## Міністерство освіти і науки України

## ФАКУЛЬТЕТ ІНФОРМАТИКИ

КАФЕДРА ІНФОРМАЦІЙНИХ УПРАВЛЯЮЧИХ СИСТЕМ ТА ТЕХНОЛОГІЙ

#### Реєстраційний №\_\_\_\_\_\_\_\_

Дата \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## КУРСОВА РОБОТА

**Тема:**

**Інтерактивна система навчання для вивчення англійської мови.**

#### Рекомендована до захисту

“\_\_\_\_” \_\_\_\_\_\_\_\_\_\_ 2007р.

Робота захищена

“\_\_\_\_” \_\_\_\_\_\_\_\_\_\_ 2007р.

з оцінкою

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Підписи членів комісії

**Зміст**

# Вступ

# Теорія

# Практична частина

# Висновки

# Література

**Вступ**

Головною метою нашої роботи, буде написання інтерактивної системи навчання вивчення англійської мови. В теоретичній частині наведемо основні відомості про інструментарій в якості якого нами була вибрана мова програмування Borland C++ Builder.

Завдяки засобам керування проектами, двосторонньої інтеграції додатка й синхронізації між засобами візуального й текстового редагування, а також вбудованому отладчіку (з ассемблерним вікном прокручування, покроковим виконанням, точками останова, трасуванням і т.п.) - C++ Builder корпорації Borland надає собою вражаюче середовище розробки, що, видимо, витримає конкурентну боротьбу з такими модними продуктами як Developer Studio фірми Microsoft.

**Теорія**

C++ Builder забезпечує високу швидкодію при компіляції й зборці 32-розрядних додатків для сучасних операційних систем Windows 95 й Windows NT, включаючи OLE взаємодія клієнт-сервер. Система навіть відображає час, витрачений на основні етапи побудови програм. Результуючі програми добре оптимізовані по швидкості виконання й витратам пам'яті. Хоча отладочний режим низького рівня повністю інтегрований у середовище C++Builder, до налагодження також довелося звикати. Дизайнер форм, Інспектор об'єктів й інших засобів залишаються доступними під час роботи програми, тому вносити зміни можна в процесі налагодження.

C++Builder поставляється в трьох варіантах: Standard (стандартний). Professional (для професіоналів розроблювачів, орієнтованих на мережну архітектуру) і Client/Server Suite (для розробки систем в архітектурі клієнт/сервер). Останні два варіанти доповнюють стандартний вихідними текстами візуальних компонентів, різномасштабним словником даних, новими функціями мови запитів SQL для баз даних, пакетом підтримки систем Internet, службою моніторингу програм, а також рядом інших засобів.

Експерименти з тестовими програмами в рамках стандартного варіанта лягли в основу матеріалу, що викладає в книзі. Випробовуючи систему, я переклав на C++ Builder кілька додатків, раніше написаних на Borland C++ версії 4.5. Завдяки візуальним компонентам, із програм зникла "кодова лушпайка" обробки повідомлень Windows і ресурсних файлів, і залишився тільки змістовний код. Користувальницький інтерфейс додатків придбав закінчений професійний вигляд.

Хоча C++ Builder представляється досить надійною системою, корпорації ще має бути спростувати розхоже твердження, що в кожній налагодженій програмі (у тому числі й у комерційній) є щонайменше одна помилка. Видимо, саме цим прагненням порозумівається зайва, на мій погляд, поспішність із рекламуванням "поліпшеної й розширеної" версії Borland C++ версії 5.02.

C++ Builder підтримує зв'язок з різними базами даних 3-х видів:

dBASE й Paradox: Sybase, Oracle, InterBase й Informix; Excel, Access, FoxPro й Btrieve. Механізм BDE (Borland Database Engine) надає обслуговуванню зв'язків з базами даних дивну простоту й прозорість. Провідник Database Explorer дозволяє зображувати зв'язки й об'єкти баз даних графічно. Використовуючи компоненти баз даних, я побудував електронну записну книжку по таблиці dBASE за півгодини роботи на комп'ютері. Спадкування готових форм й їх "припасування" під специфічні вимоги помітно скорочують тимчасові витрати на рішення подібних задач.

Довідкова служба C++ Builder надавала мені допомогу в цій і багатьох інших подібних ситуаціях. Є повний опис кожного керуючого компонента, включаючи списки властивостей і методів, а також численні приклади. Виклад матеріалу в книзі було значно поліпшене й систематизовано завдяки відомостям, почерпнутим мною з довідкової служби.

# Застосування BORLAND C++ BUILDER для створення ігрових програм:

# Borland C++ Builder - випущене недавно компанією Borland засіб швидкої розробки проектів, що дозволяє створювати проекту мовою C++, використовуючи при цьому середовище розробки та бібліотеку компонентів Delphi. У даній частині роботи розглядається середовище розробки C++ Builder та основні прийоми, застосовувані при проектуванні користувальницького інтерфейсу.

### Середовище розробки C++ Builder

C++ Builder являє собою SDI-додаток, головне вікно якого містить інструментальну панель, що набудоване (ліворуч) і палітру компонентів (праворуч). Крім цього, при запуску C++ Builder з'являються вікно інспектора об'єктів (ліворуч) і форма нового проекту (праворуч). Під вікном форми проекту перебуває вікно редактора коду.

