

Kursusarbejde 2

Grundlæggende Programmering

Arne Jørgensen, 300473-2919
klasse dm032-1a

31. oktober 2003

Indhold

1. Kode	2
1.1. hotel.h	2
1.2. hotel.cc	3
1.3. soehøj.cc	4
1.4. Om koden	7
2. Datafiler	7
2.1. takster.dat	7
2.2. bestillinger.dat	7
3. Afprøvning	7

1. Kode

1.1. hotel.h

```
#ifndef HOTEL_H
#define HOTEL_H
#include <iostream>
using namespace std;
5
enum Vaerelse { ENKELT, DOBBELT, LUKSUS};

class Takster
{
10 public:
    double enkeltvaerelse, dobbeltvaerelse, luksusvaerelse, enkelt_stat, dobbelt_stat;
    int rabat_pct, rabat_dage, udsigt_pct, moms;

    Takster();
15 // Pre: –
    // Post: taksterne er læst ind fra filen takster.dat
};

class Hotelophold
20 {
private:
    bool statsaftale, soeudsig;
    int antal_overnatninger, antal_vaerelser;
    Vaerelse vaerelse;
25
public:
    Hotelophold();
    // Pre: –
    // Post: nyt objekt er konstrueret og attributterne tildelt en række standardværdier
30
    double pris();
    // Pre: –
    // Post: returnerer prisen på et ophold ud fra de oplysninger om opholdet der er sat
35
    double pris_med_moms();
    // Pre: –
    // Post: returnerer prisen for opholdet med moms lagt til

    void SetStatsaftale(bool statsaftale_ = true);
40 // Pre: –
    // Post: hvis parametren er true er opholdet på statsaftale

    void SetVaerelsestype(Vaerelse vaerelse_);
45 // Pre: –
    // Post: opholdet foretages på statsaftale

    void SetSoeudsig(bool soeudsig_ = true);
    // Pre: –
    // Post: hvis parametren er true er opholdet med søudsig
50

    void SetAntalOvernatninger(int overnatninger);
    // Pre: overnatninger bør være større end nul
    // Post: antallet af overnatninger er sat for opholdet

55
    void SetAntalVaerelser(int vaerelser);
    // Pre: vaerelser bør være større end nul
    // Post: antallet af værelser er sat for opholdet

    friend ostream& operator << (ostream& str, Hotelophold& ophold);
60 // Pre: hvis str er en fil er filen åbnet
    // Post: skriver oplysninger om opholdet på output-strømmen, kommasepareret
};

ostream& operator << (ostream& str, Vaerelse vaerelse);
65 // Pre: hvis str er en fil er filen åbnet
    // Post: skriver en læsevenlig betegnelse for vaerelse på output-strømmen

#endif // HOTEL_H
```

