Dokumentacja projektu - Programowanie w C 2

Politechnika Świętokrzyska w Kielcach

Grzegorz Bujak Jędrzej Cieślikiewicz Paweł Cieszkowski

Spis treści

Opis projektu

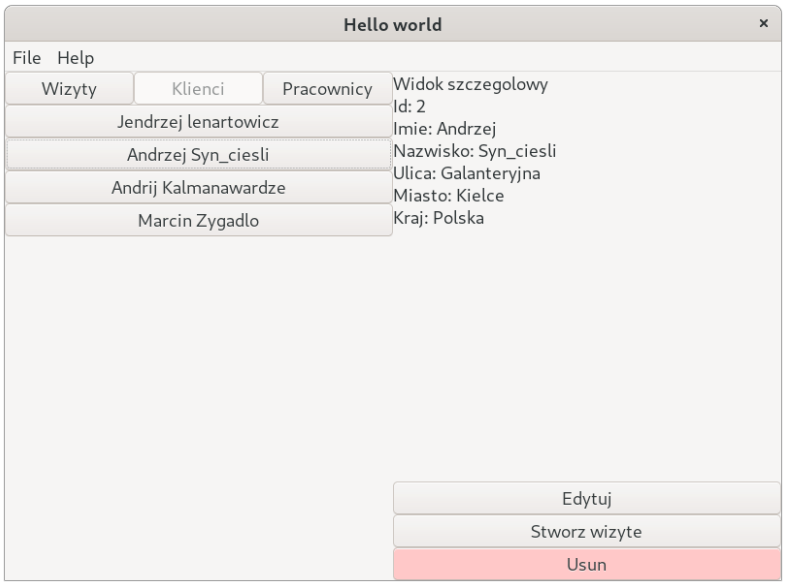
Przygotowaliśmy projekt o temacie “e-warsztat”. Przygotowany program ma ułatwiać organizację pracy w serwisie samochodowym. Wykonaliśmy to zadanie przygotowując program graficzny spełniający następujące funkcje:

- Baza klientów (lista klientów i danych o klientach)
- Baza pracowników
- Zapisywanie klientów na wizytę u konkretnego pracownika

Program został przygotowany przy użyciu biblioteki wxWidgets. Ta biblioteka umożliwia przygotowanie aplikacji graficznych działających na systemach operacyjnych Windows, MacOS oraz GNU/Linux.

Dokumentacja nie skupia się na poszczególnych klasach, lecz funkcjonalnościach programu - kilka razy wraca do opisywania tej samej klasy. Uważamy, że w ten sposób łatwiej zrozumieć implementację konkretnej funkcjonalności, która może przydać do innego projektu, bez zapoznawania się z całą strukturą aplikacji.

Interfejs graficzny



Rysunek 1: interfejs graficzny

Interfejs graficzny aplikacji przedstawiony na zdjęciu nr. 1 jest podzielony na dwie połowy. Lewa połowa przedstawia graficzną reprezentację bazy danych aplikacji.

Na szczycie lewego panelu znajdują się zakładki dla każdego typu danych przechowywanych w aplikacji. Poniżej paska z zakładkami dla każdej jednostki wybranego typu danych renderowany jest przycisk, którego naciśnięcie sprawia wyświetlenie szczegółowych danych o tej jednostce w prawym panelu.

Prawy panel zawiera duże pole tekstowe, w którym wyświetlane są szczegółowe informacje o przechowywanych danych. Ponadto, na spodzie prawego panelu znajdują się trzy przyciski służące do:

- edycji danych, których szczegóły wyświetlane są powyżej
- umówienie klienta na wizytę (przycisk aktywny tylko, gdy wybrane dane są typu klient)
- usunięcia wybranych danych z aplikacji

Implementacja

Klasa aplikacji (wxApp)

Protopyp klasy aplikacji (plik include/app.hpp):

```
#pragma once

#include <wx/wx.h>
#include "frame.hpp"

class MyApp : public wxApp {
    public:
    MyFrame* main_frame;
    virtual bool OnInit();
};
```

