

Kursusarbejde 1

Grundlæggende Programmering

Arne Jørgensen, 300473-2919
klasse dm032-1a

3. oktober 2003

Kode

```
//File Name: kaalhovede.cc
//Author: Arne Jørgensen
//Email Address: arne@arnested.dk, arnjour@niels.brock.dk
//Description: Et program der holder styr på en sending kålhoveder
5 //Last Changed: October 2, 2003

#include <iostream>
#include <cmath>
#include "Sending.h"

10 void opret_objekt(Sending& ny_sending);
// Pre: ny_sending er et objekt (i princippet uinitialiseret)
// Post: ny_sending indeholder et nykonstrueret objekt initialiseret med indlæste værdi

15 void indtast_nyt_kaalhovede(Sending& min_sending);
// Pre: min_sending indeholder objektet kålhovedet skal tilføjes
// Post: kålhovedets vægt er indlæst og lagt i objektet min_sending

void tilfoej_hoved(Sending& min_sending, double vaegt);
20 // Pre: kålhovedet med vægten vaegt ønskes tilføjet til sendingen min_sending
// Post: kålhovedet er tilføjet sendingen såfremt kålhovedet ligger indenfor grænseværdierne

void fjern_hoved(Sending& min_sending, double vaegt);
// Pre: kålhovedet med vægten vaegt ønskes fjernet fra sendingen min_sending
25 // Post: kålhovedet er fjernet fra sendingen såfremt kålhovedet ligger indenfor grænseværdierne
// og ikke overstiger den samlede vægt i sendingen

void udskriv_vejledning();
// Pre: –
30 // Post: til skærmen er udskrevet en brugsvejledning til programmet

void udskriv_loebende_status(Sending& min_sending);
// Pre: min_sending indeholder en sending der ønskes en status over
// Post: status over manglende/for meget vægt er udskrevet til skærmen
35

void udskriv_afsluttende_status(Sending& min_sending);
// Pre: min_sending indeholder en sending der ønskes en status over
// Post: den samlede og endelige vægt på sendingen er udskrevet til skærmen

40 int main()
{
    udskriv_vejledning();

    // jeg konstruerer objektet med nogle dummy-værdier da de rette
45 // værdier bliver indlæst mm. i opret_objekt()
    Sending min_sending(0, 0.0, 0.0, 0, 0);
    opret_objekt(min_sending);

    // tilføj og fjern kålhoveder sålænge er ikke er for meget eller for lidt
50 do
    {
        indtast_nyt_kaalhovede(min_sending);
        udskriv_loebende_status(min_sending);
    } while (min_sending.for_meget() > 0.0 || min_sending.mangler() > 0.0);
55 udskriv_afsluttende_status(min_sending);
    return 0;
}

void opret_objekt(Sending& min_sending)
60 {
    using namespace std;

    int kg_der_skal_vaere;
    double max_kg_pr_hovede, min_kg_pr_hovede, pct_over, pct_under;
65
```

```

    cout << "Indtast sendingens vægt i kg: ";
    cin >> kg_der_skal_vaere;
    cout << "Hvormange procent lavere end " << kg_der_skal_vaere << " kg er tilladeligt: ";
    cin >> pct_under;
70  cout << "Hvormange procent højere end " << kg_der_skal_vaere << " kg er tilladeligt: ";
    cin >> pct_over;
    cout << endl;
    cout << "Indtast den mindste tilladte vægt for et kålhovede: ";
    cin >> min_kg_pr_hovede;
75  cout << "Indtast den højeste tilladte vægt for et kålhovede: ";
    cin >> max_kg_pr_hovede;

    //konstruér et nyt anonymt objekt ud fra de indtastede data og til det til min_sending
    min_sending = Sending(kg_der_skal_vaere, max_kg_pr_hovede, min_kg_pr_hovede, pct_over, pct_under);
80 }

void indtast_nyt_kaalhovede(Sending& min_sending)
{
    using namespace std;
85    double vaegt;

    cout << endl;
    cout << "Indtast vægten på det tilføjede/fjernede kålhovede: ";
90    cin >> vaegt;
    // skal hovedet tilføjes eller fjernes?
    if (vaegt < 0)
        fjern_hoved(min_sending, fabs(vaegt));
    else
95    tilfoej_hoved(min_sending, vaegt);
}

void tilfoej_hoved(Sending& min_sending, double vaegt)
{
100    using namespace std;

    if (min_sending.for_stort(vaegt) || min_sending.for_lille(vaegt))
        cout << "Den indtastede vægt skal være mellem " << min_sending.hent_min_pr_hoved()
            << " og " << min_sending.hent_max_pr_hoved() << " kg.\n";
105    else
        {
            min_sending.tilfoej_hoved(vaegt);
            cout << "Kålhovedet er tilføjet.\n";
        }
110 }

void fjern_hoved(Sending& min_sending, double vaegt)
{
    using namespace std;
115    if (min_sending.for_stort(vaegt) || min_sending.for_lille(vaegt))
        cout << "Den indtastede vægt skal være mellem " << min_sending.hent_min_pr_hoved()
            << " og " << min_sending.hent_max_pr_hoved() << " kg.\n";
    else
120    {
        if (min_sending.fjern_hoved(vaegt))
            cout << "Kålhovedet er fjernet.\n";
        else
            cout << "Kålhovedet er ikke fjernet da det er større end den totale vægt i sendingen.\n";
125    }
}

void udskriv_vejledning()
{
130    using namespace std;

    cout << "Pakning af kål\n\n"

