

```

/**
 *
 *  Obligatorisk oppgave nr 1 Våren 2022, PROG1003
 *
 *  @file Oblig1.cpp
 *  @author Loke Svelland
 *
 */

#include <iostream>    // cout / cin
#include <iomanip>      // setw
#include <vector>       // FOR å KUNNE BRUKE: STLs vector
#include <string>       // STRING-KLASSEN
#include "LesData2.h"  // Verktøykasse for lesing av diverse data
using namespace std;  // Slipper da å bruke std::cout / std::cin

const int ANTSTOPP = 11; ///< Totalt antall ulike busstopp.

struct Rute
{
    vector<string> stopp; // Rutens ulike stoppesteder.
    int ruteNr,          // Reelt rutenr, f.eks. 42, 165, 718
        totMin;          // Totalt antall minutter å kjøre på ruten
};                        // (fra første til siste stoppested).

vector<Rute *> gRuter; ///< Pekere til rutene.

const vector<string> gBusstopp = ///< Navn på alle busstopp.
{
    "Skysstasjonen", "Fahlstrøms plass", "Sykehuset",
    "Gjøvik stadion", "Bergslia", "Overby", "Nybrua",
    "NTNU", "Kallerud", "Hunndalen", "Mustad fabrikker"};

const int gMinutter[ANTSTOPP][ANTSTOPP] = ///< Min.mellom stoppesteder.
// const vector <vector <int> > gMinutter = //
// Alternativt (nå lært i vår).
{
    {0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0}, // Skysstasjonen = 0
    {3, 0, 3, 0, 0, 0, 3, 0, 0, 0, 4}, // Fahlstrøms plass = 1
    {0, 3, 0, 1, 0, 0, 0, 0, 0, 0, 0}, // Sykehuset = 2
    {0, 0, 1, 0, 3, 0, 0, 0, 0, 0, 0}, // Gjøvik stadion = 3
    {0, 0, 0, 3, 0, 2, 0, 0, 0, 0, 0}, // Bergslia = 4
    {0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0}, // Overby = 5
    {0, 3, 0, 0, 0, 0, 0, 2, 0, 0, 2}, // Nybrua = 6
    {0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 4}, // NTNU = 7
    {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, // Kallerud = 8
    {0, 0, 0, 0, 0, 0, 0, 4, 0, 0, 2}, // Hunndalen = 9
    {0, 4, 0, 0, 0, 0, 0, 2, 0, 0, 2}}; // Mustad fabrikker = 10

void skrivMeny();
void skrivStopp();
void skrivRuter();
void ruteSkrivData(const Rute rute);
void skrivNesteStoppesteder(const int stopp);
void nyRute();
void slett();
void slett(const int nr);
void slettRute();
bool ruteLesData(Rute &rute);

/**
 * hovedprogram
 */
int main()
{
    char kommando;

```

```

    skrivMeny();
    kommando = lesChar("\nKommando");

    while (kommando != 'Q')
    {
        switch (kommando)
        {
            case 'N':    nyRute();           break;
            case 'S':    slettRute();        break;
            case 'A':    skrivRuter();       break;
            case 'B':    skrivStopp();       break;
            default:     skrivMeny();        break;
        }
        kommando = lesChar("\nKommando");
    }

    return 0;
}

/**
 * Skriver menyen ut på skjermen
 *
 */
void skrivMeny() {

    cout << "\nValg: \n"
         << "\tN - Ny rute\n"
         << "\tS - slett Rute\n"
         << "\tA - Skriv alle ruter\n"
         << "\tB - Skriv stopp\n"
         << "\tQ - avslutt\n";

}

/**
 * Skriver ut alle stoppene
 *
 */
void skrivStopp() {

    cout << "\nAlle stoppesteder:\n\n";
    //Skriver alle stopp stedene nummerert
    for(int i = 0; i < gBusstopp.size(); i++) {
        cout << setw(2) << i+1 << ". " << gBusstopp[i] << '\n';
    }

}

/**
 * SKriver ut alle ruter
 *
 */
void skrivRuter() {

    if(gRuter.size() > 0) {
        for(int i = 0; i < gRuter.size(); i++) {
            cout << "\nRute nr." << setw(2) << i+1 << " ";
            ruteSkrivData(*gRuter[i]);
        }
    } else if(gRuter.size() == 0) {
        cout << "\t\nIngen ruter lagt til\n\n\n";
    }

}

/**