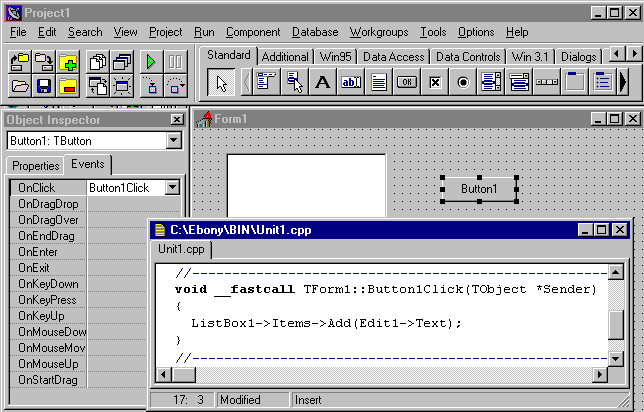


Рис.1. Середовище розробки C++ Builder

Форми є основою проектів C++ Builder. Створення користувальницького інтерфейсу проекту полягає в додаванні у вікно форми елементів об'єктів C++ Builder, називаних компонентами. Компоненти C++ Builder розташовуються на палітрі компонентів, виконаної у вигляді многостранічного блокнота. Важлива особливість C++ Builder полягає в тому, що він дозволяє створювати власні компоненти та набудовувати палітру компонентів, а також створювати різні версії палітри компонентів для різних проектів.

### Компоненти C++ Builder

Компоненти розділяються на видимі (візуальні) і невидимі (невізуальні). Візуальні компоненти з'являються під час виконання точно так само, як і під час проектування. Прикладами є кнопки та редагують поля, що. Невізуальні компоненти з'являються під час проектування як піктограми на формі. Вони ніколи не видні під час виконання, але мають певну функціональність (наприклад, забезпечують доступ до даних, викликають стандартні діалоги Windows 95та ін.)

Для додавання компонента у форму можна вибрати мишею потрібний компонент у палітрі та клацнути лівою клавішею миші в потрібнім місці проектованої форми. Компонент з'явиться на формі, і далі його можна переміщати, міняти розміри та інші характеристики.

Кожен компонент C++ Builder має три різновиди характеристик: властивості, події та методи.

Якщо вибрати компонент із палітри та додати його до форми, інспектор об'єктів автоматично покаже властивості та події, які можуть бути використані з тім компонентом. У верхній частині інспектора об'єктів є список, що випадає, що дозволяє вибирати потрібний об'єкт із наявних на формі.

### Властивості компонентів

Властивості є атрибутами компонента, що визначають його зовнішній вигляд і поводження. Багато властивостей компонента в колонку властивостей мають значення, яке встановлюється за замовчуванням (наприклад, висота кнопок). Властивості компонента відображаються а сторінці властивостей (Properties). Інспектор об'єктів відображає опубліковані (published) властивості компонентів. Крім published-властивостей, компоненти можуть і найчастіше мають загальні (public), опубліковані властивості, які доступні тільки під час виконання проекту. Інспектор об'єктів використається для установки властивостей під час проектування. Список властивостей розташовується на сторінці властивостей інспектори об'єктів. Можна визначити властивості під час проектування або написати код для видозміни властивостей компонента під час виконання проекту.

При визначенні властивостей компонента під час проектування потрібно вибрати компонент на формі, відкрити сторінку властивостей в інспекторі об'єктів, вибрати обумовлена властивість і змінити його за допомогою редактора властивостей (це може бути пусте поле для уведення тексту або числа, що випадає список, що розкривається список, діалогова панель і т.д.).

### Події

Сторінка подій (Events) інспектори об'єктів показує список подій, розпізнаваних компонентом (програмування для операційних систем із графічним користувальницьким інтерфейсом, зокрема, для Windows 95 або Windows NT пре думає опис реакції проекту на ті або інші події, а сама операційна система займається постійним опитуванням комп'ютера з метою виявлення настання якої-небудь події). Кожен компонент має свій власний набір оброблювачів подій. В C++ Builder варто писати функції, називані оброблювачами подій, і зв'язувати події із цими функціями. Створюючи оброблювач тої чи іншої події, ви доручаєте програмі виконати написану функцію, якщо ця подія відбудеться.

Для того, щоб додати оброблювач подій, потрібно вибрати на формі за допомогою миші компонент, якому необхідний оброблювач подій, потім відкрити сторінку подій інспектори об'єктів і двічі клацнути лівою клавішею миші на колонку з поруч із подією, щоб змусити C++ Builder згенерувати прототип оброблювача подій і показати його в редакторі коду. При цьому автоматично генерується текст порожньої функції, і редактор відкривається в тім місці, де варто вводити код. Курсор позиціюється усередині операторних дужок { ... }. Далі потрібно ввести код, що повинен виконуватися при настанні події. Оброблювач подій може мати параметри, які вказуються після імені функції в круглих дужках.

### Методи

Метод є функцією, що пов'язана з компонентом, і яка оголошується як частина об'єкта. Створюючи оброблювачі подій, можна викликати методи, використовуючи наступну нотацію: ->, наприклад:

Edit1->Show();

Відзначимо, що при створенні форми пов'язані з нею модуль і заголовний файл із розширенням \*.h генеруються обов'язково, тоді як при створенні нового модуля він не зобов'язаний бути пов'язаний з формою (наприклад, якщо в ньому втримуються процедури розрахунків). Імена форми та модуля можна змінити, причому бажано зробити це відразу після створення, поки на них не з'явилося багато посилань в інших формах і модулях.

### Менеджер проектів

Файли, що утворять додаток – форми та модулі - зібрані в проект. Менеджер проектів показує списки файлів і модулів проекту та дозволяє створювати навігацію між ними. Можна викликати менеджер проектів , вибравши пункт меню View/Project Manager. За замовчуванням знову створений проект одержує ім'я Project1.cpp.