Implementacja (plik src/app.cpp):

```
#include <wx/wx.h>
#include "../include/app.hpp"
#include "../include/frame.hpp"

wxIMPLEMENT_APP(MyApp);

bool MyApp::OnInit() {
    main_frame = new MyFrame(
        "Hello world", {50, 50}, {800, 600});
    main_frame->Show(true);
    return true;
}
```

Klasa aplikacji jest typowa dla aplikacji korzystających z biblioteki wxWidgets. Jest to klasa (nazwana MyApp) dziedzicząca publicznie z klasy wxApp. Implementuje metodę OnInit, w której tworzy instancję klasy MyFrame i wywołuje jej metodę "Show".

Klasa MyFrame

```
#pragma once

#include <wx/wx.h>
#include "display_list.hpp"
#include "globals.hpp"

class MyFrame : public wxFrame {
    public:
    MyFrame(const wxString& title, const wxPoint& pos,
            const wxSize& size);
    private:
    void OnHello (wxCommandEvent& event);
    void OnExit  (wxCommandEvent& event);
    void OnAbout (wxCommandEvent& event);
    void OnSave  (wxCommandEvent& event);
    void OnOpen  (wxCommandEvent& event);
    void OnNew   (wxCommandEvent& event);

    wxDECLARE_EVENT_TABLE();
};

enum {
    ID_Hello = 1
};
```

Klasa MyFrame dziedziczy publicznie po klasie wxFrame. Metody, jakie zaimplementowaliśmy w tej klasie to konstruktor i kilka metod obsługujących wydarzenia.

- OnExit - funkcja obsługuje wybranie obcji exit w menubarze
- OnHello, OnAbout - funkcje obsługujące nazwane po sobie opcje w menubarze Wyświetlają informacje o programie
- OnSave, OnOpen - obsługują zapisywanie i wczytywanie danych do/z plików
- OnNew - funkcja obsługuje opcję stworzenia nowych danych

Implementacja zapisywania i wczytywania danych

Interfejs (klasa MyFrame)

```
void MyFrame::OnOpen(wxCommandEvent& event) {
    wxFileDialog openFileDialog(this, _("Open XYZ file"), "", "",
                                "XYZ files (*.xyz)|*.xyz", wxFD_OPEN|wxFD_FILE_MUST_EXIST);

    if (openFileDialog.ShowModal() == wxID_CANCEL)
        return;

    std::ifstream stream;
    stream.open(openFileDialog.GetPath());
    if (stream.fail()) {
        wxLogError("Cannot open file " + openFileDialog.GetPath());
        return;
    }

    g_datastore.load(stream);
    g_display_list->display(0);
}

void MyFrame::OnSave(wxCommandEvent& event) {
    wxFileDialog
        saveFileDialog(this, _("Save XYZ file"), "", "",
                       "XYZ files (*.xyz)|*.xyz", wxFD_SAVE|wxFD_OVERWRITE_PROMPT);
    if (saveFileDialog.ShowModal() == wxID_CANCEL)
        return;

    std::ofstream stream;
    stream.open(saveFileDialog.GetPath());
    if (stream.fail()) {
        wxLogError("Cannot open file " + saveFileDialog.GetPath());
        return;
    }

    g_datastore.save(stream);
}
```

W tym miejscu zaimplementowany jest tylko interfejs graficzny zapisywania i wczytywania danych. Prawdziwa implementacja znajduje się w klasie `datastore_t`.

Jest to typowa implementacja interfejsu wyboru pliku w bibliotece wxWidgets. Niewiele różni się od przykładowej implementacji znajdującej się na wikipedii biblioteki. Interfejs wyboru pliku nie jest przygotowany przez nas i różni się prawie całkowicie na różnych systemach operacyjnych.

"Prawdziwa" implementacja (klasa datastore_t)

Wszystkie informacje przechowywane w pamięci naszego programu znajdują się w klasie `datastore_t`. Jedna instancja tej klasy używana jest w całym programie, jako zmienna globalna `g_datastore`.