**ВІДОКРЕМЛЕНИЙ СТРУКТУРНИЙ ПІДРОЗДІЛ**

**«ОПТИКО-МЕХАНІЧНИЙ ФАХОВИЙ КОЛЕДЖ КИЇВСЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ ІМЕНІ ТАРАСА ШЕВЧЕНКА»**

Циклова комісія спеціальності «Інженерія програмного забезпечення»

**КУРСОВА РОБОТА**

**з дисципліни «Інструментальні засобі візуального програмування»**

на тему «Розробка автоматизованого робочого місця працівника бібліотеки»

|  |  |  |
| --- | --- | --- |
| ЗАХИЩЕНО З ОЦІНКОЮ  Національна шкала \_\_\_\_\_\_\_\_\_\_\_\_  Кількість балів \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Керівник  \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Соловйова О.В.  (підпис) |  | ВИКОНАВ |
| студент 3-го курсу гр. ІПЗ-31 спеціальність 121 «Інженерія програмного забезпечення» |
| Шестопал Іван Павлович  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (підпис виконавця) |

Члени комісії \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Заболотний В.О.

(підпис)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Соловйова О.В.

(підпис)

м. Київ — 2021

**ВІДОКРЕМЛЕНИЙ СТРУКТУРНИЙ ПІДРОЗДІЛ**

**«ОПТИКО-МЕХАНІЧНИЙ ФАХОВИЙ КОЛЕДЖ КИЇВСЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ ІМЕНІ ТАРАСА ШЕВЧЕНКА»**

ЗАТВЕРДЖУЮ

В.о. директора

\_\_\_\_\_\_ Гапріндашвілі Б.В.

« » 2022р.

|  |  |  |
| --- | --- | --- |
| **Дисципліна:** | Інструментальні засоби візуального програмування | |
| **Спеціальність:** | 121 «Інженерія програмного забезпечення» | |
| **Курс ІІІ** | **Група IПЗ-31** | **Семестр** **6** |

**ЗАВДАННЯ  
на курсову роботу студента**

|  |
| --- |
| Шестопала Івана Павловича |
| **1.Тема курсової роботи:**  Розробка автоматизованого робочого місця працівника бібліотеки |
| Затверджена протоколом циклової комісії №6 від 25.01.2022р. |
| **2.Технічні вимоги:** ЄСПД, ЄСКД, ДСТУ 3008-95 |
| **3.Термін здачі студентом закінченої роботи: «**08» червня 2022р. |
| **4.Вихідні дані до виконання роботи:** |
| * Проаналізувати предметну область, виконати опис проектованої системи |
| і правил її функціонування; |
| * Розробити графічний інтерфейс користувача для роботи з його |
| елементами: головне вікно програми, система меню форм, інформація про програму; |
| **5.Зміст пояснювальної записки** (перелік питань, які підлягають розробці): |
| вступ; постановка задачі; опис мови і середовища програмування; розробка |
| інтерфейсу програми; висновки; список літератури; програмний лістинг. |
| **6. Дата видачі завдання:** «25» січня 2022р. |

|  |  |  |
| --- | --- | --- |
| Керівник роботи |  | Соловйова Олена Володимирівна |
| Завдання прийняв до виконання |  | Шестопал Іван Павлович |

«25» січня 2022р.

**КАЛЕНДАРНИЙ ПЛАН-ГРАФІК**

**виконання курсової роботи з дисципліни**

**«Інструментальні засоби візуального програмування»**

|  |  |  |
| --- | --- | --- |
| **№ з/п** | **Етапи** | **Термін виконання** |
|  | Обрання та затвердження теми, попередня постановка задачі | 01.04-05.04.2022 |
|  | Підбір літератури, веб-джерел, ознайомлення зі стандартами | 06.04-12.04.2022 |
|  | Вивчення та аналіз предметної області об'єкту курсової роботи | 13.04-19.04.2022 |
|  | Підготовка оглядової частини | 20.04-26.04.2022 |
|  | Розробка архітектури та загальної структури системи | 27.04-03.05.2022 |
|  | Програмна реалізація системи | 04.05-10.05.2022 |
|  | Оформлення роботи, передача її на рецензування керівникові | 11.05-17.05.2022 |
|  | Рецензування роботи, уточнення назви, висновків, підготовка презентаційних матеріалів | 18.05-24.05.2022 |
|  | Попередній захист курсової роботи | 25.05-07.06.2022 |
|  | Захист курсової роботи | 08.06.2022 |

|  |  |  |
| --- | --- | --- |
| Керівник роботи |  | Соловйова Олена Володимирівна |
| Студент |  | Шестопал Іван Павлович |
|  |  | «\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2022р. |