За замовчуванням проект спочатку містить файли для однієї форми та вихідного коду одного модуля. Однак більшість проектів містять кілька форм і модулів. Щоб додати модуль або форму до проекту, потрібно клацнути правою кнопкою миші та вибрати пункт New Form з контекстного меню. Можна також додавати існуючі форми та модулі до проекту, використовуючи кнопку Add контекстного меню менеджера проектів і вибираючи модуль або форму, яку потрібно додати. Форми та модулі можна видалити в будь-який момент протягом розробки проекту. Однак, через те, що форма зв'язані завжди з модулем, не можна видалити одне без видалення іншого, за винятком случаючи, коли модуль не має зв'язку з формою. Видалити модуль із проекту можна, використовуючи кнопку Remove менеджера проектів.

Якщо вибрати кнопку Options у менеджері проектів, відкриється діалогова панель опцій проекту, у якій можна вибрати головну форму проекту, визначити, які форми будуть створюватися динамічно, які параметри компіляції модулів (у тому числі створених в Delphi 2.0, тому що C++ Builder може включати їх у проекти) і компонування.

Важливим елементом середовища розробки C++ Builder є контекстне меню, яке з‘являється при натисканні на праву клавішу миші та дозволяє швидкий доступ до найбільше часто використовуваних команд.

Зрозуміло, C++ Builder має убудовану систему контекстно-контекстно-залежної допомоги, доступної для будь-якого елемента інтерфейсу та довідкової інформації, що є великим джерелом, про C++ Builder.

### Створення проектів у С++ Builder

Першим кроком у розробці проекту C++ Builder є створення проекту. Файли проекту містять згенерований автоматично вихідний текст, що стає частиною проекту, коли воно скомпільовано та підготовлене до виконання. Щоб створити новий проект, потрібно вибрати пункт меню File/New Application.

C++ Builder створює файл проекту з ім'ям за замовчуванням Project1.cpp, а також make-файл із ім'ям за замовчуванням Project1.mak. При внесенні змін у проект, таких, як додавання нової форми, C++ Builder обновляє файл проекту.

Проект або додаток звичайно мають кілька форм. Додавання форми до проекту створює наступні додаткові файли:

* Файл форми з розширеням.DFM, що містить інформацію про ресурси вікон для конструювання форми
* Файл модуля з розширеням.CPP, що містить код на C++.
* Заголовний файл із розширенням .H, що містить опис класу форми.

Коли ви додаєте нову форму, файл проекту автоматично обновляється.

Для того щоб додати одну або більше форм до проекту , виберіть пункт меню File/New Form. З'явиться порожня форма, що буде додана до проекту. Можна скористатися пунктом меню File/New, вибрати сторінку Forms і вибрати підходящий шаблон з репозиторія об'єктів.

Для того, щоб просто відкомпілювати поточний проект, з меню Compile потрібно вибрати пункт меню Compile. Для того, щоб відкомпілювати проект і створити виконує файл, що, для поточного проекту, з меню Run потрібно вибрати пункт меню Run. Компонування проекту є інкрементним (перекомпілюються тільки модулі, що змінилися).

Якщо при виконанні проекту виникає помилка часу виконання, C++ Builder робить паузу у виконанні програми та показує редактор коду з курсором, установленим на операторі, що є джерелом помилки. Перш ніж робити необхідну корекцію, варто запустити знову додаток, вибираючи пункт меню Run з контекстного меню або з меню Run, закрити додаток і лише потім вносити зміни в проект. У цьому випадку зменшиться ймовірність втрати ресурсів Windows.

### Приклад: створення найпростішого проекту

Тепер спробуємо створити найпростіший додаток, що дозволяє вводити текст у редагує поле, що, і додавати цей текст до списку при натисканні мишею на кнопку. Виберемо пункт меню File/New Application для створення проектутазбережемо його головну форму під ім'ям samp1.cpp, а сам проект під ім'ям samp.mak. Помістимо на форму компонента Button, EditтаListBox зі сторінки Standard палітри компонентів.

Після цього виберемо на формі компонентів Edit і видалимо поточне значення властивості Text. Потім установимо властивість Caption для Button1 рівним "Додати".

Щоб додати оброблювач події OnClick для кнопки Додати, потрібно вибрати цю кнопку на формі, відкрити сторінку подій в інспекторі об'єктів і двічі клацнути мишею на колонку праворуч від події OnClick. У відповідному рядку уведення з'явиться ім'я функції. C++ Builder згенерує прототип оброблювача подій і покаже його в редакторі коду. Після цього варто ввести наступний код в операторні дужки { ... } тіла функції:

void \_\_fastcall TForm1::Button1Click(TObject \*Sender)

{

if (!(Edit1->Text == ""))

{

ListBox1->Items->Add(Edit1->Text);

Edit1->Text = "" ;

}

}

Для компіляції проекту в меню Run виберемо пункт Run. Тепер можна що-небудь увести в редагує поле, що, нажати мишею на кнопку Додати та переконатися, що вводять строки, що, додаються до списку.

Тепер модифікуємо додаток, додавши кнопки Видалити та Вихід. Для цього додамо ще дві кнопки, змінимо їхню властивість Caption і створимо оброблювачі подій, пов'язаних з натисканням на ці кнопки:

Для кнопки Видалити:

void \_\_fastcall TForm1::Button2Click(TObject \*Sender)

{

if (!(ListBox1->ItemIndex == -1))

ListBox1->Items->Delete(ListBox1->ItemIndex);

}

Для кнопки Вихід:

Close();

Збережемо та скомпілюємо додаток, а потім протестуємо його.