1.2. hotel.cc

```
#include "hotel.h"
#include <fstream>

double Hotelophold::pris()
5 {
    Takster takster;
    double pris = 0.0;

    if (statsaftale)
10 {
        switch (vaerelse)
            {
                case ENKELT:
                    pris = takster.enkelt_stat;
                    break;
15                case DOBBELT:
                    pris = takster.dobbelt_stat;
                    break;
                case LUKSUS:
                    exit(1); // Lüksus dobbeltværelse med bad må ikke forekomme med statsaftale
                    break;
                };
            }
        else
25 {
            switch (vaerelse)
                {
                    case ENKELT:
                        pris = takster.enkeltvaerelse;
                        break;
30                    case DOBBELT:
                        pris = takster.dobbeltvaerelse;
                        break;
                    case LUKSUS:
                        pris = takster.luksusvaerelse;
                        break;
35                };
            };

40 pris *= antal_overnatninger;
    pris *= antal_vaerelser;

    // træk rabat fra hvis det er et langt ophold
    if (antal_overnatninger > takster.rabat_dage)
45     pris *= 1.00 - (takster.rabat_pct/100.0);

    // læg tillæg til hvis opholdet er med søudsigt (ej statsaftale)
    if (!statsaftale && soeudsigt)
        pris *= 1.00 + (takster.udsigt_pct/100.0);
50

    // rund af til nærmeste hele øre og returnér
    pris = (floor((pris*100)+0.5)/100.0);
    return pris;
}
55

double Hotelophold::pris_med_moms()
{
    Takster takster;

60 //returnér med moms lagt til
    return pris()*(1.00+(takster.moms/100.0));
}

void Hotelophold::SetStatsaftale(bool statsaftale_)
65 {
    statsaftale = statsaftale_;
}

void Hotelophold::SetVaerelsestype(Vaerelse vaerelse_)
70 {
```

```

    vaerelse = vaerelse_;
}

void Hotelophold::SetSoeudsig(bool soeudsig_)
75 {
    soeudsig = soeudsig_;
}

void Hotelophold::SetAntalOvernatninger(int overnatninger)
80 {
    antal_overnatninger = overnatninger;
}

void Hotelophold::SetAntalVaerelser(int vaerelser)
85 {
    antal_vaerelser = vaerelser;
}

Hotelophold::Hotelophold()
90 {
    SetStatsaftale(false);
    SetVaerelsestype(ENKELT);
    SetSoeudsig(false);
    SetAntalOvernatninger(1);
95 }

Takster::Takster()
{
    ifstream takstfil("takster.dat");
100
    if (takstfil.fail())
        exit(1); // kunne ikke indlæse taksterne fra filen
    takstfil >> enkeltvaerelse >> dobbeltvaerelse >> luksusvaerelse
        >> enkelt_stat >> dobbelt_stat >> rabat_pct >> udsigt_pct
105     >> rabat_dage >> moms;
}

ostream& operator << (ostream& str, Vaerelse vaerelse)
{
110     switch(vaerelse)
        {
            case ENKELT:
                str << "Enkeltværelse med bad";
                break;
115             case DOBBELT:
                str << "Dobbeltværelse med bad";
                break;
            case LUKSUS:
                str << "Luksus dobbeltværelse med bad";
120             break;
        };
    return str;
}

125 ostream& operator << (ostream& str, Hotelophold& ophold)
{
    str.setf(ios::fixed);
    str.precision(2);
    str << ophold.vaerelse << ", " << ophold.antal_vaerelser << ", " << ophold.antal_overnatninger << ", "
130     << ophold.statsaftale << ", " << ophold.soeudsig << ", " << ophold.pris_med_moms();
    return str;
}

```

1.3. soehoej.cc

```

#include <iostream>
#include <fstream>
#include "hotel.h"

5 void forespoergsel();
// Pre: –
// Post: kontrolstruktur til at lave en forespørgsel af et ophold