**Зміст**

[**Реферат** 5](#_heading=h.1fob9te)

**ВСТУП** 6

[**РОЗДІЛ 1. ОПИС НАПРЯМКІВ ДОСЛІДЖЕННЯ**](#_heading=h.tyjcwt) 7

[**1.1 Постановка задачі**](#_heading=h.3dy6vkm) 7

[**1.2 Опис предметної області**](#_heading=h.1t3h5sf) 7

[**РОЗДІЛ 2. ІНСТРУМЕНТАЛЬНІ**](#_heading=h.4d34og8) **ЗАСОБИ РОЗРОБКИ** [9](#_heading=h.4d34og8)

[**2.1 Ознайомлення з C++ 9**](#_heading=h.2s8eyo1)

[**2.2 Ознайомлення з C++ Builder**](#_heading=h.26in1rg) 10

**РОЗДІЛ 3.** [**РЕЗУЛЬТАТИ**](#_heading=h.35nkun2) **РОБОТИ ПРОГРАМИ** 13

[**Висновок**](#_heading=h.1ksv4uv) [16](#_heading=h.35nkun2)

[**Список використаної літератури**](#_heading=h.17dp8vu)17

**Додаток 1. Код головної форми**18

# **Реферат**

Курсова робота складається із вступу, трьох розділів, висновка, списку використаних джерел, та одного додатку. Робота містить 5 рисунків. Загальний обсяг роботи становить 23 сторінки, основний текст викладено на 11 сторінках.

# **ВСТУП**

Виконання функцій бібліотекаря зазвичай передбачає багато ручної роботи з формулярами та каталогами на паперових носіях. Це призводить до значних витрат часу та зусиль працівників бібліотеки, ускладнює пошук та реєстрацію даних. Автоматизація багатьох необхідних функцій спростить та прискорить роботу бібліотекаря.

**Мета та завдання курсової роботи**

Метою роботи є визначення та автоматизація процесів, пов’язаних з роботою бібліотеки. Конкретним завданням курсової роботи є розробка графічного інтерфейсу користувача автоматизованого робочого місця (АРМ) працівника бібліотеки.

**Використане програмне забезпечення**

Засоби програмування, використані для розробки АРМ працівника бібліотеки:

- С++;

- C++ Builder

# **РОЗДІЛ 1.ОПИС НАПРЯМКІВ ДОСЛІДЖЕННЯ**

**1.1 Опис предметної області**

Основними бібліотечними процесами є робота з книжковим фондом та з читачами. Бібліотекар має контролювати актуальність каталожних даних, додавати книги до каталогу при нових надходженнях або видаляти при втраті або псуванні книг. Робота з читачами передбачає ведення їх реєстру, запис нових читачів або виключення тих, що припинили користування бібліотекою. Для кожного читача має відстежуватися та реєструватися видача та повернення книжок.

**1.2 Постановка задачі**

Для автоматизації визначених в підрозділі 1.1 процесів АРМ працівника бібліотеки має передбачати, як мінімум, наступний функціонал:

1. Робота з книжковим фондом.Можливість переглядати повний каталог книг з відповідними реквізитами, які включають, зокрема, автора, назву, рік видання, видавництво, кількість сторінок та кількість книг у наявності. Також можливість додавання книжок до каталогу та видалення з нього. Можливість пошуку/фільтрування за автором та назвою.

2. Робота з реєстром читачів. Можливість переглядати повний реєстр читачів книг з відповідними персональними даними, які включають, зокрема, ПІБ, стать, рік народження, номер телефону, видані йому книги. Також можливість додавання нових читачів до реєстру та видалення з нього. Можливість пошуку/фільтрування за ПІБ.

