopgave i RD2-SWC

17. august 2020

Alle hjælpemidler er tilladt. Husk at de generelle regler om snyd stadig gælder.

Aflevering

Der skal afleveres én samlet PDF med opgavebesvarelsen af både C++-delen og softwareudviklingsdelen, som alene er det dokument, der bliver bedømt. I kan kun aflevere et dokument som altså både skal indeholde C++ afleveringen med kildekode og test samt softwareudviklingsdelen.

Til hver C++ opgave er der en *test-kode*, som skal *eksekveres* og et *screendump* skal indsættes i afleveringen. Hvis programmet ikke kan compile, så indsættes i stedet *fejlmeddelelsen* fra compileren.

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 8 sider, 1 forside, 5 sider med C++ opgaver og 2 sider med softwareudviklingsopgaven.

Opgave 1 (1 time, 25 point)

Klassen TimeAndMaterials repræsenterer tid og materialer, som anvendes i forbindelse med udførelse af et projekt. Header-filen til TimeAndMaterials klassen ses herunder

```
#ifndef TIMEANDMATERIALS_H
#define TIMEANDMATERIALS_H

#include <ostream>

class TimeAndMaterials {
public:
    TimeAndMaterials();
    TimeAndMaterials(double amount);

    virtual ~TimeAndMaterials() = default;

    double getAmount() const { return mAmount; }
    void setAmount(double amount) { mAmount = amount; }

    virtual double getPrice() const = 0;
    virtual std::string getText() const = 0;
    virtual double totalPrice() const { return getAmount()*getPrice(); }

private:
    double mAmount;
};

#endif // TIMEANDMATERIALS_H
```

Opgaven vurderes ud fra en korrekt anvendelse af de tilegnede C++ færdigheder.

Det forventes, at der anvendes referencer, når der er tale om ikke-simple datatyper. Desuden forventes det, at const-keyword anvendes, hvor det er muligt.

a) Implementer klassen WorkHours og klassen Materials

I første del af opgaven skal klassen TimeAndMaterials implementeres færdig og der skal implementeres to klasser, som nedarver fra TimeAndMaterials.

På klassen TimeAndMaterials skal de to constructorer implementeres og initialisere member variablen.

Klassen WorkHours skal have følgende funktionalitet og krav:

- Nedarve fra TimeAndMaterials
- Constructor med parameteren standardPrice (double) og default værdi 0
- Constructor med parametrene standardPrice (double) og extraPrice (double)
- Constructor med parametrene standardPrice, extraPrice samt antal (amount)
- Constructors skal initialisere alle member variable (attributter)
- Constructors skal kalde en passende moder-constructor fra TimeAndMaterials
- Overskrive funktionen getPrice fra moderklassen (summen af standardpris og ekstrapris)
- Overskrive funktionen getText fra moderklassen, som returnerer en beskrivelse af arbejdet
- Set-funktion setStandardPrice til at sætte standardPrice
- Set-funktion setExtraPrice til at sætte extraPrice
- Set-funktion til at sætte en tekst, som beskriver arbejdet.

Klassen Material skal have følgende funktionalitet og krav

- Nedarve fra TimeAndMaterials
- Default constructor
- Constructor med parametrene description (beskrivelse af materialet, string) og enhedspris (double)
- Constructors skal initialisere alle member variable (attributter)
- Constructors skal kalde en passende moder-constructor fra TimeAndMaterials
- Set-funktionen setPrice, som skal sætte enhedspris (double)
- Set-funktionen setDescription, som skal sætte en beskrivelse (string)
- Overskrive funktionen getPrice
- Overskrive funktionen getText

b) Tilføj operatoren

Tilføj følgende operatorer til klassen TimeAndMaterials

- Insertion operator (<<)

Insertion operatoren skal give et output på dette format:

8 tegn til antal (højrejusteret), et mellemrum, 40 tegn til den beskrivende tekst (venstrejusteret), 10 tegn til prisen (højrejusteret), 10 tegn til linjetotal som findes vha totalPrice-funktionen (højrejusteret).