Отже, ми познайомилися із середовищем розробки Borland C++ Builder і створили простий додаток.

**Практична частина**

Так як середою розробки був вибраний інструментарій C++ Builder, то зрозуміло, що програма буде мати ряд особливостей, серед, яких слід відзначити наступні: програма являє собою єдиний модуль, для спрощення розробки був використаний лише текстовий інтерфейс, та розроблена програма має значний об‘єм.

У зв‘язку з останнім наведемо лише фрагмент коду головного модуля. (загальний лістинг розробленої програми знаходиться на дискеті разом з роботою)

Лістинг. (angl.cpp)

#include <iostream.h>

#include <dos.h>

#include <conio.h>

#include <string.h>

const char numberOfAnswer[] = "\nPress 5 for 50/50, 6 phone friend, 7 walk away.\n\n";

const char numberOfAnswer2[] = "Please enter the number of your choice: ";

const char people1[] = "John";

const char people2[] = "Mary";

const char people3[] = "Paul";

const char people4[] = "Kathy";

const char people5[] = "Greg";

const char people6[] = "Tracy";

const char people7[] = "Peter";

const char people8[] = "Lara";

const char people9[] = "Jason";

const char people0[] = "April";

const char people01[] = "Jeremy";

const char people02[] = "Paula";

const char people03[] = "Chris";

const char people04[] = "Jennifer";

const char people05[] = "Steve";

const char goodAnswer1[] = "\n\nCorrect!\n\n";

const char goodAnswer2[] = "\n\nGreat Job!!\n\n";

const char goodAnswer3[] = "\n\nKeep up the good work!\n\n";

const char goodAnswer4[] = "\n\nThat's right!!\n\n";

const char goodAnswer5[] = "\n\nWonderful answer!!!\n\n";

const char goodAnswer6[] = "\n\nRight!\n\n";

const char goodAnswer7[] = "\n\nYou must have studied. That's the right answer!\n\n";

const char goodAnswer8[] = "\n\nVery good!!!\n\n";

const char goodAnswer9[] = "\n\nYou're absolutely, positively correct!\n\n";

const char goodAnswer0[] = "\n\nWow!!\n\n";

const char goodAnswer01[] = "\n\nExcellent!\n\n";

const char goodAnswer02[] = "\n\nFabulous!\n\n";

const char goodAnswer03[] = "\n\nRight on!\n\n";

const char goodAnswer04[] = "\n\nYou're a genius!\n\n";

const char goodAnswer05[] = "\n\nSplendid!\n\n";

const char badAnswer1[] = "\n\nI'm sorry, that's the wrong answer. :( \n\n";

const char badAnswer2[] = "\n\nNope! That's not it!!!\n\n";

const char badAnswer3[] = "\n\nHahaha. You call that an answer?\n\n";

const char badAnswer4[] = "\n\nWrong.\n\n";

const char badAnswer5[] = "\n\nI'm ashamed of you picking that wrong answer!\n\n";

const char badAnswer6[] = "\n\nCome on! I know you're smarter than that! :( \n\n";

const char badAnswer7[] = "\n\nMaybe you should quit.\n\n";

const char badAnswer8[] = "\n\nWell the good news is there's another question ahead!\n\n";

const char badAnswer9[] = "\n\nI hope you make a better choice next time!\n\n";

const char badAnswer0[] = "\n\nI don't think so.\n\n";

const char badAnswer01[] = "\n\nThat ain't it.\n\n";

const char badAnswer02[] = "\n\nAnother wrong answer.\n\n";

const char badAnswer03[] = "\n\nDon't do this to me! Wrong answer. :( \n\n";

const char badAnswer04[] = "\n\nYou scare me. Wrong!\n\n";

const char badAnswer05[] = "\n\nNo no no! Not that one.\n\n";

const char question1[] = "What color is a dollar bill?\n";

const char question2[] = "If you have 3 quarters, you have...\n";

const char question3[] = "George Washington cut down what kind of tree?\n";

const char question4[] = "If A = B and B = C, then...\n";

const char question5[] = "Complete the phrase: Into the valley rode the...\n";

const char question6[] = "Who always saved Timmy?\n";

const char question7[] = "Who's first gold record was Hard Headed Woman?\n";

const char question8[] = "Who was author of Tom Sawyer and Huckleberry Finn?\n";

const char question9[] = "What was Shirley Temple's real life last name?\n";

const char question0[] = "Who plays the Priest in Anne of Green Gables?\n";

const char question01[] = "If a Sirloin Steak is cooked rare, it is generally what color in the middle?\n";

const char question02[] = "If Coca Cola is applied to an egg shell, the shell turns what color?\n";

const char question03[] = "What is the sum of 30 times 30 plus 90 plus 25?\n";

const char question04[] = "Who is the voice of Bart Simpson?\n";

const char question05[] = "Who was the original Clarke Kent?\n";

int yourAnswer;

int yourAnswer1;

char yourName[128];

int counter = 0;

int counter2 = 0;

int correctAnswers = 0;

unsigned long int score;

int walkAway();

void continueProgram();

int playMillionaire();

int fiftyFiftyUsed;

int phoneFriendUsed;

void answer1();

void answer2();

void answer3();

void answer4();

void answer5();

void answer6();

void answer7();

void answer8();

void answer9();

void answer0();

void answer01();

void answer02();

void answer03();

void answer04();

void answer05();

// Beginning of 50/50 functions

void answer1()

{

clrscr(); // Clears the screen

cout << question1; // Prints question1, defined above, to the screen

cout << " 2. Purple\n";

cout << "3. Green \n";

cout << numberOfAnswer2;

cin >> yourAnswer;

if (yourAnswer == 3)