3. Робота безпосередньо з читачами. Реєстрація видачі та повернення книжок до бібліотечного фонду. Для кожного читача має фіксуватися які книжки та коли йому видані та ним повернені.

Графічний інтерфейс АРМ працівника бібліотеки повинен включати засоби навігації та пошуку у каталозі книжок та реєстрі читачів (наприклад, кнопки, меню, полоси прокрутки), засоби для прийняття рішень (наприклад, кнопки вибору), а також відповідні форми для внесення та відображення інформації.

# **РОЗДІЛ 2. ІНСТРУМЕНТАЛЬНІ ЗАСОБИ РОЗРОБКИ**

**2.1. Ознайомлення з C++**

Мова програмування С++ представляє високорівневу мову програмування загального призначення, що компілюється, яка підходить для створення найрізноманітніших додатків. На сьогоднішній день С++ є однією з найпопулярніших та найпоширеніших мов.

Своїм корінням він сягає мови С, яка була розроблена в 1969-1973 роках в компанії Bell Labs програмістом Деннісом Рітчі (Dennis Ritchie). На початку 1980-х років датський програміст Бьорн Страуструп (Bjarne Stroustrup), який на той час працював у компанії Bell Labs, розробив С++ як розширення до мови С. Практично спочатку C++ легко доповнював мову С деякими можливостями об'єктно-орієнтованого програмування. Згодом до нього були додані нові можливості, які зробили його не просто доповненням до С, а абсолютно новою мовою програмування. З тих пір обидві мови стали розвиватися незалежно друг від друга.

С++ є потужною мовою, успадкувавши від С багаті можливості по роботі з пам'яттю. Тому нерідко С++ знаходить своє застосування у системному програмуванні, зокрема, при створенні операційних систем, драйверів, різних утиліт, антивірусів тощо. ОС Windows переважно написана на С++. Але лише системним програмуванням застосування цієї мови не обмежується. С++ можна використовувати у програмах будь-якого рівня, де важливі швидкість роботи та продуктивність. Нерідко він застосовується до створення графічних додатків, різних прикладних програм. Також особливо часто його використовують для створення ігор із багатою насиченою візуалізацією. Крім того, останнім часом набирає хід мобільний напрямок, де С++ теж знайшов своє застосування. І навіть у веб-розробці також можна використовувати С++ для створення веб-програм або будь-яких допоміжних сервісів, які обслуговують веб-програми. Загалом С++ – мова широкого користування, якою можна створювати практично будь-які види програм.

С++ є мовою, що компілюється, а це означає, що компілятор транслює вихідний код на С++ у виконуваний файл, який містить набір машинних інструкцій. Але різні платформи мають свої особливості, тому скомпільовані програми не можна просто перенести з однієї платформи в іншу і там вже запустити. Однак на рівні вихідного коду програми на С++ переважно мають переносимість, якщо не використовуються якісь специфічні для поточної ОС функції. А наявність компіляторів, бібліотек та інструментів розробки майже під всі поширені платформи дозволяє компілювати той самий вихідний код на С++ у додатки під ці платформи.

На відміну від С мова C++ дозволяє писати додатки в об'єктно-орієнтованому стилі, представляючи програму як сукупність класів і об'єктів, що взаємодіють між собою. Це полегшує створення великих додатків.

**2.2. Ознайомлення з C++ Builder**

C++Builder - це продукт, що розроблявся компанією Borland, а потім її підрозділом CodeGear, який зараз належить компанії Embarcadero Technologies. Він призначений для швидкої розробки додатків Windows (RAD – Rapid Application Development) мовою С++. З допомогою C++ Builder можна створювати як консольні програми, так і графічний інтерфейс користувача (GUI – Graphical User Interface), тобто створювати меню, діалогові вікна, головне вікно тощо.

Інтегроване середовище C++Builder (IDE- Integrated Development Environment) поділено на три частини. Верхнє вікно є основним. Воно містить оперативну панель (speedbar) зліва та палітру компонентів (Component Palette) праворуч. Оперативна панель дозволяє одним кліком на відповідній кнопці виконувати такі дії, як відкриття, збереження та компіляція проектів.

