Міністерство освіти та науки України

Закарпатський Державний Університет

Факультет інформатики

Кафедра загальної інформатики та математичного моделювання

## Реєстраційний №\_\_\_\_\_\_\_\_

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

# Курсова робота

### З основ програмування

#### Тема: Гра “Перший мільйон”

##### Рекомендовано до захисту

“\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_\_2005р.

Робота захищена

“\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_\_2005р.

з оцінкою

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

## Підписи членів комісії

Студента 2-го курсу

заочного відділення

Ісака І.І.

Науковий керівник:

викл. Дробнич О.В.

###### Ужгород

**2005**

**Зміст**

Вступ……………………………………………………….3

**1.Постановка задачі...……………………………………4**

**2.Вибір методу розв’язку та структура бази даних.....6**

**3.Опис алгоритму розв’язку……………………………7**

**4. Опис основних функцій програми………………….9**

**5.Лістинг програми…………………………………….11**

**Висновки………………………………………………...29**

**Список використаної літератури…………………….30**

**Додаток…………………………………………………..31**

**Вступ.**

Декілька років тому двоє британських журналістів вигадали нову телевізійну гру, відому тепер у всьому світі як "WHO WANTS TO BE A MILLIONAIRE?" - "ХТО ХОЧЕ СТАТИ МІЛЬЙОНЕРОМ?".

З'явившись у ефірі у вересні 1998 року на каналі ITV, ця гра одразу стала сенсацією британського телебачення, потім приголомшила Америку, а нині ліцензію на показ шоу продано вже понад 52 країнам світу.

Задум авторів програми полягав у тому, щоб учасник, подолавши всі інтелектуальні перешкоди, мав шанс виграти неймовірно привабливий приз. Якщо гравець правильно відповідає на 15 запитань, то в результаті отримує чек на 1.000.000. Сенс гри полягає у тому, що гравцеві доводиться задіяти весь свій розумовий потенціал.

На кожне запитання є чотири варіанти відповіді, але правильний - тільки один. Вибір правильного варіанта - і є суттю гри.

Запитання поділяються на кілька рівнів складності: перші п'ять, як правило, досить прості, п'ять наступних вимагають уже не тільки кмітливості, а й ґрунтовних знань, а останні п'ять - найскладніші. Утім, гравець має право скористатися трьома підказками - "50/50" (комп'ютер знімає дві неправильні відповіді), "Допомога залу" (гравець може звернутися по допомогу до глядачів у студії), "Допомога друга" (можна подзвонити другові й порадитися з ним). Проте в кожному разі без грошей ніхто не залишається, а розмір суми прямо залежить від інтелектуального рівня гравця. Правила гри дуже прості, і дозволяють гравцеві у будь-який момент часу забрати виграну суму грошей.

Тому написавши комп’ютерну гру “Перший мільйон”, ми зможемо надати кожному бажаючому можливість відчути себе гравцем популярної телевізійної гри.

**Постановка задачі.**

Мета даної курсової роботи – написати на одній з мов програмування гру “Перший мільйон”.

Задача полягає в тому, що алгоритм програми повинен відповідати правилам телевізійної гри “Перший мільйон”. Тобто вона повинна містити такі основні компоненти: питання та чотири варіанти відповіді на нього, три підсказки (50/50, допомога залу, дзвінок другу). За кожну правильну відповідь учасник отримує право на певний грошовий приз. Рівень складності запитань поступово наростає, зате кожне наступне "коштує" більше, ніж попереднє, і задається воно тільки після одержання правильної відповіді на попереднє. Таким чином, розмір призу наростає від ста до 1 мільйона гривень.

 У ході гри гравець має право на 3 підказки: "50/50" (комп'ютер знімає дві хибні відповіді), "Допомога залу" (гравець може звернутися по допомогу до глядачів у студії), "Допомога друга" (можна зателефонувати другові й порадитися з ним). Кожну з трьох підказок гравець має право використати лише один раз. Ці підказки він може використати в будь-якій послідовності і на будь-якому етапі гри, відповідаючи на різні запитання чи на одне й те ж будь-якого рівня складності. Використання підказок ніяк не впливає на розмір належного грошового призу.

Якщо гравець сумнівається, чи зможе правильно відповісти на запитання, програма повинна дозволити йому відмовитися від подальшої участі у грі, перед тим як він зробить свій вибір і одержати приз, який відповідає вартості останнього запитання, на яке гравець відповів.

У випадку, якщо гравець дасть неправильну відповідь на запитання в діапазоні 1-5, він вибуває з гри без призу. Якщо це станеться у проміжку 6-10, він одержить приз, який відповідає 5-му запитанню, якщо ж він дістався до діапазону 11-15, то в разі невдалої відповіді на котресь із запитань він "відкочується" лише до 10-го рівня, тобто він вибуває з гри, одержавши приз за 10-е запитання.