{

cout << goodAnswer1;

delay(3000);

correctAnswers++; // Increases correctAnswers by one

score = 100; // Sets score to 100

}

if (yourAnswer != 3)

{

cout << badAnswer1;

delay(3000);

score = 0;

walkAway();

}

}

void answer2()

{

clrscr();

cout << question2;

cout << "1. $.75 \n";

cout << " 4. $1.00\n";

cout << numberOfAnswer2;

cin >> yourAnswer;

if (yourAnswer == 1)

{

cout << goodAnswer2;

delay(3000);

correctAnswers++;

score = 200;

}

if (yourAnswer != 1)

{

cout << badAnswer2;

delay(3000);

score = 0;

walkAway();

}

}

void answer3()

{

clrscr();

cout << question3;

cout << " 2. Cherry\n";

cout << " 4. Magnolia\n";

cout << numberOfAnswer2;

cin >> yourAnswer;

if (yourAnswer == 2)

{

cout << goodAnswer3;

delay(3000);

correctAnswers++;

score = 300;

}

if (yourAnswer != 2)

{

cout << badAnswer3;

delay(3000);

score = 0;

walkAway();

}

}

void answer4()

{

clrscr();

cout << question1;

cout << "1. A = B + C \n";

cout << " 4. A = C \n";

cout << numberOfAnswer2;

cin >> yourAnswer;

if (yourAnswer == 4)

{

cout << goodAnswer4;

delay(3000);

correctAnswers++;

score = 500;

}

if (yourAnswer != 3)

{

cout << badAnswer4;

delay(3000);

score = 0;

walkAway();

}

}

void answer5()

{

clrscr();

cout << question5;

cout << "1. 600 \n";

cout << "3. 300 \n";

cout << numberOfAnswer2;

cin >> yourAnswer;

if (yourAnswer == 1)

{

cout << goodAnswer5;

delay(3000);

correctAnswers++;

score = 1000;

}

if (yourAnswer != 1)

{

cout << badAnswer5;

delay(3000);

score = 0;

walkAway();

}

}

void answer6()

{

clrscr();

cout << question6;

cout << "1. Lassie 2. Old Yeller\n";

cout << " \n";

cout << numberOfAnswer2;

cin >> yourAnswer;

if (yourAnswer == 1)

{

cout << goodAnswer6;

delay(3000);

correctAnswers++;

score = 2000;

}

if (yourAnswer != 1)

{

cout << badAnswer6;

delay(3000);

score = 1000;

walkAway();

}

}

void answer7()

{

clrscr();

cout << question7;

cout << "1. Jerry Lee Lewis \n";

cout << " 4. Elvis Presley\n";

cout << numberOfAnswer2;

cin >> yourAnswer;

if (yourAnswer == 4)

{

cout << goodAnswer7;

delay(3000);

correctAnswers++;

score = 4000;

}

if (yourAnswer != 1)

{

cout << badAnswer7;

delay(3000);

score = 1000;

walkAway();

}

}

void answer8()

{

clrscr();

cout << question8;

cout << " 2. Mark Twain\n";

cout << " 4. Edward Furlong\n";

cout << numberOfAnswer2;

cin >> yourAnswer;

if (yourAnswer == 2)

{

cout << goodAnswer8;

delay(3000);

correctAnswers++;

score = 8000;

}

if (yourAnswer != 2)

{

cout << badAnswer8;

delay(3000);

score = 1000;

walkAway();

}

}

void answer9()

{

clrscr();

cout << question9;

cout << "1. Black 2. Temple\n";

cout << " \n";

cout << numberOfAnswer2;

cin >> yourAnswer;

if (yourAnswer == 1)

{

cout << goodAnswer9;

delay(3000);

correctAnswers++;

score = 16000;

}

if (yourAnswer != 1)

{

cout << badAnswer9;

delay(3000);

score = 1000;

walkAway();

}

}

void answer0()

{

clrscr();

cout << question0;

cout << "1. Gilbert Blithe \n";

cout << "3. Sedrick Smith \n";

cout << numberOfAnswer2;

cin >> yourAnswer;

if (yourAnswer == 3)

{

cout << goodAnswer0;

delay(3000);

correctAnswers++;

score = 32000;

}

if (yourAnswer != 3)

{

cout << badAnswer0;

delay(3000);

score = 1000;

walkAway();

}

}

void answer01()

{

clrscr();

cout << question01;

cout << " 2. Black\n";

cout << "3. Red \n";

cout << numberOfAnswer2;

cin >> yourAnswer;

if (yourAnswer == 3)

{

cout << goodAnswer01;

delay(3000);

correctAnswers++;

score = 64000;

}

if (yourAnswer != 3)

{

cout << badAnswer01;

delay(3000);

score = 32000;

walkAway();

}

}

void answer02()

{

clrscr();

cout << question02;

cout << "1. Brown \n";

cout << "3. White \n";

cout << numberOfAnswer2;

cin >> yourAnswer;

if (yourAnswer == 1)

{

cout << goodAnswer02;

delay(3000);

correctAnswers++;

score = 125000;

}

if (yourAnswer != 1)

{

cout << badAnswer02;

delay(3000);

score = 32000;

walkAway();

}

}

void answer03()

{

clrscr();

cout << question03;

cout << " \n";

cout << "3. 1025 4. 1015\n";

cout << numberOfAnswer2;

cin >> yourAnswer;

if (yourAnswer == 4)

{

cout << goodAnswer03;

delay(3000);

correctAnswers++;

score = 250000;