```

```
void registrering();
10 // Pre: –
    // Post: kontrolstruktur til at lave en registrering af et ophold

void indtast_oplysninger(Hotelophold& ophold);
15 // Pre: ophold er et konstrueret objekt hvor oplysninger om opholdet ønskes indlæst i
    // Post: ophold indeholder alle (og korrekte) oplysninger om opholdet

char input_ja_nej();
// Pre: der er stillet et ja/nej spørgsmål på cout og svaret ønskes indlæst
// Post: returnerer et 'j' eller 'n' for hhv. et ja og et nej-svar
20

int main()
{
    using namespace std;

25    int valg;
    do
    {
        cout << endl;
        cout << "1: Kunde forespørgsel\n";
30        cout << "2: Registrering\n";
        cout << "9: Afslut\n";
        cout << endl;
        cout << "Indtast dit valg; 1, 2 eller 9: ";
        cin >> valg;
35        cout << endl;
        switch (valg)
        {
            case 1:
                forespoergsel();
40                break;
            case 2:
                registrering();
                break;
        };
45    } while (valg != 9);

    return 0;
}

50 void forespoergsel()
{
    using namespace std;

    Hotelophold ophold;
55
    indtast_oplysninger(ophold);
    cout.setf(ios::fixed);
    cout.precision(2);
    cout << "Pris i alt excl. moms: " << ophold.pris() << endl;
60    cout << "Pris i alt incl. moms: " << ophold.pris_med_moms() << endl;
}

void registrering()
{
65    using namespace std;

    Hotelophold ophold;
    int kundennummer, dato;

70    cout << "Indtast kundennummer: ";
    cin >> kundennummer;
    cout << "Indtast dato for ophold (aaaammdd): ";
    cin >> dato;
    indtast_oplysninger(ophold);
75    cout.setf(ios::fixed);
    cout.precision(2);
    cout << "Pris i alt excl. moms: " << ophold.pris() << endl;
    cout << "Pris i alt incl. moms: " << ophold.pris_med_moms() << endl;

80    ofstream bestillinger("bestillinger.dat", ios::app);
    if (bestillinger.fail())
```

```
    cout << "Kunne ikke skrive til bestillingsfilen. Ikke registreret!\n";
else
{
85     bestillinger << kundennummer << ", "
           << dato << ", "
           << ophold << endl;
}
}
90 void indtast_oplysninger(Hotelophold& ophold)
{
    using namespace std;

95     char statsaftale, soeudsigt;
    int vaerelse, antal_vaerelser, antal_overnatninger;

    // statsaftale
    cout << "Statsaftale? [j/n] ";
100    statsaftale = input_ja_nej();
    cout << endl;
    if (statsaftale == 'j')
        ophold.SetStatsaftale();
    else
105        ophold.SetStatsaftale(false);

    // vaerelsestype
    cout << "1: " << ENKELT << endl;
    cout << "2: " << DOBBELT << endl;
110    if (statsaftale != 'j')
        cout << "5: " << LUKSUS << endl;
    cout << "\nIndtast vaerelsestype: ";
    cin >> vaerelse;
    while (vaerelse != 1 && vaerelse != 2 && (vaerelse != 5 || statsaftale == 'j'))
115    {
        cout << "Ikke et gyldigt svar, prøv igen: ";
        cin >> vaerelse;
    }
    cout << endl;
120    switch (vaerelse)
    {
        case 1:
            ophold.SetVaerelsestype(ENKELT);
            break;
125        case 2:
            ophold.SetVaerelsestype(DOBBELT);
            break;
        case 5:
            ophold.SetVaerelsestype(LUKSUS);
130            break;
    }

    // soeudsigt
    if (statsaftale != 'j')
135    {
        cout << "Udsigt over søen? [j/n] ";
        soeudsigt = input_ja_nej();
        cout << endl;
        if (soeudsigt == 'j')
140            ophold.SetSoeudsigt();
        else
            ophold.SetSoeudsigt(false);
    }

145    // antal overnatninger
    cout << "Antal overnatninger? ";
    cin >> antal_overnatninger;
    while (antal_overnatninger < 1)
    {
150        cout << "Mindst en overnatning: ";
        cin >> antal_overnatninger;
    }
    cout << endl;
    ophold.SetAntalOvernatninger(antal_overnatninger);
```

```
155 //antal værelser
    cout << "Antal værelser? ";
    cin >> antal_vaerelser;
    while (antal_vaerelser < 1)
160     {
        cout << "Mindst et værelse: ";
        cin >> antal_vaerelser;
    }
    cout << endl;
165 ophold.SetAntalVaerelser(antal_vaerelser);
}

char input_ja_nej()
{
170     using namespace std;
    char input;

    cin >> input;
    while (tolower(input) != 'j' && tolower(input) != 'n')
175     {
        cout << "Svar venligst 'j' eller 'n' ";
        cin >> input;
    }
    return tolower(input);
180 }
```

1.4. Om koden

Klassen `Hotelophold` er forsøgt holdt forholdsvis simpel med kun de nødvendige sæt-funktioner, samt naturligvis beregning(er). I `hotel.h` defineres endvidere en *enum* med navne for de tre værelsestyper så man koden bliver lettere læselig.