Зважаючи на те, що програма – лише аналог телевізійної гри, а гравець тільки один, у програмі буде відсутній відбірковий тур, буде тільки основний етап. Так як реального залу та друга, якому можна зателефонувати, у комп’ютерному варіанті гри не існує, роль залу та друга, якому телефонуємо, буде виконувати програма.

Вибір методу розв’язку.

Дану задачу зручно вирішити методами об’єктно–орієнтованого програмування на мові Borland C++ Builder 6. Мова С++ є універсальною, має широку вигоду в області використовування графіки, а концепція ООП, підтримувана нею дозволить максимально наблизити рішення задачі до образу людського рішення, спростивши тим самим задачу. З допомогою даної візуальної мови програмування, можна легко створити потрібний інтерфейс, звукове та візуальне оформлення гри “Перший мільйон”.

Структура бази даних.

 Для збереження питань у програмі використовуються 3 бази:

Level1.dbf, Level2.dbf, Level3.dbf. Три бази потрібні для того, щоб зберігати питання трьох рівней складності.

 Для зчитування питань програма використовує ще 3 бази: Temp1.dbf, Temp2.dbf, Temp3.dbf, які служать для тимчасового збереження в них питань та відповідей; тому, що після того, як програма зчитає питання, воно видаляється (це зроблено для того, щоб питання не повторювались)

 Кожна з баз складається з 6 полів: VOPROS, A, B, C, D, PR\_OTV. Поле VOPROS служить для збереження питання, у полях A, B, C, D містяться варіанти відповідей, у полі PR\_OTV зберігаються номери правильних варіантів відповідей.

Структура таблиці баз даних.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Назва поля | VOPROS | A | B | C | D | PR\_OTV |
| Тип і розмір  | C [128] | C [32] | C [32] | C [32] | C [32] | N [1] |

Опис аглоритму розв’язку.

Суть алгоритму, що описує зчитування питань з бази та їх перевірки можна описати наступним чином:

1.Відповідь на питання правильна? Якщо так, то п.2, інакше перехід на п.9.

2.Перевіряємо чи досягнуто 15 питання. Якщо так, то перехід на п.10, інакше п.3.

3.Якщо номер питання дорівнює 1, то зчитуємо питання з першої бази, якщо дорівнює 2 – зчитуємо з другої бази, якщо 3 – зчитуємо з третьої.

4.Збільшуємо позицію досягнутого питання.

5.Виводимо мітку досягнутої суми.

6.Обновляємо панелі з питанням та відповідями.

7.Перевіряємо чи досягнуто неспалимої суми. Якщо досягнуто, то п.11, інакше п.8.

8.Виводимо інформаційне вікно з інформацією про досягнуту суму виграшу.Перехід на п.1.

9.Позначаємо червоним кольором панель з варіантом, обраним користувачем, подаємо звукове повідомлення. Якщо досягнуто першої неспалимої суми, то виграш дорівнює 1 000 грн., якщо другої неспалимої суми, то виграш – 100 000 грн., інакше виграш дорівнює 0 грн. Виводимо вікно з інформацією про виграну суму та кінець гри. Перехід на п.12.

10.Виводимо вікно з привітанням про виграш 1 000 000 грн. Перехід на п.12.

11.Якщо досягнуто 5 питання, то досягнуто першої неспалимої суми. Якщо досягнуто 10 питання, то досягнуто другої неспалимої суми. Вивід вікна з інформацією про досягнення неспалимої суми. Перехід на п.8.

12. Кінець гри

Алгоритм підказки “допомога залу” побудований таким способом:

1.Дана підказка вже використовувалася? Якщо так, то п.5, інакше п.2.

2.Зчитуємо з бази номер правильного варіанту відповіді.

3.Для стовбчика з правильним варіантом відповіді генеруємо випадкове

 число з діапазону 50-100 та додаємо до нього 50. Таким чином стовпчик з правильним варіантом відповіді завжди буде більшим за 50%.

4.Числа для інших трьох стовпчиків генеруємо так, щоб відсотки всіх чотирьох стовпчиків в сумі дорівнювали 100.

5.Повернення в основну програму.

Алгоритм підказки “дзвінок другу”:

 1.Дана підказка вже використовувалася? Якщо так, то п.5, інакше п.2.

2.Довжина числа, введеного в якості номера телефону, дорівнює 6 символів? Якщо так, то перехід на п.4, інакше п.3.

3.Вводимо наступну цифру. Перехід на п.2.

4.Звіряємо введений номер з одним із запропонованих номерів абонентів, виводимо відповідь в залежності від обраного абонента.

5.Повернення в основну програму.

Алгоритм підказки “50 на 50”:

 1.Дана підказка вже використовувалася? Якщо так, то п.4, інакше п.2.

 2.Зчитуємо з бази номер правильного варіанту відповіді.

 3.Забираємо з панелей два неправильних варіанти відповідей.

 4. Повернення в основну програму.