}

if (yourAnswer != 4)

{

cout << badAnswer03;

delay(3000);

score = 32000;

walkAway();

}

}

void answer04()

{

clrscr();

cout << question04;

cout << " 2. Nancy Cartwright\n";

cout << "3. Nancy Drew \n";

cout << numberOfAnswer2;

cin >> yourAnswer;

if (yourAnswer == 2)

{

cout << goodAnswer04;

delay(3000);

correctAnswers++;

score = 500000;

}

if (yourAnswer != 2)

{

cout << badAnswer04;

delay(3000);

score = 32000;

walkAway();

}

}

void answer05()

{

clrscr();

cout << question05;

cout << "1. Christopher Reeves \n";

cout << "3. George Reeves \n";

cout << numberOfAnswer2;

cin >> yourAnswer;

if (yourAnswer == 3)

{

cout << goodAnswer05;

delay(3000);

correctAnswers++;

score = 1000000;

}

if (yourAnswer != 3)

{

cout << badAnswer05;

delay(3000);

score = 32000;

walkAway();

}

}

// End of 50/50 functions

int walkAway() // Decides how the program will end

{

delay(3000); // Pauses for 3 seconds

if (correctAnswers >= 1 && correctAnswers < 15)

{

cout << "\n\n" << yourName <<", you answered " << correctAnswers << "\n"; // Prints how many questions answered of

cout << "out of " << counter << " attempted questions.\n"; // how many attempted to the screen

cout << "You've won " << "$" << score << "!";

}

if (correctAnswers < 1) // Executes if correctAnswers is less than 1

{

cout << "\n\nOh well, maybe next time.\n\n"; // Prints to screen

score = 0;

cout << "You won " << "$" << score << ". :( \n\n";

}

if (correctAnswers == 15) // Executes if correctAnswers is equivalent to 10

{

while (counter2 < 10)

{

counter2++;

clrscr();

delay(300);

cout << "1,000,000 1,000,000 1,000,000 1,000,000 1,000,000 1,000,000 1,000,000 1,000,000\n";

cout << "1,000,000 1,000,000 1,000,000 1,000,000 1,000,000 1,000,000 1,000,000 1,000,000\n";

cout << "1,000,000 1,000,000 1,000,000 1,000,000 1,000,000 1,000,000 1,000,000 1,000,000\n";

cout << "1,000,000 1,000,000 1,000,000 1,000,000 1,000,000 1,000,000 1,000,000 1,000,000\n";

cout << "1,000,000 1,000,000 1,000,000 1,000,000 1,000,000 1,000,000 1,000,000 1,000,000\n";

cout << "1,000,000 1,000,000 1,000,000 1,000,000 1,000,000 1,000,000 1,000,000 1,000,000\n";

cout << "1,000,000 1,000,000 1,000,000 1,000,000 1,000,000 1,000,000 1,000,000 1,000,000\n";

cout << "1,000,000 1,000,000 1,000,000 1,000,000 1,000,000 1,000,000 1,000,000 1,000,000\n";

cout << "1,000,000 1,000,000 1,000,000 1,000,000 1,000,000 1,000,000 1,000,000 1,000,000\n";

cout << "1,000,000 1,000,000 1,000,000 1,000,000 1,000,000 1,000,000 1,000,000 1,000,000\n";

cout << "1,000,000 1,000,000 1,000,000 1,000,000 1,000,000 1,000,000 1,000,000 1,000,000\n";

cout << "1,000,000 1,000,000 1,000,000 1,000,000 1,000,000 1,000,000 1,000,000 1,000,000\n";

cout << "1,000,000 1,000,000 1,000,000 1,000,000 1,000,000 1,000,000 1,000,000 1,000,000\n";

cout << "1,000,000 1,000,000 1,000,000 1,000,000 1,000,000 1,000,000 1,000,000 1,000,000\n";

cout << "1,000,000 1,000,000 1,000,000 1,000,000 1,000,000 1,000,000 1,000,000 1,000,000\n";

cout << "1,000,000 1,000,000 1,000,000 1,000,000 1,000,000 1,000,000 1,000,000 1,000,000\n";

cout << "1,000,000 1,000,000 1,000,000 1,000,000 1,000,000 1,000,000 1,000,000 1,000,000\n";

cout << "1,000,000 1,000,000 1,000,000 1,000,000 1,000,000 1,000,000 1,000,000 1,000,000\n";

cout << "1,000,000 1,000,000 1,000,000 1,000,000 1,000,000 1,000,000 1,000,000 1,000,000\n";

cout << "1,000,000 1,000,000 1,000,000 1,000,000 1,000,000 1,000,000 1,000,000 1,000,000\n";

cout << "1,000,000 1,000,000 1,000,000 1,000,000 1,000,000 1,000,000 1,000,000 1,000,000\n";

cout << "1,000,000 1,000,000 1,000,000 1,000,000 1,000,000 1,000,000 1,000,000 1,000,000\n";

cout << "1,000,000 1,000,000 1,000,000 1,000,000 1,000,000 1,000,000 1,000,000 1,000,000\n";

cout << "1,000,000 1,000,000 1,000,000 1,000,000 1,000,000 1,000,000 1,000,000 1,000,000\n";

cout << "1,000,000 1,000,000 1,000,000 1,000,000 1,000,000 1,000,000 1,000,000 1,000,000\n";

delay(300);

clrscr();

delay(300);

counter2++; // Increases counter2 by one

cout << "$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$\n";

cout << "$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$\n";

cout << "$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$\n";

cout << "$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$\n";

cout << "$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$\n";

cout << "$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$\n";

cout << "$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$\n";

cout << "$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$\n";

cout << "$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$\n";

cout << "$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$\n";

cout << "$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$\n";

cout << "$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$\n";

cout << "$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$\n";

cout << "$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$\n";

cout << "$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$\n";

cout << "$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$\n";

cout << "$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$\n";

cout << "$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$\n";

cout << "$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$\n";

cout << "$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$\n";

cout << "$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$\n";

cout << "$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$\n";

cout << "$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$\n";

cout << "$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$\n";

cout << "$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$\n";

delay(300);

clrscr();

delay(300);