Палітра компонентів містить великий набір компонентів, які можна вставляти у свої форми (до компонентів відносяться текстові позначки, вікна списку, кнопки тощо). Для зручності усі компоненти поділені на групи. Компонент (component) – це функціонально закінчена ділянка двійкового коду, що виконує деяку спеціальну зумовлену задачу, наприклад, відображення тексту, елемента редагування або вікна списку.

Під оперативною панеллю та палітрою компонентів, впритул до лівого краю екрану, розташований інспектор об'єктів (Object Inspector). За допомогою інспектора об'єктів можна змінювати властивості компонентів та події, на які вони реагують.

Вікно інспектора об'єктів має одну або дві вкладки залежно від того, який компонент вибрано. Вкладка Properties (Властивості) є завжди. Властивості об'єкта визначають його поведінку. Наприклад, зміна якості Color (Колір) призведе до зміни кольору фону даного об'єкта. Список доступних властивостей змінюється від компонента до компонента, проте компоненти мають кілька стандартних властивостей (наприклад, Width (Ширина) і Height (Висота). Властивість (property) визначає поведінку компонента.

Зазвичай інспектор об'єктів має вкладку Events (Події) на додаток до вкладки Properties. Події відбуваються тоді, коли користувач взаємодіє з компонентом, наприклад, клацає на ньому мишею. Можна написати код, який буде відповідати подібним подіям та виконувати певні дії при їх виникненні. Як і властивості, події, на які можлива реакція, різні від компонента до компоненту. Подія (event) – це метод, що викликається у компоненті в результаті взаємодії цього компонента з користувачем.

Праворуч від інспектора об'єктів розташована робоча область (workspace). Тут спочатку розташовується редактор форм (Form Editor). Редактор форм дозволяє створювати форми. Форма може бути основним вікном програми, діалоговим вікном або вікном будь-якого іншого типу. У процесі створення форми за допомогою редактора форм у ній можна розміщувати компоненти, змінювати їхнє положення та розміри.

За редактором форм прихований редактор коду (Code Editor). У вікні редактора коду набирається текст програми. Інспектор об'єктів, редактор форм, редактор коду та палітра компонентів працюють взаємопов'язано під час побудови програми.

**РОЗДІЛ 3. РЕЗУЛЬТАТИ РОБОТИ ПРОГРАМИ**

Робота користувача починається з головного меню (Рис. 3.1), який містить каталог книг та реєстр читачів.