Taksterne indlæses i et særskilt objekt (via konstruktoren). Takstklassen er af pladshensyn holdt mere simpel end jeg egentlig bryder mig om. I min oprindelige implementation var attributterne `private` og med tilhørende hent-funktioner. Men da det fylder en del i rapporten og der i øvrigt ingen kompleksitet er i klassen har jeg skåret hent-funktionerne væk og gjort attributterne `public`. Kønt er det ikke.

Output-operatoren, `<<`, er overladet så den kan udskrive pæne, læsevenlige navne for de tre værelsestyper hvis den kaldes med værelsestype (som defineret i `enum`'en) som anden paramater. Operatoren er endvidere overladet så det udskriver et helt opholdsobjekt kommasepareret til brug i registreringen (brugt i linje 87 i `soehoej.cc`).

Klasserne er bevist holdt helt fri for læsninger fra `cin` og skrivinger til `cout`. Dette er i stedet lagt i frie funktioner og `main`-løkken.

2. Datafiler

2.1. takster.dat

Takstfilen er udvidet med et ekstra felt i forhold til opgaveformuleringen. Sidste felt indeholder momssatsen.

```
698.00 998.00 1298.00 598.00 798.00 5 8 5 25
```

2.2. bestillinger.dat

```
12345, 20040807, Luksus dobbeltværelse med bad, 8, 6, 0, 1, 79904.88
10000, 20031030, Enkeltværelse med bad, 1, 1, 1, 0, 747.50
10001, 20031030, Dobbeltværelse med bad, 1, 5, 0, 0, 6237.50
10002, 20031030, Dobbeltværelse med bad, 1, 6, 0, 0, 7110.75
10003, 20031030, Dobbeltværelse med bad, 1, 7, 0, 0, 8295.88
10004, 20031030, Enkeltværelse med bad, 3, 5, 1, 0, 11212.50
```

3. Afprøvning

Programmet er afprøvet med en række testdata. I figur 1 til 2 på den følgende side ses skærmbilleder fra et par af afprøvningerne (for sammenlignelighedens skyld de data der blev anvendt i opgaveformuleringen).

```
~/Datamatiker/GPR/Kursusarbejde2
Eil Redigér Vis Terminal Navigation Hjælp
1: Kunde forespørgsel
2: Registrering
9: Afslut

Indtast dit valg; 1, 2 eller 9: 1

Statsaftale? [j/n] j

1: Enkeltværelse med bad
2: Dobbeltværelse med bad

Indtast værelsestype: 1

Antal overnatninger? 2

Antal værelser? 4

Pris i alt excl. moms: 4784.00
Pris i alt incl. moms: 5980.00

1: Kunde forespørgsel
2: Registrering
9: Afslut

Indtast dit valg; 1, 2 eller 9: █
```

Figur 1: En forespørgsel på et ophold med statsaftale (testdata fra opgaveformuleringen).

```
~/Datamatiker/GPR/Kursusarbejde2
Eil Redigér Vis Terminal Navigation Hjælp
1: Kunde forespørgsel
2: Registrering
9: Afslut

Indtast dit valg; 1, 2 eller 9: 2

Indtast kundenummer: 12345
Indtast dato for ophold (aaaammdd): 20031030
Statsaftale? [j/n] n

1: Enkeltværelse med bad
2: Dobbeltværelse med bad
5: Luksus dobbeltværelse med bad

Indtast værelsestype: 5

Udsigt over søen? [j/n] j

Antal overnatninger? 6

Antal værelser? 8

Pris i alt excl. moms: 63923.90
Pris i alt incl. moms: 79904.88

1: Kunde forespørgsel
2: Registrering
9: Afslut

Indtast dit valg; 1, 2 eller 9: █
```

Figur 2: En registrering af et ophold med søudsigt (testdata fra opgaveformuleringen)

Resultatet af et par flere yderligere afprøvninger fremgår af `bestillinger.dat`, se afsnit 2.2 på foregående side.

Programmets beregninger, kontrolstrukturer, etc., er endvidere under udviklingen afprøvet ved hjælp af diverse driver-programmer. Disse er ikke vedlagt.