}

cout << "Congradulations, " << yourName << ".\n";

cout << "You've won $1,000,000!!!\n\n";

}

cout << "\n\nTaken from Television Series 'Who Wants To Be A Millionaire'\n"; // Prints to screen

cout << "Programmed by Gordon C. December 2001\n\n"; // in any case

return 0;

}

void continueProgram() // Gives user option to continue or quit

{

delay(5000);

clrscr(); // Clears the screen

cout << "Please enter your FIRST name ONLY: ";

cin >> yourName;

delay(500);

clrscr();

cout << "Hello, " << yourName << ". Welcome to...\n";

delay(2000);

cout << "3. 1025 4. 1015\n";

cout << numberOfAnswer;

cout << numberOfAnswer2;

cin >> yourAnswer;

if (yourAnswer == 4)

{

cout << goodAnswer03;

delay(3000);

correctAnswers++;

score = 250000;

}

if (yourAnswer == 1 || yourAnswer == 2 || yourAnswer == 3)

{

cout << badAnswer03;

delay(3000);

score = 32000;

walkAway();

return 0;

}

if (yourAnswer == 5)

{

if (fiftyFiftyUsed == 1)

{

cout << "\nSorry 50/50 has already been used!\n";

delay(3000);

clrscr();

goto loop02;

}

if (fiftyFiftyUsed != 1)

{

fiftyFiftyUsed = 1;

answer03();

if (yourAnswer != 4)

{

score = 32000;

return 0;

}

}

}

if (yourAnswer == 6)

{

if (phoneFriendUsed == 1)

{

cout << "Sorry, phone a friend has been used!\n";

delay(3000);

clrscr();

goto loop02;

}

if (phoneFriendUsed != 1)

{

phoneFriendUsed = 1;

delay(2000);

cout << "Dialing...\n";

delay(2000);

cout << "You have been connected to " << people03 << ".\n";

cout << people03 << ",\n" << question03 << "\n";

cout << "1. 1035 2. 1075\n";

cout << "3. 1025 4. 1015\n";

delay(2000);

cout << "\n\n" << people03 << ": " << "I'm no math expert. I couldn't tell ya. Sorry.\n";

cout << numberOfAnswer2;

cin >> yourAnswer;

if (yourAnswer == 4)

{

cout << goodAnswer03;

delay(3000);

correctAnswers++;

score = 250000;

}

if (yourAnswer != 4)

{

cout << badAnswer03;

delay(3000);

score = 32000;

walkAway();

return 0;

}

if (yourAnswer == 7)

{

score = 32000;

walkAway();

return 0;

}

if (yourAnswer < 1 || yourAnswer > 7)

{

cout << badAnswer03;

delay(3000);

score = 32000;

walkAway();

return 0;

}

}

}

if (yourAnswer == 7)

{

score = 32000;

walkAway();

return 0;

}

if (yourAnswer != 1 && yourAnswer != 2 && yourAnswer != 3 && yourAnswer != 4 && yourAnswer != 5 && yourAnswer != 6 && yourAnswer != 7)

{

score = 32000;

walkAway();

return 0;

}

clrscr();

counter++;

cout << "Try for $500000\n\n";

delay(2000);

clrscr();

loop03:

cout << question04;

delay(3000);

cout << "1. Nancy Sinatra 2. Nancy Cartwright\n";

delay(1000);

cout << "3. Nancy Drew 4. Nancy Reagan\n";

cout << numberOfAnswer;

cout << numberOfAnswer2;

cin >> yourAnswer;

if (yourAnswer == 2)

{

cout << goodAnswer04;

delay(3000);

correctAnswers++;

score = 500000;

}

if (yourAnswer == 1 || yourAnswer == 3 || yourAnswer == 4)

{

cout << badAnswer04;

delay(3000);

score = 32000;

walkAway();

return 0;

}

if (yourAnswer == 5)

{

if (fiftyFiftyUsed == 1)

{

cout << "\nSorry 50/50 has already been used!\n";

delay(3000);

clrscr();

goto loop03;

}

if (fiftyFiftyUsed != 1)

{

fiftyFiftyUsed = 1;

answer04();

if (yourAnswer != 2)

{

score = 32000;

return 0;

}

}

}

if (yourAnswer == 6)

{

if (phoneFriendUsed == 1)

{

cout << "Sorry, phone a friend has been used!\n";

delay(3000);

clrscr();

goto loop03;

}

if (phoneFriendUsed != 1)

{

phoneFriendUsed = 1;

delay(2000);

cout << "Dialing...\n";

delay(2000);

cout << "You have been connected to " << people04 << ".\n";

cout << people04 << ",\n" << question04 << "\n";

cout << "1. Nancy Sinatra 2. Nancy Cartwright\n";

cout << "3. Nancy Drew 4. Nancy Reagan\n";

delay(2000);

cout << "\n\n" << people04 << ": " << "I love that show! It's Nancy Cartwright. 100% sure.\n";

cout << numberOfAnswer2;

cin >> yourAnswer;

if (yourAnswer == 2)