```

```

*
* @param rute - ruten som blir skrevet ut
*/
void ruteSkrivData(const Rute rute) {

    cout << "\t\nRute nr." << rute.ruteNr << '\n'
         << "\tTotal minutt rute: " << rute.totMin << '\n'
         << "\tRute stopp: ";

    // skriver ut en pil mellom stoppestedene
    for (int i=0; i<rute.stopp.size(); i++){
        if (i>0) cout << " --> ";
        cout << rute.stopp[i];
    }
}

/**
 * Skriver ut neste lovlige stoppested
 *
 * @param stopp - nr på stoppe sted
 */
void skrivNesteStoppesteder(const int stopp) {

    for(size_t i = 0; i < ANTSTOPP; i++) {
        if(gMinutter[stopp][i] != 0) {
            cout << i + 1 << ". " << gMinutter[stopp][i] << " min " << gBusstopp[i] <<
"\n";
        }
    }
}

/**
 * Bruker registrerer ny rute
 */
void nyRute(){
    Rute* nyrute;

    nyrute = new Rute; //opprettet ny rute
    cout << "\nNy rute: \n";
    if(ruteLesData(*nyrute) == true){ //Innlesing success:
        gRuter.push_back(nyrute); //legges bakerst i vektoren
        ruteSkrivData(*nyrute); //skriv ut alle dens datamedl
    }
    else{ //error
        cout << "\n!INNLESING MISLYKTES!... rute slettes";
        delete nyrute; //slett ruten
    }
}

/**
 *
 * @param rute - ruten som blir lest
 * @return true
 * @return false
 *
 * ! - Hvis en legger inn partall antall stopp når bruker leser inn stopp til
 * ! - rutene får en opp "std::bad_alloc" og programmet kræsjer, finner ikke ut
 * ! - hvorfor
 */
bool ruteLesData(Rute & rute) {
    int fStopp = 0,
        nStopp = 0;

    rute.totMin = 0;

```

```

        // Leser inn rute nr
rute.ruteNr = lesInt("\n\tRute nr: ", 1, 999);
        // skriver alle stopp
skrivStopp();

        // Bruker velger startsted
fStopp = lesInt("\nVelg startsted: \n", 1, 11);
rute.stopp.push_back(gBusstopp[fStopp - 1]);
        // Skriver ut neste lovlige stopp
skrivNesteStoppesteder(fStopp - 1);

do
{
        // Leser inn neste stopp
nStopp = lesInt("\nNeste stopp: ", 0, ANTSTOPP);
        // Lagrer busstopp
rute.stopp.push_back(gBusstopp[nStopp - 1]);
rute.totMin += gMinutter[fStopp - 1][nStopp - 1];
        // Skriver ut neste lovlige stopp
skrivNesteStoppesteder(nStopp - 1);
        // leser inn neste stopp
fStopp = lesInt("\nNeste stopp: ", 0, ANTSTOPP);
        // lagrer vekk navn og min på stopp
rute.stopp.push_back(gBusstopp[fStopp - 1]);
rute.totMin += gMinutter[nStopp - 1][fStopp - 1];
        // skriver neste lovlige stopp
skrivNesteStoppesteder(fStopp - 1);

        fStopp = nStopp;

} while (fStopp != 0 && nStopp != 0);

if(rute.stopp.size() > 1) {
    return true;
} else {return false;}
}

/**
 * Bruker velger her hvilke(n) rute som skal slettes
 *
 */
void slettRute() {
    int valg = 0;

    if(gRuter.size() == 0) {
        cout << "\t\nIngen ruter lagt til\n\n";
    } else {
        // skriver ut alle ruter
        for(int i = 0; i < gRuter.size(); i++) {
            cout << "\n" << "Rute nr. " << i+1;
            ruteSkrivData(*gRuter[i]);
        }

        valg = lesInt("\nHvilken rute skal slettes: ", -1, gRuter.size());
        if(valg == -1) {
            // sletter alle ruter
            void slett();
        } else if(valg == 0) {
            // ingen slettes, går ut av funksjonen
            cout << "\t\nIngen rute fjernet, returnerer til startmeny\n\n";
            exit;
        } else if(valg <= 1) {
            // sletter valgt rute
            slett(valg);
        }
    }
}
}

```

```

/**
 * Sletter alle ruter
 *
 */
void slett() {

    for(int i = 0; i < gRuter.size(); i++) // Sletter alle rutene i vektoren
        delete gRuter[i];
    gRuter.clear();

    gRuter.clear(); // fjerner alle pekerne i vektoren
    cout << "\n\nvektoren er tom - antallet er: " << gRuter.size() << "\n\n";

}

/**
 *
 * @param nr - nr på rute som blir slettet
 */
void slett(const int nr) {
    delete gRuter[nr-1]; //sletter valgt rute
    gRuter[nr-1] = gRuter[gRuter.size()-1]; //bakerste flyttes til plassen
    gRuter.pop_back(); // siste peker slettes også
}

```