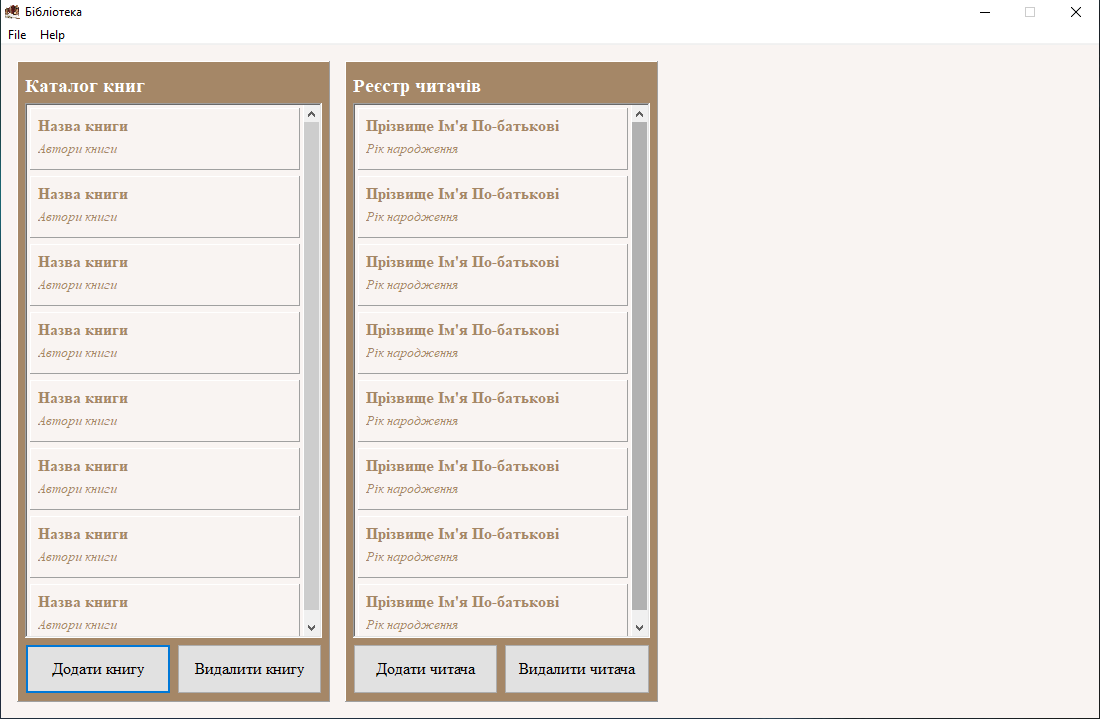


Рисунок 3.1 – Вигляд головної форми програми

При натисканні курсором миші на назву книги в каталозі у правій частині вікна відкривається інформація про цю книгу (Рис. 3.2). Аналогічно при натисканні курсором миші на ПІБ читача в реєстрі у правій частині вікна відкривається інформація про цього читача (Рис. 3.3). Нова книга додається до каталогу натисканням кнопки «Додати книгу». При цьому відкривається форма для введення інформації (Рис. 3.4). Аналогічним чином додається новий читач до реєстру (Рис. 3.5).