{

cout << goodAnswer04;

delay(3000);

correctAnswers++;

score = 500000;

}

if (yourAnswer != 4)

{

cout << badAnswer04;

delay(3000);

score = 32000;

walkAway();

return 0;

}

if (yourAnswer == 7)

{

score = 32000;

walkAway();

return 0;

}

if (yourAnswer < 1 || yourAnswer > 7)

{

cout << badAnswer04;

delay(3000);

score = 32000;

walkAway();

return 0;

}

}

}

if (yourAnswer == 7)

{

score = 32000;

walkAway();

return 0;

}

if (yourAnswer != 1 && yourAnswer != 2 && yourAnswer != 3 && yourAnswer != 4 && yourAnswer != 5 && yourAnswer != 6 && yourAnswer != 7)

{

score = 32000;

walkAway();

return 0;

}

clrscr();

counter++;

cout << "Try for the Grand Prize $1,000,000!!!\n\n";

delay(3000);

clrscr();

loop04:

cout << question05;

delay(3000);

cout << "1. Christopher Reeves 2. Don Adams\n";

delay(1000);

cout << "3. George Reeves 4. Dean Caan\n";

cout << numberOfAnswer;

cout << numberOfAnswer2;

cin >> yourAnswer;

if (yourAnswer == 3)

{

cout << goodAnswer05;

delay(3000);

correctAnswers++;

score = 1000000;

}

if (yourAnswer == 1 || yourAnswer == 2 || yourAnswer == 4)

{

cout << badAnswer05;

delay(3000);

score = 32000;

walkAway();

return 0;

}

if (yourAnswer == 5)

{

if (fiftyFiftyUsed == 1)

{

cout << "\nSorry 50/50 has already been used!\n";

delay(3000);

clrscr();

goto loop04;

}

if (fiftyFiftyUsed != 1)

{

fiftyFiftyUsed = 1;

answer05();

if (yourAnswer != 3)

{

score = 32000;

return 0;

}

}

}

if (yourAnswer == 6)

{

if (phoneFriendUsed == 1)

{

cout << "Sorry, phone a friend has been used!\n";

delay(3000);

clrscr();

goto loop04;

}

if (phoneFriendUsed != 1)

{

phoneFriendUsed = 1;

delay(2000);

cout << "Dialing...\n";

delay(2000);

cout << "You have been connected to " << people05 << ".\n";

cout << people05 << ",\n" << question05 << "\n";

cout << "1. Christopher Reeves 2. Don Adams\n";

cout << "3. George Reeves 4. Dean Caan\n";

delay(2000);

cout << "\n\n" << people05 << ": " << "I know Christopher Reeves is well known for that. I think it's him. 50% sure.\n";

cout << numberOfAnswer2;

cin >> yourAnswer;

if (yourAnswer == 3)

{

cout << goodAnswer05;

delay(3000);

correctAnswers++;

score = 1000000;

}

if (yourAnswer != 3)

{

cout << badAnswer05;

delay(3000);

score = 32000;

walkAway();

return 0;

return 0;

}

}

}

if (yourAnswer == 7)

{

score = 32000;

walkAway();

return 0;

}

if (yourAnswer != 1 && yourAnswer != 2 && yourAnswer != 3 && yourAnswer != 4 && yourAnswer != 5 && yourAnswer != 6 && yourAnswer != 7)

{

score = 32000;

walkAway();

return 0;

}

walkAway();

return 0;

}

int main()

{

clrscr();

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

cout << "\* Who Wants To Be \*\n";

cout << "\* A \*\n";

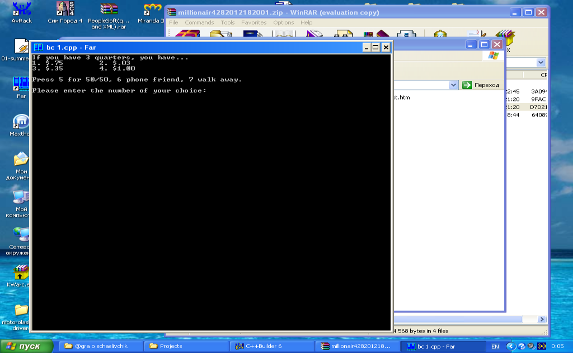
cout << "\* Millionaire?! \*\n";

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

continueProgram();

}

**Приклад скріншоту роботи програми**



Скрін 1. Простий текстовий інтерфейс роботи користувача з програмою

Програма має простий інтуїтивний інтерфейс, який схожий на той, що використовували старі ігрові програми – так звані текстові квести. Розібратися з ним не потребує багато зусиль. Ще раз підкреслимо що всі листинги та головний запускаємий модуль знаходяться на дискетці.

**Висновки**

Результатом нашої роботи є програма яка вчитель англійської мови. Дана програма має досить простий інтерфейс та алгоритмічну структуру. В якості інструменту розробки ми взяли мову програмування C++Builder.

Саме такий вибір дозволив в значній мірі спростити процес розробки та реалізації даного програмного продукту. Відмітимо лише, що дана програма має простий текстовий інтерфейс управління і не потребує попереднього навчання при роботі при роботі з собою.

**Література**

1. Касаткин А.И., Вальвачев А.Н. Профессиональное прогрпммирование на языке Си. Мн., 1992. 240 С.
2. Нейбауэр А. Моя первая программа на С/С++. П., 1995. 368 С.
3. Бруно Бабэ. Просто и ясно о Borland C++. М., 1996. 400 С.
4. Шамас Н.К. Основы С++ и обьектно-ориентированного программирования. К., 1996. 448 С.