



Національний університет
водного господарства та природокористування

Міністерство освіти і науки України

Національний університет водного господарства та
природокористування

Кафедра автоматизації, електротехнічних та комп'ютерно-
інтегрованих технологій

04-03-224

МЕТОДИЧНІ ВКАЗІВКИ

до лабораторної роботи № 4

із навчальної дисципліни

“Інформатика та комп'ютерна техніка”

для здобувачів вищої освіти першого (бакалаврського)

рівня за спеціальністю

151 “Автоматизація та комп'ютерно-інтегровані технології”

денної та заочної форм за скороченим терміном навчання

Рекомендовано науково-
методичною комісією
зі спеціальності 151
“Автоматизація та
комп'ютерно-інтегровані
технології”
Протокол № 1
від 4 вересня 2018 р.

Рівне – 2018



Методичні вказівки до лабораторної роботи № 4 із навчальної дисципліни “Інформатика та комп’ютерна техніка” для здобувачів вищої освіти першого (бакалаврського) рівня за спеціальністю: 151 “Автоматизація та комп’ютерно-інтегровані технології” денної та заочної форм за скороченим терміном навчання / А. П. Сафоник, І. С. Аврука. – Рівне : НУВГП, 2018. – 13 с.

Укладачі: А. П. Сафоник - доктор технічних наук, професор кафедри автоматизації, електротехнічних та комп’ютерно-інтегрованих технологій НУВГП, І. С. Аврука - старший викладач кафедри автоматизації, електротехнічних та комп’ютерно-інтегрованих технологій НУВГП.

Відповідальний за випуск - В. В. Древецький, д.т.н., професор, завідувач кафедри автоматизації, електротехнічних та комп’ютерно-інтегрованих технологій.



4.1. Мета роботи

Розробити програму для перевірки знань англійських слів шляхом тестування. Набути навиків роботи з такими об'єктами: геометрична фігура (Shape), таймер (Timer), мультимедійний програвач (MediaPlayer), спарена кнопка з полем редагування (CSpinEdit), індикатори стану (Gauge та ProgressBar).

У результаті виконання навчальної дисципліни студент вищої освіти повинен

- **знати:** типи програм в середовищі C++ Builder, види об'єктів та їх основні властивості.
- **вміти:** працювати в середовищі C++ Builder, програмувати кнопки в даному середовищі, працювати з об'єктами (геометрична фігура, таймер, мультимедійний програвач, спарена кнопка з полем редагування, індикатор стану), створювати програму для перевірки знань англійських слів шляхом тестування.

4.2. Теоретичні відомості

Розглянемо деякі нові об'єкти.

Таймер (Timer, піктограма на закладці System) використовують для повторення фрагмента коду програми з певною періодичністю. Відповідний фрагмент розташовують у тілі функції опрацювання події OnTimer таймера. Періодичність вмикання таймера у мілісекундах задають властивістю Interval. Геометрична фігура (Shape, піктограма на закладці Additional) призначена для зображення елементарних геометричних фігур і має, зокрема, такі властивості:

| Властивість | Опис властивості | Приклади значень |
|-------------|------------------|------------------|
|-------------|------------------|------------------|

| | | |
|--------------------|--|--|
| Brush | Характеристики кольору (Color) і стилю (Style) заливки | Brush-Color: clMaroon Brush-Style: bsSolid, bsVertical |
| Властивість | Опис властивості | Приклади значень |
| Shape | Форма фігури | stRoundRect – прямокутник зі округленими краями stEllipse, stSquare |
| Pen | Характеристики границі фігури | Комплексна властивість |

Мультимедійний програвач (MediaPlayer, піктограма на закладці System) призначений для програвання відео- та аудіофайлів. Керування програвачем може здійснюватися як за допомогою традиційних кнопок Play, Pause, Stop, Next тощо на етапі виконання програми, так і з програмного коду шляхом виконання методів цього об'єкта, наприклад:

```
MediaPlayer1->FileName="повне ім'я відео- чи аудіофайлу";
MediaPlayer1 ->Open();
MediaPlayer1 ->Play();
```

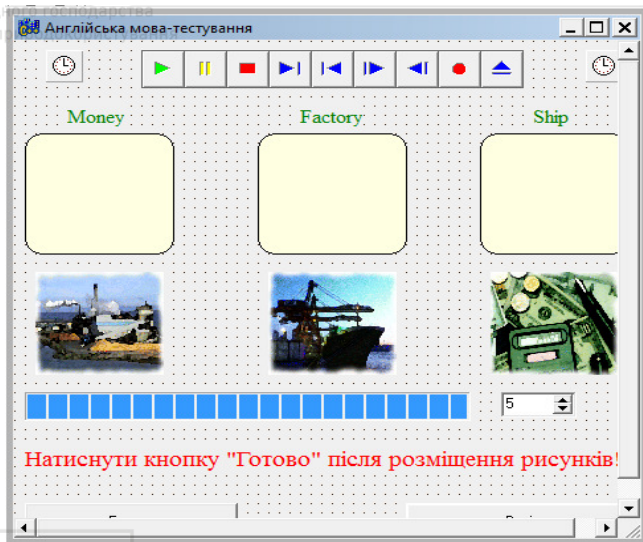


Рис. 4.1. Створення форми для тестування знань

Спарена кнопка з полем редагування (CSpinEdit, піктограма на закладці Samples) призначена для введення та корекції цілочислового значення деякої величини під час роботи програми. Властивості кнопки:

| Властивість | Опис властивості | Приклади значень |
|---------------|-----------------------------------|-------------------|
| EditirEnabled | Можливість прямого редагування | True, False |
| Enabled | Можливість будь-якого редагування | True, False |
| Increment | Крок зміни | 1; 4 (ціле число) |
| MaxValue | Максимальне значення | Ціле число |
| MinValue | Мінімальне значення | Ціле число |

Індикатор стану (CGauge, піктограма W на закладці Samples, ProgressBar, піктограма "" на закладці Win32) використовують для наочної демонстрації стану виконання деякого процесу. Розглянемо три властивості індикатора CGauge:

| Властивість | Опис властивості | Приклади значень |
|--------------------|---------------------------------------|--|
| Kind | Тип індикатора | gkHorizontalBar – горизонтальний рядок, gkPie – кругова діаграма gkText – процентне відображення |
| Progress | Відображає стан індикатора | Ціле число між MaxValue і MinValue |
| ShoeText | Додатково відображає стан у процентах | True, False |

4.3. Програма роботи

4.3.1. Розробити програму для перевірки знань англійських слів шляхом тестування.



Національний університет
водного господарства
та природокористування

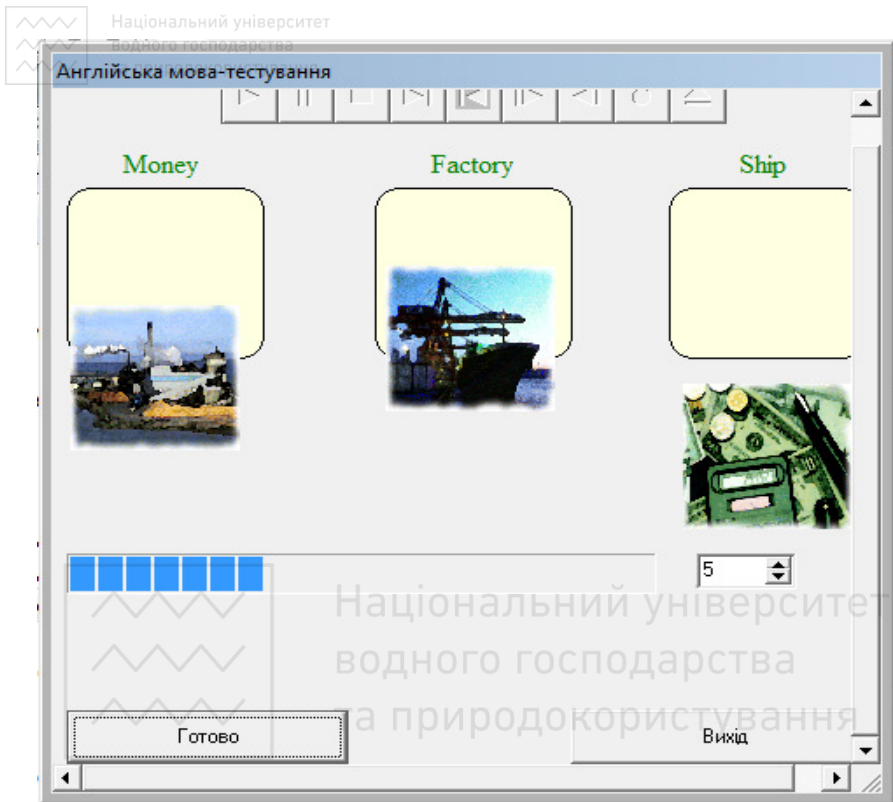


Рис.4. 2. Тестування з англійської мови

Користувач має за обмежений час методом перетягування розташувати три малюнки під відповідними англійськими словами (див. рис. 4.2). Застосувати індикатор часу виконання завдання, звукові ефекти та навести аналіз - підсумок тестування.

4.4. Обладнання та програмне забезпечення

4.4.1. Персональний комп'ютер.

4.4.2. Програмне забезпечення: C++ Builder 6.0.

4.5. Порядок виконання роботи і опрацювання результатів

4.5.1. Завантажте середовище візуального програмування C++Builder.

4.5.2. Замініть заголовок (Caption) форми з "Form1" на "Англійська мова - тестування".



4.5.3. Відмовтесь від усіх системних кнопок форми, задавши значення False в усіх позиціях властивості BorderIcons: biSystemMenu, biMinimize, biMaximize та biHelp. Роботу з програмою завершуватимемо натисканням на кнопку "Вихід".