Опис основних функцій програми.

void init\_form ()

– виконує ініціалізацію глобальних змінних, та присвоює їм початкове значення. Встановлює вікно програми в початковий стан.

void verify (int p)

– перевіряє відповідь гравця та виконує певні функції в залежності від істинності відповіді на питання.

void nepravilno (int panel)

– визивається при неправильній відповіді, виводить вікно з інформацією про кінець гри.

void score (int s)

* виводить вікно з інформацією про досягнуту суму.

void next (void)

* перміщає мітку досягнутого питання на одну позицію вверх та збільшує досягнуту суму.

void no\_fire ()

* виконує перевірку на досягнення неспалимої суми та виводить вікно з інформацією про досягнення неспалимої суми.

void random\_question (void)

* забезпечує хаотичний вибір питання з бази даних питань.

void out\_text (AnsiString x0,AnsiString x1,AnsiString x2,AnsiString x3,AnsiString x4)

 – виводить на панелі текст питання та варіанти відповідей.

void update (void)

* оновляє текст питання та варіантів відповідей на панелях.

void fifty (void)

* прибирає дві неправильні відповіді.

void play (AnsiString file\_name)

* відтворює звукові файли.

void save\_rating (void)

 - записує виграну суму в таблицю рейтингу гравців

void level (void)

* вибирає одну з трьох баз з питаннями для зчитування, в залежності від номера питання та не дозволяє повторюватись питанням.

void restore\_base1 (void) , void restore\_base2 (void),

 void restore\_base3 (void)

* заповнють тимчасові таблиці Temp1.dbf, Temp2.dbf,Temp3 даними з таблиць Level1.dbf , Level2.dbf, Level3.dbf.

Лістинг програми.