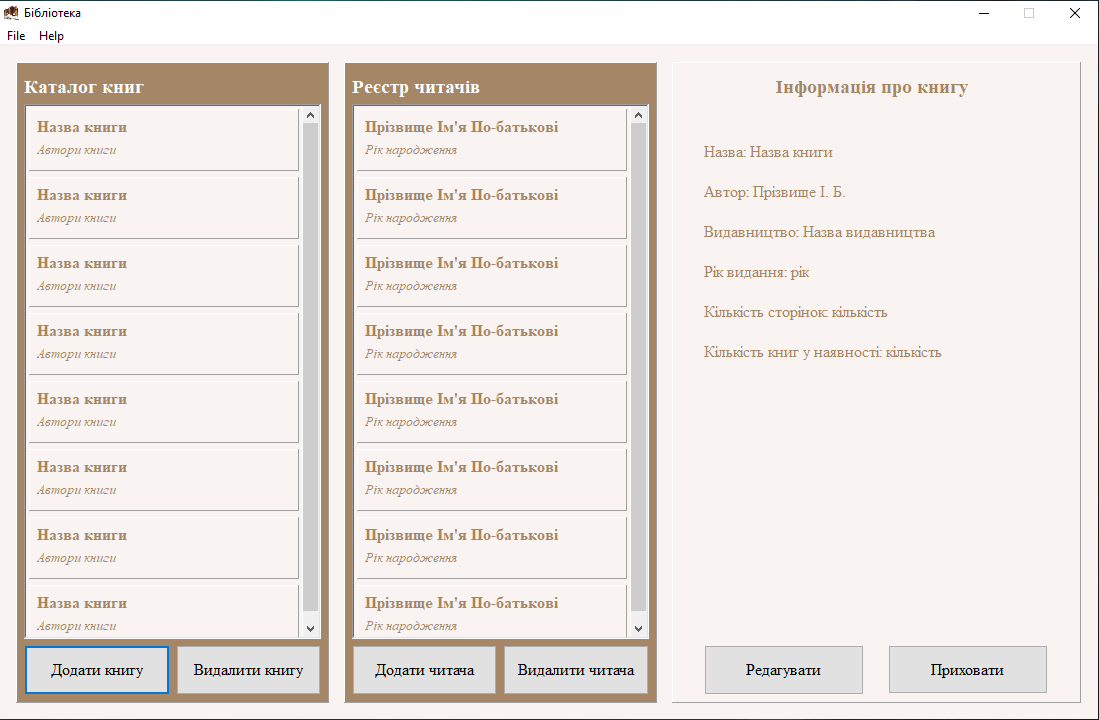


Рисунок 3.2 – Вигляд головної форми з інформацією про обрану книгу

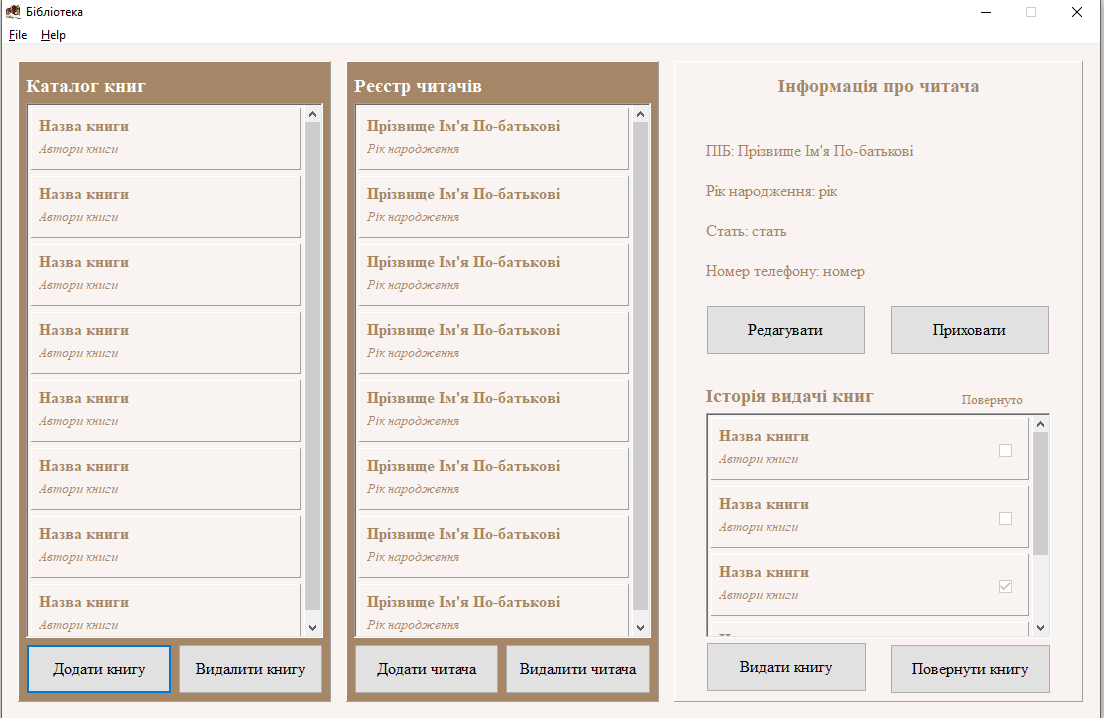


Рисунок 3.3 – Вигляд головної форми з інформацією про обраного читача

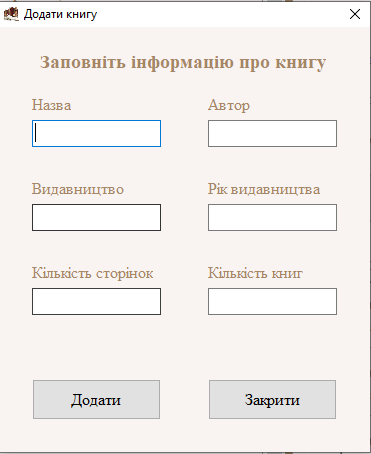


Рисунок 3.4 – Вигляд форми додавання книги до каталогу

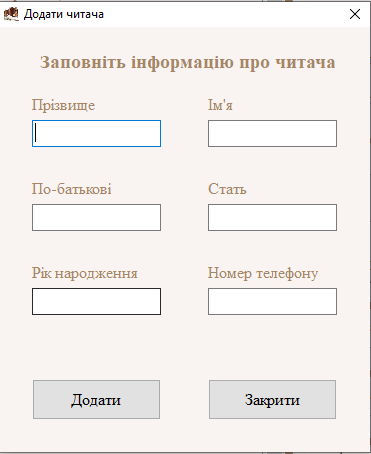


Рисунок 3.5 – Вигляд форми додавання читача до реєстру

**Висновок**

В рамках курсової роботи були визначені процеси, пов'язані з роботою бібліотек та підходи до їх автоматизації. Було розроблено графічний інтерфейс додатку для автоматизації робочого місця бібліотекаря.

Під час виконання курсової роботи були набуті навички застосування мови C++ та інтегрованого середовища розробки C++ Builder.