```

```

    <<"Først indtastes de nødvendige oplysninger om sendingen.\n"
    <<"Derefter indtastes vægten på hvert kålhoved efter nedenstående:\n\n"
135 <<"Indtast vægten af hvert kålhoved, der fjernes fra eller tilføjes til sendingen.\n\n"
    <<"Tilføjes et kålhovede skal antal kg indtastes.\n"
    <<"Fjernes et kålhovede skal der indtastes er minus (-) foran antal kg.\n\n";
}

140 void udskriv_loebende_status(Sending& min_sending)
{
    using namespace std;

    if (min_sending.for_meget() > 0.0)
145     cout << "Maksimum for sendingen er overskredet med " << min_sending.for_meget()
        << " kg, fjern et kålhoved.\n";
    if (min_sending.mangler() > 0.0)
        cout << "Der mangler " << min_sending.mangler() << " kg.\n";
}

150 void udskriv_afsluttende_status(Sending& min_sending)
{
    using namespace std;

155     cout << "\nSendingen er færdigpakket.\n\n" << "Samlet levering: "
        << min_sending.hent_aktuelt_antal_kg() << " kg." << endl;
}

```

Stort set al programmets funktionalitet er lagt ud i frie funktioner og main (linje 40-57) indeholder således kun funktionskald og en iteration der kontrollerer indlæsningen af dataene.

De frie funktioner er – af hensyn til overskueligheden – skåret ned til at gøre brug af højst 1-2 selektioner. Hvor der har været brug for yderligere er noget af funktionaliteten flyttet ud i yderligere funktioner.

Alle de frie funktioner henter nødvendige oplysninger om sendingen direkte i objektet som de alle har fået givet med som en call-by-reference parameter.

Afprøvning

Afprøvningen af programmet i figur 1 på den følgende side viser en kørsel af programmet på data der er brugt som eksempel i opgaveformuleringens side 5.

Bug i Sending.h?

Yderligere afprøvning afslørede en mulig bug i Sending.h.

I Sending.h defineres funktionen fjern_hoved:

```

125 bool Sending::fjern_hoved(double kg)
{
    if ( for_stort(kg) || for_lille(kg) )
        return false;
    if (kg < aktuelt_antal_kg)
130     {
        aktuelt_antal_kg -= kg;
        return true;
    }
    return false;
135 }

```

Som det fremgår af funktionen fjernes et kålhoved kun fra sendingen hvis dets vægt er mindre end den samlede vægt af sendingen. Funktionen burde nok snarere være defineret med et mindre end eller lig med (\leq) i linje 129 for at undgå situationen i figur 2 på næste side.

```

129     if (kg <= aktuelt_antal_kg)

```

```
~/Datamatiker/GPR/Kursusarbejde1
Fil Redigér Vjs Terminal Navigation Hjælp
Pakning af kål

Først indtastes de nødvendige oplysninger om sendingen.
Derefter indtastes vægten på hvert kålhoved efter nedenstående:

Indtast vægten af hvert kålhoved, der fjernes fra eller tilføjes til sendingen.

Tilføjes et kålhovede skal antal kg indtastes.
Fjernes et kålhovede skal der indtastes er minus (-) foran antal kg.

Indtast sendingens vægt i kg: 10
Hvormange procent lavere end 10 kg er tilladeligt: 5
Hvormange procent højere end 10 kg er tilladeligt: 8

Indtast den mindste tilladte vægt for et kålhovede: 3
Indtast den højeste tilladte vægt for et kålhovede: 5

Indtast vægten på det tilføjede/fjernede kålhovede: 2
Den indtastede vægt skal være mellem 3 og 5 kg.
Der mangler 9.5 kg.

Indtast vægten på det tilføjede/fjernede kålhovede: 3
Kålhovedet er tilføjet.
Der mangler 6.5 kg.

Indtast vægten på det tilføjede/fjernede kålhovede: 4.5
Kålhovedet er tilføjet.
Der mangler 2 kg.

Indtast vægten på det tilføjede/fjernede kålhovede: 5
Kålhovedet er tilføjet.
Maksimum for sendingen er overskredet med 1.7 kg, fjern et kålhoved.

Indtast vægten på det tilføjede/fjernede kålhovede: -3
Kålhovedet er fjernet.

Sendingen er færdigpakket.

Samlet levering: 9.5 kg.
```

Figur 1: Afprøvning med data fra opgaveformuleringen

```
~/Datamatiker/GPR/Kursusarbejde1
Fil Redigér Vjs Terminal Navigation Hjælp
Pakning af kål

Først indtastes de nødvendige oplysninger om sendingen.
Derefter indtastes vægten på hvert kålhoved efter nedenstående:

Indtast vægten af hvert kålhoved, der fjernes fra eller tilføjes til sendingen.

Tilføjes et kålhovede skal antal kg indtastes.
Fjernes et kålhovede skal der indtastes er minus (-) foran antal kg.

Indtast sendingens vægt i kg: 10
Hvormange procent lavere end 10 kg er tilladeligt: 5
Hvormange procent højere end 10 kg er tilladeligt: 8

Indtast den mindste tilladte vægt for et kålhovede: 3
Indtast den højeste tilladte vægt for et kålhovede: 5

Indtast vægten på det tilføjede/fjernede kålhovede: 4
Kålhovedet er tilføjet.
Der mangler 5.5 kg.

Indtast vægten på det tilføjede/fjernede kålhovede: -4
Kålhovedet er ikke fjernet da det er større end den totale vægt i sendingen.
Der mangler 5.5 kg.

Indtast vægten på det tilføjede/fjernede kålhovede: █
```

Figur 2: Forsøg på at fjerne et kålhoved hvis vægt er lig med den samlede vægt