**Модуль Millionar.cpp**

**#include <vcl.h>**

**#include <dos.h>**

**#pragma hdrstop**

**#include "Millionar.h"**

**#include "Millionar2.h"**

**#include "Telefon.h"**

**#include "Auditorium.h"**

**#include "Welcome.h"**

**#include "Stop.h"**

**#include "EditQuest.h"**

**#include "Password.h"**

**#include "Rating.h"**

**#include "Score.h"**

**#include "Game\_end.h"**

**#include "No\_Fire.h"**

**#include "Final.h"**

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

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

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

**TForm1 \*Form1;**

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

**\_\_fastcall TForm1::TForm1(TComponent\* Owner)**

 **: TForm(Owner)**

**{**

**}**

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

**int position=1,telefon=1,fift=1,auditoria=1,pr\_otv,**

 **count\_fields=0,next\_player=0,auditorium;// Global variables**

**long summa;**

**AnsiString v0="",v1="",v2="",v3="",v4="";**

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

**void out\_text (AnsiString x0,AnsiString x1,AnsiString x2,**

 **AnsiString x3,AnsiString x4);**

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

**void restore\_base1 (void);**

**void restore\_base2 (void);**

**void restore\_base3 (void);**

**void random\_question (void);**

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

**void level (void)**

**{**

 **if (position<6)**

 **{**

 **if (Form1->Table4->RecordCount==0) restore\_base1();**

 **random\_question ();**

 **v0=Form1->Table4->FieldValues[("VOPROS")];**

 **v1=Form1->Table4->FieldValues[("A")];**

 **v2=Form1->Table4->FieldValues[("B")];**

 **v3=Form1->Table4->FieldValues[("C")];**

 **v4=Form1->Table4->FieldValues[("D")];**

 **pr\_otv=Form1->Table4->FieldValues[("PR\_OTV")];**

 **Form1->Table4->Delete();**

 **}**

 **if (position>=6 && position<11)**

 **{**

 **if (Form1->Table5->RecordCount==0) restore\_base2();**

 **random\_question ();**

 **v0=Form1->Table5->FieldValues[("VOPROS")];**

 **v1=Form1->Table5->FieldValues[("A")];**

 **v2=Form1->Table5->FieldValues[("B")];**

 **v3=Form1->Table5->FieldValues[("C")];**

 **v4=Form1->Table5->FieldValues[("D")];**

 **pr\_otv=Form1->Table5->FieldValues[("PR\_OTV")];**

 **Form1->Table5->Delete();**

 **}**

 **if (position>=11)**

 **{**

 **if (Form1->Table6->RecordCount==0) restore\_base3();**

 **random\_question ();**

 **v0=Form1->Table6->FieldValues[("VOPROS")];**

 **v1=Form1->Table6->FieldValues[("A")];**

 **v2=Form1->Table6->FieldValues[("B")];**

 **v3=Form1->Table6->FieldValues[("C")];**

 **v4=Form1->Table6->FieldValues[("D")];**

 **pr\_otv=Form1->Table6->FieldValues[("PR\_OTV")];**

 **Form1->Table6->Delete();**

 **}**

**}**

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

**void play (AnsiString file\_name)**

**{**

 **Form1->Player1->FileName=file\_name;**

 **Form1->Player1->Open();**

 **Form1->Player1->Play();**

**}**

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

**void flash (int panel,TColor color)**

**{**

 **switch (panel)**

 **{**

 **case 1:Form1->Panel1->Color=color;break;**

 **case 2:Form1->Panel2->Color=color;break;**

 **case 3:Form1->Panel3->Color=color;break;**

 **case 4:Form1->Panel4->Color=color;break;**

 **}**

**}**

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

**void fifty (void)**

**{**

 **switch (auditorium)**

 **{**

 **case 1:**

 **{**

 **v1=v1;**

 **v2=v4="";**

 **}break;**

 **case 2:**

 **{**

 **v2=v2;**

 **v3=v1="";**

 **}break;**

 **case 3:**

 **{**

 **v3=v3;**

 **v4=v1="";**

 **}break;**

 **case 4:**

 **{**

 **v4=v4;**

 **v1=v2="";**

 **}**

 **}**

 **out\_text (v0,v1,v2,v3,v4);**

 **fift=0;**

**}**

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

**void clear\_level (void)**

**{**

 **TColor form\_color=Form1->Color;**

 **Form1->Label1->Color=form\_color;**

 **Form1->Label2->Color=form\_color;**

 **Form1->Label3->Color=form\_color;**

 **Form1->Label4->Color=form\_color;**

 **Form1->Label5->Color=form\_color;**

 **Form1->Label6->Color=form\_color;**

 **Form1->Label7->Color=form\_color;**

 **Form1->Label8->Color=form\_color;**

 **Form1->Label9->Color=form\_color;**

 **Form1->Label10->Color=form\_color;**

 **Form1->Label11->Color=form\_color;**

 **Form1->Label12->Color=form\_color;**

 **Form1->Label13->Color=form\_color;**

 **Form1->Label14->Color=form\_color;**

 **Form1->Label15->Color=form\_color;**

**}**

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

**void update (void)**

**{**

 **Form1->Panel1->Color=clBtnFace;**

 **Form1->Panel2->Color=clBtnFace;**

 **Form1->Panel3->Color=clBtnFace;**

 **Form1->Panel4->Color=clBtnFace;**

 **level();**

 **out\_text (v0,v1,v2,v3,v4);**

**}**

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

**void out\_text (AnsiString x0,AnsiString x1,AnsiString x2,**

 **AnsiString x3,AnsiString x4)**

**{**

 **Form1->Panel1->Caption=x1;**

 **Form1->Panel2->Caption=x2;**

 **Form1->Panel3->Caption=x3;**

 **Form1->Panel4->Caption=x4;**

 **Form1->Panel5->Caption=x0;**

**}**

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