# **Список використаної літератури**

1. ГОСТ 2.105-95 ЕСКД Общие требования к текстовым документам.
2. ДСТУ 1.5-93 Загальні вимоги до побудови, викладу, оформлення та змісту.
3. C++ Builder. Введение в структурное программирование

URL: https://2dip.su/рефераты/1002122/

1. Начинаем работать в Borland C++ Builder

URL: http://www.codenet.ru/progr/bcb/borland\_builder/3.php

1. Wikipedia: C++.

URL: https://uk.wikipedia.org/wiki/C++

1. Введение в C++. Язык программирования C++

https://metanit.com/cpp/tutorial/1.1.php

# **Додаток 1**

**Код головної форми**

1. Файл *main\_form.cpp*

**#include <vcl.h>**

**#pragma hdrstop**

**#include "main\_form.h"**

**//---------------------------------------------------------------------------**

**#pragma package(smart\_init)**

**#pragma resource "\*.dfm"**

**TMainForm \*MainForm;**

**//---------------------------------------------------------------------------**

**\_\_fastcall TMainForm::TMainForm(TComponent\* Owner)**

**: TForm(Owner)**

**{**

**}**

**//---------------------------------------------------------------------------**

**void \_\_fastcall TMainForm::bookPanelClick(TObject \*Sender)**

**{**

**if (aboutReaderPanel -> Visible == true) {**

**aboutReaderPanel -> Visible = false;**

**}**

**aboutBookPanel -> Visible = true;**

**}**

**//---------------------------------------------------------------------------**

**void \_\_fastcall TMainForm::aboutBookHideButtonClick(TObject \*Sender)**

**{**

**aboutBookPanel -> Visible = false;**

**}**

**//---------------------------------------------------------------------------**

**void \_\_fastcall TMainForm::readerPanelClick(TObject \*Sender)**

**{**

**if (aboutBookPanel -> Visible == true) {**

**aboutBookPanel -> Visible = false;**

**}**

**aboutReaderPanel -> Visible = true;**

**}**

**//---------------------------------------------------------------------------**

**void \_\_fastcall TMainForm::aboutReaderHideButtonClick(TObject \*Sender)**

**{**

**aboutReaderPanel -> Visible = false;**

**}**

**//---------------------------------------------------------------------------**

**void \_\_fastcall TMainForm::addBookButtonClick(TObject \*Sender)**

**{**

**addBookForm -> ShowModal();**

**}**

**//---------------------------------------------------------------------------**

**void \_\_fastcall TMainForm::addReaderButtonClick(TObject \*Sender)**

**{**

**addReaderForm -> ShowModal();**

**}**

**//---------------------------------------------------------------------------**

**void \_\_fastcall TMainForm::N1Click(TObject \*Sender)**

**{**

**Close();**

**}**

2. Файл *main\_form.h*

**#ifndef main\_formH**

**#define main\_formH**

**//---------------------------------------------------------------------------**

**#include <System.Classes.hpp>**

**#include <Vcl.Controls.hpp>**

**#include <Vcl.StdCtrls.hpp>**

**#include <Vcl.Forms.hpp>**

**#include <Vcl.ExtCtrls.hpp>**

**#include <Vcl.Dialogs.hpp>**

**#include <Vcl.Menus.hpp>**

**#include "addBook\_form.h"**

**#include "addReader\_form.h"**

**//---------------------------------------------------------------------------**

**class TMainForm : public TForm**

**{**

**\_\_published: // IDE-managed Components**

**TPanel \*booksPanel;**

**TPanel \*readersPanel;**

**TPanel \*aboutReaderPanel;**

**TPanel \*aboutBookPanel;**

**TScrollBox \*ScrollBox1;**

**TScrollBox \*ScrollBox2;**

**TButton \*addBookButton;**

**TButton \*removeBookButton;**

**TButton \*addReaderButton;**

**TButton \*removeReaderButton;**

**TPanel \*bookPanel;**

**TLabel \*bookTitleLabel;**

**TLabel \*bookAuthorLabel;**

**TPanel \*readerPanel;**

**TLabel \*readerNameLabel;**

**TLabel \*readerYearLabel;**

**TLabel \*aboutBookHeaderLabel;**

**TLabel \*aboutBookTitleLabel;**

**TLabel \*aboutBookAuthorLabel;**

**TLabel \*aboutBookPblhouseLabel;**

**TLabel \*aboutBookPblyearLabel;**

**TLabel \*aboutBookQntpgLabel;**

**TLabel \*aboutBookQntavblLabel;**

**TButton \*aboutBookEditButton;**

**TButton \*aboutBookHideButton;**

**TLabel \*aboutReaderHeaderLabel;**

**TLabel \*aboutReaderNameLabel;**

**TLabel \*aboutReaderYearLabel;**

**TLabel \*aboutReaderGenderLabel;**

**TLabel \*aboutReaderPhoneLabel;**

**TButton \*aboutReaderEditButton;**

**TButton \*aboutReaderHideButton;**

**TScrollBox \*ScrollBox3;**

**TButton \*giveBookButton;**

**TButton \*returnBookButton;**

**TMainMenu \*MainMenu1;**

**TMenuItem \*File1;**

**TMenuItem \*Help1;**

**TMenuItem \*Help2;**

**TMenuItem \*N1;**

**TLabel \*booksHeaderLabel;**

**TLabel \*readersHeaderLabel;**

**TLabel \*aboutReaderHistoryHeaderLabel;**

**TLabel \*returnedLabel;**

**TPanel \*Panel1;**

**TLabel \*Label1;**

**TLabel \*Label2;**

**TCheckBox \*CheckBox1;**

**TPanel \*Panel2;**

**TLabel \*Label5;**

**TLabel \*Label6;**

**TCheckBox \*CheckBox2;**

**TPanel \*Panel3;**

**TLabel \*Label7;**

**TLabel \*Label8;**

**TCheckBox \*CheckBox3;**

**TPanel \*Panel4;**

**TLabel \*Label9;**

**TLabel \*Label10;**

**TCheckBox \*CheckBox4;**

**TPanel \*Panel5;**

**TLabel \*Label11;**

**TLabel \*Label12;**

**TCheckBox \*CheckBox5;**

**TPanel \*Panel6;**

**TLabel \*Label13;**

**TLabel \*Label14;**

**TPanel \*Panel7;**

**TLabel \*Label15;**

**TLabel \*Label16;**

**TPanel \*Panel8;**

**TLabel \*Label17;**

**TLabel \*Label18;**

**TPanel \*Panel9;**

**TLabel \*Label19;**

**TLabel \*Label20;**

**TPanel \*Panel10;**

**TLabel \*Label21;**

**TLabel \*Label22;**

**TPanel \*Panel11;**

**TLabel \*Label23;**

**TLabel \*Label24;**

**TPanel \*Panel12;**

**TLabel \*Label25;**

**TLabel \*Label26;**

**TPanel \*Panel13;**

**TLabel \*Label27;**

**TLabel \*Label28;**

**TPanel \*Panel14;**

**TLabel \*Label29;**

**TLabel \*Label30;**

**TPanel \*Panel15;**

**TLabel \*Label31;**

**TLabel \*Label32;**

**TPanel \*Panel16;**

**TLabel \*Label33;**

**TLabel \*Label34;**

**TPanel \*Panel17;**

**TLabel \*Label35;**

**TLabel \*Label36;**

**TPanel \*Panel18;**

**TLabel \*Label37;**

**TLabel \*Label38;**

**TPanel \*Panel19;**

**TLabel \*Label39;**

**TLabel \*Label40;**

**void \_\_fastcall bookPanelClick(TObject \*Sender);**

**void \_\_fastcall aboutBookHideButtonClick(TObject \*Sender);**

**void \_\_fastcall readerPanelClick(TObject \*Sender);**

**void \_\_fastcall aboutReaderHideButtonClick(TObject \*Sender);**

**void \_\_fastcall addBookButtonClick(TObject \*Sender);**

**void \_\_fastcall addReaderButtonClick(TObject \*Sender);**

**void \_\_fastcall N1Click(TObject \*Sender);**

**private: // User declarations**

**public: // User declarations**

**\_\_fastcall TMainForm(TComponent\* Owner);**

**};**

**//---------------------------------------------------------------------------**

**extern PACKAGE TMainForm \*MainForm;**

**//---------------------------------------------------------------------------**

**#endif**