

```

/**
 * @file oblig1.cpp
 * @author Steffen S. Martinsen (steffema@stud.ntnu.no)
 * @version 0.1
 * @date 2023-02-01
 *
 * Program som holder orden på en students faste gjøremål hver uke.
 * Programmet kan lese inn et gjøremål, skrive ut opplysninger om et gjøremål,
 * og slette gjøremålene om ønskelig.
 */
#include <iostream>
#include <string>
#include <vector>
#include <iomanip>
#include "LesData2.h"
using namespace std;

const vector <string> UKEDAG{"Man", "Tirs", "Ons", "Tors", ///< Vektor med alle
                             ukedager.
                             "Fre", "Lor", "Son"};

struct Gjøremål {
    string   beskrivelse,    // Beskrivelse av gjøremålet.
            ukedag;          // Ukedagen det skjer.
    float    tid;            // Tiden det tar (ikke klokkeslett).
};

vector <Gjøremål*> gGjøremålene;    ///< Oversikt over alle gjøremålene.

void fjernAlleGjøremål();
void fjernEttGjøremål();
void gjøremålLesData(Gjøremål & gjøremål);
void gjøremålSkrivData(const Gjøremål* gjøremål);
void nyttGjøremål();
void skrivAlleGjøremål(const string dag = "Alle");
void skrivEnDagsGjøremål();
void skrivMeny();

int main() {
    char kommando;

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

    while (kommando != 'Q') {
        switch (kommando) {
            case 'N': nyttGjøremål();           break;
            case 'A': skrivAlleGjøremål();      break;
            case 'D': skrivEnDagsGjøremål();    break;
            case 'F': fjernEttGjøremål();       break;
            default:  skrivMeny();               break;
        }
        kommando = lesChar("\nKommando");
    }
}

```

```

    fjernAlleGjøremål();

    return 0;
}

/**
 * Fjerner alle gjøremålene.
 */
void fjernAlleGjøremål() {
    for (int i = 0; i < gGjøremålene.size(); i++) {
        delete gGjøremålene[i];
    }
    gGjøremålene.clear();
}

/**
 * Fjernet ett enkelt gjøremål.
 */
void fjernEttGjøremål() {
    int kommando;

    skrivAlleGjøremål();
    kommando = lesInt("Velg gjøremål som fjernes", 0,
        int(gGjøremålene.size()));

    if (kommando != 0) {
        delete gGjøremålene[kommando-1];
        for (int i = kommando; i < gGjøremålene.size(); i++) {
            gGjøremålene[i-1] = gGjøremålene[i];        // Flytter alle
                structer en plass nedover.
        }
        gGjøremålene.pop_back();    // Sletter den bakerste structen, som
            nå er tom.
    }
}

/**
 * Leser inn data for et gjøremål.
 *
 * @param gjøremål - gjøremålet det leses inn data for.
 */
void gjøremålLesData(Gjøremål & gjøremål) {
    int ukedag;
    cout << "Beskriv gjøremålet:\n";
    getline(cin, gjøremål.beskrivelse);

    ukedag = lesInt("Ukedag", 1, 7);
    gjøremål.ukedag = UKEDAG[ukedag-1];

    // 0 til 60 mimnutter.
    gjøremål.tid = lesFloat("Tid gjøremålet vil ta (Timer)", 0, 23);
}

/**

```

```

* Skriver ut data for et gjøremål.
*
* @param gjøremål - gjøremålet det skrives ut data for.
*/
void gjøremålSkrivData(const Gjøremål* gjøremål) {
    cout << "Beskrivelse: " << gjøremål->beskrivelse
         << "\tUkedag: " << gjøremål->ukedag
         << "\tTid: " << setprecision(2) << gjøremål->tid
         << "\n";
}

/**
* Funksjon som oppretter nytt gjøremål.
*
* @see gjøremålLesData(...)
*/
void nyttGjøremål() {
    Gjøremål* nyttGjøremål = new Gjøremål;
    cout << "\nOppretter nytt gjøremål.\n\n";
    gjøremålLesData(*nyttGjøremål);
    // Legger til det nye gjøremålet bakerst i vektoren.
    gGjøremålene.push_back(nyttGjøremål);
}

/**
* Skriver ut alle gjøremål for en gitt dag. Om ingen dag er gitt skrives gjøremål ut for alle dager.
*
* @param dag - dagen det skrives ut for. Om ingen dag er gitt skrives gjøremål ut for alle dager.
*/
void skrivAlleGjøremål(const string dag) {
    // Skriver ut alle dager
    if (dag == "Alle") {
        for (int i = 0; i < gGjøremålene.size(); i++) {
            cout << "Gjøremål nr. " << i+1 << "\n";
            gjøremålSkrivData(gGjøremålene[i]);
        }
    } else {
        // Skriver ut om gitt dag matcher.
        int counter = 0;
        for (int i = 0; i < gGjøremålene.size(); i++) {
            if (gGjøremålene[i]->ukedag == dag) {
                counter++;
            }
        }
        if (counter < 1 || gGjøremålene.size() < 1) {
            cout << "Finnes ingen gjøremål denne dagen.\n";
        } else {
            counter = 0; // Nullstiller counter for gjenbruk.
            for (int i = 0; i < gGjøremålene.size(); i++) {
                if (gGjøremålene[i]->ukedag == dag) {
                    counter++;
                    cout << "Gjøremål nr. " << counter << "\n";
                    gjøremålSkrivData(gGjøremålene[i]);
                }
            }
        }
    }
}

```

```

    }
}

/**
 * Funksjon for å velge dag å skrive ut gjøremål for.
 *
 * @see skrivAlleGjøremål(...)
 */
void skrivEnDagsGjøremål() {
    int kommando;

    do {
        cout << "Vennligst velg en ukedag:\n"
        << "1. Mandag\n" << "2. Tirsdag\n"
        << "3. Onsdag\n" << "4. Torsdag\n"
        << "5. Fredag\n" << "6. Lørdag\n"
        << "7. Søndag\n" << "\nValg: ";
        cin >> kommando;
        cin.ignore();
    } while (kommando < 1 || kommando > 7);

    skrivAlleGjøremål(UKEDAG[kommando-1]);
}

/**
 * Menyen brukt i main.
 */
void skrivMeny() {
    cout << "\nVennligst velg en kommando:\n\n"
    << "N = Nytt gjøremål\n"
    << "A = Skriv ut alle gjøremål\n"
    << "D = Skriv ut gjøremål for en dag\n"
    << "F = Fjern ett gjøremål\n";
}

```