**void random\_question (void)**

**{**

 **int base;**

 **if (position<=6) base=1;**

 **if (position>=6 && position<=11) base=2;**

 **if (position>11) base=3;**

 **randomize();**

 **switch (base)**

 **{**

 **case 1:**

 **{**

 **randomize();**

 **int rnd=random (Form1->Table4->RecordCount);**

 **Form1->Table4->First();**

 **Form1->Table4->MoveBy(rnd);**

 **}break;**

 **case 2:**

 **{**

 **randomize();**

 **int rnd=random (Form1->Table5->RecordCount);**

 **Form1->Table5->First();**

 **Form1->Table5->MoveBy(rnd);**

 **}break;**

 **case 3:**

 **{**

 **randomize();**

 **int rnd=random (Form1->Table6->RecordCount);**

 **Form1->Table6->First();**

 **Form1->Table6->MoveBy(rnd);**

 **}break;**

 **}**

**}**

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

**void no\_fire ()**

**{**

 **if (position==6)**

 **{**

 **NoFire->Summa->Left=160;**

 **NoFire->Summa->Caption=("1 000 грн.");**

 **NoFire->Level->Caption="першої";**

 **NoFire->ShowModal();**

 **}**

 **if (position==11)**

 **{**

 **NoFire->Summa->Left=144;**

 **NoFire->Summa->Caption=("32 000 грн.");**

 **NoFire->Level->Caption="другої";**

 **NoFire->ShowModal();**

 **}**

**}**

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

**void next (void)**

**{**

 **clear\_level();**

 **random\_question ();**

 **switch (position)**

 **{**

 **case 1:{Form1->Label1->Color=clYellow;summa=100;break;}**

 **case 2:{Form1->Label2->Color=clYellow;summa=200;break;}**

 **case 3:{Form1->Label3->Color=clYellow;summa=300;break;}**

 **case 4:Form1->Label4->Color=clYellow;summa=500;break;**

 **case 5:Form1->Label5->Color=clYellow;summa=1000;break;**

 **case 6:Form1->Label6->Color=clYellow;summa=2000;break;**

 **case 7:Form1->Label7->Color=clYellow;summa=4000;break;**

 **case 8:Form1->Label8->Color=clYellow;summa=8000;break;**

 **case 9:Form1->Label9->Color=clYellow;summa=16000;break;**

 **case 10:Form1->Label10->Color=clYellow;summa=32000;break;**

 **case 11:Form1->Label11->Color=clYellow;summa=64000;break;**

 **case 12:Form1->Label12->Color=clYellow;summa=100000;break;**

 **case 13:Form1->Label13->Color=clYellow;summa=200000;break;**

 **case 14:Form1->Label14->Color=clYellow;summa=500000;break;**

 **case 15:Form1->Label15->Color=clYellow;summa=1000000;break;**

 **}**

 **position++;**

**}**

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

**void score (int s)**

**{**

 **if (s==1)fmScore->ShowModal ();**

 **if (s==0) return;**

**}**

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

**void save\_rating (void)**

**{**

 **AnsiString name\_player=Form4->Edit1->Text;**

 **if (fmRating->Table1->RecordCount>10)**

 **{**

 **for (int i=0; i<fmRating->Table1->RecordCount; i++)**

 **{**

 **fmRating->Table1->Delete();**

 **fmRating->Table1->Next();**

 **}**

 **fmRating->Table1->First();**

 **}**

 **if (position==6) summa=1000;**

 **if (position==11) summa=100000;**

 **fmRating->Table1->Edit();**

 **fmRating->Table1->FieldValues[("NAME")]=name\_player;**

 **fmRating->Table1->FieldValues[("BODY")]=summa;**

 **fmRating->Table1->Next();**

**}**

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

**void nepravilno (int panel)**

**{**

 **AnsiString sound="Gong.wav";**

 **TColor color=clRed;**

 **switch (panel)**

 **{**

 **case 1:Form1->Panel1->Color=color;break;**

 **case 2:Form1->Panel2->Color=color;break;**

 **case 3:Form1->Panel3->Color=color;break;**

 **case 4:Form1->Panel4->Color=color;break;**

 **}**

 **play(sound);**

 **EndGame->ShowModal();**

 **save\_rating ();**

**}**

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

**void verify (int p)**

**{**

 **if (p==pr\_otv)**

 **{**

 **int s=1;**

 **TColor color=clGreen;**

 **if (position==15)**

 **{**

 **summa=1000000;**

 **save\_rating();**

 **score (s);**

 **fmFinal->Show();**

 **s=0;**

 **}**

 **next ();**

 **score (s);**

 **flash (p,color);**

 **update();**

 **no\_fire();**

 **}**

 **else nepravilno (p);**

**}**

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

**void restore\_base1 (void)**

**{**

 **Form1->Table1->First();**

 **Form1->Table4->First();**

 **for (int i=0; i<Form1->Table1->RecordCount; i++)**

 **{**

 **Form1->Table4->Insert();**

 **Form1->Table4->FieldValues[("VOPROS")]=Form1->Table1->FieldValues[("VOPROS")];**

 **Form1->Table4->FieldValues[("A")]=Form1->Table1->FieldValues[("A")];**

 **Form1->Table4->FieldValues[("B")]=Form1->Table1->FieldValues[("B")];**

 **Form1->Table4->FieldValues[("C")]=Form1->Table1->FieldValues[("C")];**

 **Form1->Table4->FieldValues[("D")]=Form1->Table1->FieldValues[("D")];**

 **Form1->Table4->FieldValues[("PR\_OTV")]=Form1->Table1->FieldValues[("PR\_OTV")];**

 **Form1->Table1->Next();**

 **Form1->Table4->Next();**

 **}**

**}**

**void restore\_base2 (void)**

**{**

 **Form1->Table2->First();**

 **Form1->Table5->First();**

 **for (int i=0; i<Form1->Table2->RecordCount; i++)**

 **{**

 **Form1->Table5->Insert();**

 **Form1->Table5->FieldValues[("VOPROS")]=Form1->Table2->FieldValues[("VOPROS")];**

 **Form1->Table5->FieldValues[("A")]=Form1->Table2->FieldValues[("A")];**

 **Form1->Table5->FieldValues[("B")]=Form1->Table2->FieldValues[("B")];**

 **Form1->Table5->FieldValues[("C")]=Form1->Table2->FieldValues[("C")];**

 **Form1->Table5->FieldValues[("D")]=Form1->Table2->FieldValues[("D")];**

 **Form1->Table5->FieldValues[("PR\_OTV")]=Form1->Table2->FieldValues[("PR\_OTV")];**

 **Form1->Table2->Next();**

 **Form1->Table5->Next();**

 **}**

**}**

**void restore\_base3 (void)**

**{**

 **Form1->Table3->First();**

 **Form1->Table6->First();**

 **for (int i=0; i<Form1->Table3->RecordCount; i++)**

 **{**

 **Form1->Table6->Insert();**

 **Form1->Table6->FieldValues[("VOPROS")]=Form1->Table3->FieldValues[("VOPROS")];**

 **Form1->Table6->FieldValues[("A")]=Form1->Table3->FieldValues[("A")];**

 **Form1->Table6->FieldValues[("B")]=Form1->Table3->FieldValues[("B")];**

 **Form1->Table6->FieldValues[("C")]=Form1->Table3->FieldValues[("C")];**

 **Form1->Table6->FieldValues[("D")]=Form1->Table3->FieldValues[("D")];**

 **Form1->Table6->FieldValues[("PR\_OTV")]=Form1->Table3->FieldValues[("PR\_OTV")];**

 **Form1->Table3->Next();**

 **Form1->Table6->Next();**

 **}**

**}**

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

**void init\_form ()**

**{**

 **telefon=0;auditoria=0;fift=0;summa=0;// Global variables**

 **count\_fields=0;**

 **clear\_level ();**

 **Form1->Image8->Show();**

 **Form1->StaticText3->Hide();**

 **Form1->StaticText4->Hide();**

 **Form1->Panel1->Hide();**

 **Form1->Panel2->Hide();**

 **Form1->Panel3->Hide();**

 **Form1->Panel4->Hide();**

 **Form1->Panel5->Hide();**

 **if (Form1->Table4->RecordCount==0) restore\_base1 ();**

 **if (Form1->Table5->RecordCount==0) restore\_base2 ();**

 **if (Form1->Table6->RecordCount==0)restore\_base3 ();**

 **out\_text ("","","","","");**

**}**

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

**void \_\_fastcall TForm1::Image3Click(TObject \*Sender)**

**{**

 **if (auditoria==1)**

 **{**

 **auditorium=pr\_otv;**

 **Form5->ShowModal();**

 **}**

**}**

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

**void \_\_fastcall TForm1::N9Click(TObject \*Sender)**

**{**

 **if (auditoria==1)**

 **{**

 **auditorium=pr\_otv;**

 **Form5->ShowModal();**

 **}**

**}**

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

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

**{**

 **init\_form ();**

**}**

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

**void \_\_fastcall TForm1::Button8Click(TObject \*Sender)**

**{**

 **Label1->Font->Color=clGreen;**

**}**

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

**void \_\_fastcall TForm1::N3Click(TObject \*Sender)**

**{**

 **Form2->Show ();**

**}**

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

**void \_\_fastcall TForm1::N8Click(TObject \*Sender)**

**{**

 **Form1->Close();**

**}**

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

**void \_\_fastcall TForm1::N10Click(TObject \*Sender)**

**{**

 **if (telefon==0) Form3->Close();**

 **else Form3->ShowModal();**

**}**

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

**void \_\_fastcall TForm1::Image1Click(TObject \*Sender)**

**{**

 **if (telefon==0) Form3->Close();**

 **else Form3->ShowModal();**

**}**

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

**void \_\_fastcall TForm1::Image1DragOver(TObject \*Sender, TObject \*Source,**

 **int X, int Y, TDragState State, bool &Accept)**

**{**

 **Image4->Visible=false;**

**}**

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

**void \_\_fastcall TForm1::Image1MouseDown(TObject \*Sender,**

 **TMouseButton Button, TShiftState Shift, int X, int Y)**

**{**

 **Image4->Show();**

**}**

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

**void \_\_fastcall TForm1::Image1MouseUp(TObject \*Sender, TMouseButton Button,**

 **TShiftState Shift, int X, int Y)**

**{**

 **Image4->Hide();**

**}**

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

**void \_\_fastcall TForm1::Image3MouseDown(TObject \*Sender,**

 **TMouseButton Button, TShiftState Shift, int X, int Y)**

**{**

 **Image5->Show();**

**}**

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

**void \_\_fastcall TForm1::Image3MouseUp(TObject \*Sender, TMouseButton Button,**

 **TShiftState Shift, int X, int Y)**

**{**

 **Image5->Hide();**

**}**

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

**void \_\_fastcall TForm1::Image5MouseUp(TObject \*Sender, TMouseButton Button,**

 **TShiftState Shift, int X, int Y)**

**{**

 **Image5->Hide();**

**}**

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

**void \_\_fastcall TForm1::Image2MouseDown(TObject \*Sender,**

 **TMouseButton Button, TShiftState Shift, int X, int Y)**

**{**

 **Image6->Show();**

**}**

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

**void \_\_fastcall TForm1::Image2MouseUp(TObject \*Sender, TMouseButton Button,**

 **TShiftState Shift, int X, int Y)**

**{**

 **Image6->Hide();**

**}**

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