Se test-kode på næste side (HUSK screendump af testen):

Test-kode

```
std::cout << "-----" << std::endl;
std::cout << "--" << std::endl;
std::cout << "-----" << std::endl;
std::cout << "-----" << std::endl;

WorkHours* w1 = new WorkHours;
WorkHours* w2 = new WorkHours(500);
WorkHours* w3 = new WorkHours(500, 120);
WorkHours* w4 = new WorkHours(500, 130, 2);
WorkHours* w5 = new WorkHours(*w3);

w2->setDescription("Mandetimer");
w3->setDescription("Maskintimer");
w4->setDescription("Overarbejdstimer");
w5->setDescription("Ekstratimer");
w5->setAmount(8);

Material* m1 = new Material();
Material* m2 = new Material("M\x9Btrik", 2);
Material* m3 = new Material("Gevindstang", 30);
m1->setDescription("Bolt");
m1->setPrice(10);
m1->setAmount(2);
m2->setAmount(5);
m3->setAmount(3);

std::cout << std::right << std::setw(8) << "Antal" << " " <<
std::left << std::setw(40) << "Udf\x9Brt arbejde og
materialeforbrug" <<
std::right << std::setw(10) << "Pris" <<
std::setw(10) << "I alt" << std::endl;
std::cout << *w1 << std::endl;
std::cout << *w2 << std::endl;
std::cout << *w3 << std::endl;
std::cout << *w4 << std::endl;
std::cout << *w5 << std::endl;
std::cout << *m1 << std::endl;
std::cout << *m2 << std::endl;
std::cout << *m3 << std::endl;

std::cout << std::endl << std::endl;
```

Opgave 1

Opgave 2 (1 time, 25 point)

Nu skal vi se på klassen TimeSheet. TimeSheet klassen vil anvende de to tidligere udviklede klasser Materials og WorkHours.

```
#ifndef TIMESHEET_H
#define TIMESHEET_H

#include "TimeAndMaterials.h"
#include <vector>

class TimeSheet {
public:
    TimeSheet();

    virtual ~TimeSheet();

    void addTimeAndMaterial(TimeAndMaterials* tm);
    void sortByType();

private:
    std::vector<TimeAndMaterials*> mItems;
};

#endif // TIMESHEET_H
```

a) addTimeAndMaterial, destructor samt Insertion operator

Vi skal her implementere funktionen addTimeAndMaterial som tilføjer et objekt til TimeSheet i member variablen mItems, en destructor som sletter alle oprettede objekter, samt lave en insertion operator som udskriver en tabel over alle arbejdstimer og materialer på følgende format:

8 tegn til antal (højrejusteret), et mellemrum, 1 tegn til type af omkostning, 1 mellemrum, 40 tegn til den beskrivende tekst (venstrejusteret), 10 tegn til prisen (højrejusteret), 10 tegn til linjetotal som findes vha totalPrice-funktionen (højrejusteret).

Typen af omkostningen defineres som M = Material og W = WorkHour.

b) Sortering efter type

Her skal der implementeres funktionen sortByType, som benytter en sorteringsalgoritme til at sortere alle materialer (Materials) og arbejdstimer (WorkHours) efter den oprindelige objekttype.

HINT: Anvend en lambda-funktion i forbindelse med std::sort.

Test

Se test-kode på næste side (HUSK screendump):

```
std::cout << "-----" << std::endl;
std::cout << "--" << std::endl;
std::cout << "-----" << std::endl;

TimeSheet ts;
ts.addTimeAndMaterial(w1);
ts.addTimeAndMaterial(m1);
ts.addTimeAndMaterial(w2);
ts.addTimeAndMaterial(w3);
ts.addTimeAndMaterial(m2);
ts.addTimeAndMaterial(w4);
ts.addTimeAndMaterial(m3);
ts.addTimeAndMaterial(w5);

std::cout << ts << std::endl << std::endl;

ts.sortByType();

std::cout << ts << std::endl;
```

Opgave 2