**void \_\_fastcall TForm1::Panel1MouseUp(TObject \*Sender, TMouseButton Button,**

 **TShiftState Shift, int X, int Y)**

**{**

 **Panel1->Color=clBtnFace;**

**}**

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

**void \_\_fastcall TForm1::N4Click(TObject \*Sender)**

**{**

 **telefon=1;auditoria=1;fift=1;summa=0;// Global variables**

 **position=1;**

 **clear\_level ();**

 **out\_text ("","","","","");**

 **next\_player++;**

 **Form4->Show();**

**}**

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

**void reset\_all ()**

**{**

 **Form1->Close();**

 **Form1->Show();**

**}**

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

**void \_\_fastcall TForm1::Panel1Click(TObject \*Sender)**

**{**

 **int panel=1;**

 **verify (panel);**

**}**

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

**void \_\_fastcall TForm1::Panel2Click(TObject \*Sender)**

**{**

 **int panel=2;**

 **verify (panel);**

**}**

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

**void \_\_fastcall TForm1::Panel3Click(TObject \*Sender)**

**{**

 **int panel=3;**

 **verify (panel);**

**}**

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

**void \_\_fastcall TForm1::Panel4Click(TObject \*Sender)**

**{**

 **int panel=4;**

 **verify (panel);**

**}**

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

**void \_\_fastcall TForm1::Image2Click(TObject \*Sender)**

**{**

 **if (fift==1)**

 **{**

 **AnsiString sound="Fifty.wav";**

 **play (sound);**

 **auditorium=pr\_otv;**

 **fifty ();**

 **}**

**}**

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

**void \_\_fastcall TForm1::N7Click(TObject \*Sender)**

**{**

 **PasswDlg->StaticText1->Caption="Введіть пароль:";**

 **PasswDlg->Edit1->Clear();**

 **PasswDlg->ShowModal();**

**}**

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

**void \_\_fastcall TForm1::N6Click(TObject \*Sender)**

**{**

 **fmRating->ShowModal();**

**}**

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

**void \_\_fastcall TForm1::Timer1Timer(TObject \*Sender)**

**{**

 **Timer1->Tag++;**

**}**

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

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

**{**

 **random\_question ();**

 **update ();**

**}**

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

**void \_\_fastcall TForm1::N11Click(TObject \*Sender)**

**{**

 **AnsiString sound="Stop.wav";**

 **play (sound);**

 **save\_rating ();**

 **fmStop->ShowModal();**

**}**

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

**void \_\_fastcall TForm1::N50x501Click(TObject \*Sender)**

**{**

 **if (fift==1)**

 **{**

 **AnsiString sound="Fifty.wav";**

 **play (sound);**

 **auditorium=pr\_otv;**

 **fifty ();**

 **}**

**}**

**Модуль Auditorium.cpp**

**#include <vcl.h>**

**#pragma hdrstop**

**#include "Auditorium.h"**

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

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

**#pragma link "CGAUGES"**

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

**TForm5 \*Form5;**

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

**\_\_fastcall TForm5::TForm5(TComponent\* Owner)**

 **: TForm(Owner)**

**{**

**}**

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

**void \_\_fastcall TForm5::FormActivate(TObject \*Sender)**

**{**

 **extern auditorium;**

 **extern auditoria;**

 **extern play(AnsiString);**

 **AnsiString sound="Zal.wav";**

 **play (sound);**

 **int x1,x2,x3,x4;**

 **randomize ();**

 **int x=50+random (50);**

 **int y=random (100-x);**

 **int z=random (100-(x+y));**

 **int j=random (100-(x+y+z));**

 **int n=100-(x+y+j+z);j=j+n;**

 **if (auditorium==1){x1=x;x2=y;x3=z;x4=j;}**

 **if (auditorium==2){x1=y;x2=x;x3=z;x4=j;}**

 **if (auditorium==3){x1=z;x2=y;x3=x;x4=j;}**

 **if (auditorium==4){x1=j;x2=y;x3=z;x4=x;}**

 **CGauge1->Progress=x1;**

 **CGauge2->Progress=x2;**

 **CGauge3->Progress=x3;**

 **CGauge4->Progress=x4;**

 **auditoria=0;**

**}**

**Модуль Telefon.cpp**

**#include <vcl.h>**

**#include <string.h>**

**#pragma hdrstop**

**#include "Telefon.h"**

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

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

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

**TForm3 \*Form3;**

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

**\_\_fastcall TForm3::TForm3(TComponent\* Owner)**

 **: TForm(Owner)**

**{**

**}**

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

**extern int telefon; AnsiString c;**

**extern play(AnsiString);**

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

**void out (AnsiString z)**

 **{**

 **if (telefon==0) Form3->Close();**

 **if (z=="Clear")**

 **{**

 **Form3->Edit1->Clear();**

 **z="";**

 **}**

 **Form3->Edit1->Text=Form3->Edit1->Text+z;**

 **}**

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

**void dial (void)**

 **{**

 **if ((Form3->Edit1->Text.Length())==6)**

 **{**

 **extern auditorium;**

 **AnsiString telc=Form3->Edit1->Text;**

 **double tel=StrToInt(Form3->Edit1->Text);**

 **if (tel==634512)**

 **{**

 **Form3->Label1->Caption=" Б.Єльцін не знає правильної відповіді.";**

 **telefon=0;**

 **}**

 **if (tel==398276)**

 **{**

 **Form3->Label1->Caption="В.Ющенко точно знає,що провильний варіант A.";**

 **telefon=0;**

 **}**

 **if (tel==158204)**

 **{**

 **Form3->Label1->Caption=" Б.Гейтс не сумнівається у варіанті С.";**

 **telefon=0;**

 **}**

 **if (tel==834577)**

 **{**

 **Form3->Label1->Caption=" Дід Мороз гадає, що правильна відповідь B.";**

 **telefon=0;**

 **}**

 **if (tel==597420)**

 **{**

 **Form3->Label1->Caption=" Б.Ступка думає, що правильний варіант D.";**

 **telefon=0;**

 **}**

 **if (tel==964816)**

 **{**

 **Form3->Label1->Caption=" Буратіно впевнений у варіанті В.";**

 **telefon=0;**

 **}**

 **}**

 **}**

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

**void \_\_fastcall TForm3::Button10Click(TObject \*Sender)**

**{**

 **c="0";**

 **out (c);**

 **dial ();**

**}**

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

**void \_\_fastcall TForm3::Button1Click(TObject \*Sender)**

**{**

 **c="1";**

 **out (c);**

 **dial ();**

**}**

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

**void \_\_fastcall TForm3::Button2Click(TObject \*Sender)**

**{**

 **c="2";**

 **out (c);**

 **dial ();**

**}**

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

**void \_\_fastcall TForm3::Button3Click(TObject \*Sender)**

**{**

 **c="3";**

 **out (c);**

 **dial ();**

**}**

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

**void \_\_fastcall TForm3::Button4Click(TObject \*Sender)**

**{**

 **c="4";**

 **out (c);**

 **dial ();**

**}**

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

**void \_\_fastcall TForm3::Button5Click(TObject \*Sender)**

**{**

 **c="5";**

 **out (c);**

 **dial ();**

**}**

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

**void \_\_fastcall TForm3::Button6Click(TObject \*Sender)**

**{**

 **c="6";**

 **out (c);**

 **dial ();**

**}**

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

**void \_\_fastcall TForm3::Button7Click(TObject \*Sender)**

**{**

 **c="7";**

 **out (c);**

 **dial ();**

**}**

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

**void \_\_fastcall TForm3::Button8Click(TObject \*Sender)**

**{**

 **c="8";**

 **out (c);**

 **dial ();**

**}**

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

**void \_\_fastcall TForm3::Button9Click(TObject \*Sender)**

**{**

 **c="9";**

 **out (c);**

 **dial ();**

**}**

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

**void \_\_fastcall TForm3::Button11Click(TObject \*Sender)**

**{**

 **Label1->Caption="";**

 **c="Clear";**

 **out (c);**

 **dial ();**

**}**

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

**void \_\_fastcall TForm3::FormActivate(TObject \*Sender)**

**{**

 **Label1->Caption="";**

 **Edit1->Clear();**

 **AnsiString sound="Zal.wav";**

 **play (sound);**

**}**

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

**void \_\_fastcall TForm3::FormShow(TObject \*Sender)**

**{**

 **c=" ";**

 **out (c);**

**}**

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

**void \_\_fastcall TForm3::Edit1Enter(TObject \*Sender)**

**{**

 **dial ();**

**}**

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

Висновки

У даній курсовій роботі детально описано алгоритм функціонування гри “Перший мільйон”.

Програма написана методами об’єктно-орієнтованого програмування на мові програмування високого рівня – Borland C++ Builder 6. У програмі був використаний компонент TТable для збереження і редагування таблиць з питаннями і відповідями – як один з найпростіших способів звернення до таблиць баз даних.

Питання що задаються гравцеві, методом випадкового відбору вибираються з трьох таблиць. Такий спосіб дозволяє застосовувати у грі питання трьох рівнів складності (в залежності від номера питання). Алгоритм випадкового вибору питань, складений так, що не дозволяє повторюватися питанням до тих пір, поки всі питання не будуть зчитані з бази. Окільки кількість питань у базах омежена, при багаторазовому запуску програми вони можуть повторюватися. У програмі передбачено редагування та поповнення баз питань користувачем.

 Результатом даної роботи є програма, що є аналогом телевізійної гри “Перший мільйон”, яка дозволяє користувачеві редагувати бази з питаннями,

забезпечує неповторюваність питань та має звичний користувачу інтерфейс, візуальне та звукове оформлення подібне до телевізійної гри.

**Список використаної літератури.**

1.С.Пратт “Язык программирования С++.Лекции и упражнения” -Изд. Диа- Софт 2003.

2.Глушаков С.В., Коваль А.В., Смирнов С.В. “Язык программирования С++.Учебный курс” -Изд. АСТ.

3.У.Савич “Программирование на С++” -Изд. BHV. 2004.

4.Карпова Т. С. Бази данных: модели, разработка, реализация. – СПб. : Питер, 2001г.

5.http://www.bcb.net.ru

**Додаток**

 **Мал.1. Головне вікно програми.**

Мал.2 Вікно “Допомога залу”. Мал.3. Вікно “Дзвінок другу”



 Мал.4. Вікно “Нова гра” Мал.5. Вікно “Вітання”

Мал 6. Вікно редагування питань.



 Мал 7. Меню “Гра” Мал 8. Меню “Підказки”

Мал 9. Вікно “Про програму”