4.5.4. Розташуйте на формі об'єкти так, як показано на рис. 4.1. Залайте такі властивості об'єктів:

| Властивість | Опис властивості | Приклади значень |
|---------------------------|------------------|---|
| Timer1 | Interval | 500 |
| Timer2 | Interval | 100 |
| MediaPlayer1 | Visible | False |
| | Orientation | pbHorizontal |
| ProgressBar1 | Max | 100 |
| | Min | 0 |
| | Position | 100 |
| SpinEdit | EditorEnabled | False |
| | MaxValue | 10 |
| | MinValue | 1 |
| | Value | 5 |
| BitBtn1 | Kind | bkOk |
| | Caption | Готово |
| BitBtn2 | Kind | bkClose |
| | Caption | Вихід |
| Shape1, Shape2, Shape3 | Brush-Color | clInfoBk |
| | Shape | stRoundRect |
| Image1, Image2, Image3 | Stretch | True |
| Label1 | Caption | Обов'язково натисніть на кнопку «Готово» після розміщення рисунків!!! |
| | WordWrap | True |

Зауваження 1. Розміри фігур повинні бути більшими, ніж розміри малюнків, оскільки малюнки слід розташовувати строго у середині фігур. Якщо під час накладання рисунка на фігуру рисунок



зникає, перемістіть його на передній план командою його контекстного меню Bring To Front.

4.5.5. Двічі клацніть на першому таймері і запрограмуйте блимання повідомлення "Обов'язково натисніть на кнопку "Готово" після розміщення рисунків!!!" так:

```
void __fastcall TForm1::Timer1Timer(TObject *Sender)
{
    if(Label ->Visible)           // Якщо поле світиться
        Label->Visible=false;    // поле гаситься,
    else
        Label->Visible=true;     // інакше-поле засвічується
}
```

Запустіть програму і переконайтесь, що напис на формі блимає.

4.5.6. Уведіть опис глобальних змінних програми:

Для цього у вікні тексту програми після існуючого опису змінної форми введіть такі рядки:

```
TForm1 *Form1;           // Цей рядок вже є в заготовці модуля
bool click=false;       // Наявність клацання миші
int shiftX, shiftY;     // Координати миші всередині малюнка
int x1,x2,x3,y;         // Початкові координати малюнків
enum info{right,wrong,more};
// Результати тесту
```

4.5.7. Вставте малюнки factory.bmp, shipping.bmp і finance.bmp з папки C:\ Program Files \Common Files \ Borland Shared \ Images\ Splash \ 256Color у об'єкти Image1, Image2 і Image3 відповідно. Вставте підписи до цих малюнків (див. рис.4.1).

4.5.8. Запрограмуйте конструктор форми TForm1::TForm1 і функцію Result() повідомлення результатів тесту:

```
__fastcall TForm1::TForm1(TComponent* Owner): TForm(Owner)
{
    x1=Image1->Left; // Зберігаємо початкові координати
    x2=Image2->Left; // лівих верхніх кутів
    x3=Image3->Left; // трьох малюнків у формі
    y=Image1->Top;
}
```



```
void Result(info msg)
{
    // Зупиняємо таймери:
    Form1->Timer1->Enabled=false;
    Form1->Timer2->Enabled=false;
    // Зчитуємо звуковий файл
    if(msg==right) Form1->MediaPlayer1->FileName="tada.wav";
        else Form1->MediaPlayer1->FileName="ding.wav";
    // Відкриваємо плеєр
    Form1->MediaPlayer1->Open();
    // Відтворимо звук
    Form1->MediaPlayer1->Play();
    // Відкриваємо вікно повідомлення
    if(msg==right)Application-
>MessageBoxA("Правильно!", "Повідомлення ",MB_OK);
        if(msg==wrong) Application-
>MessageBoxA("Неправильно!", "Повідомлення",MB_OK);
        if(msg==more)Application->MessageBoxA("Спробуйте ще
раз!", "Повідомлення",MB_OK);
    Form1->Timer1->Enabled=true;           // Запускаємо таймери
    Form1->Timer1->Enabled=true;
    Form1->ProgressBar1->Position=100; // Індикатор стану – на
                                                початок
    Form1->Image1->Left=x1;           // Повертаємо малюнки у
    Form1->Image2->Left=x2;           // вихідне положення
    Form1->Image3->Left=x3;
    Form1->Image1->Top=y;
    Form1->Image2->Top=y;
    Form1->Image3->Top=y;
}
```

Функція Result() не є методом класу Form1. Тому зміна властивостей будь-яких об'єктів форми в ній відбувається шляхом явного зазначення імені форми Form1 перед іменем відповідного об'єкта, наприклад, Form1->Timer1->Enabled.

4.5.9. Забезпечте перетягування першого малюнка на формі, запрограмувавши опрацювання таких трьох подій для об'єкта



Image1: OnMouseDown (Якщо Натиснути Мишу), **OnMouseMove** (Якщо Тягнути Мишу) та **OnMouseUp** (Якщо Відпустити Мишу). Для заповнення заготовок наведених функцій скористайтесь закладкою Events вікна інспектора об'єктів першого малюнка.

```
void __fastcall TForm1::Image1MouseDown(TObject *Sender,  
    TMouseButton Button, TShiftState Shift, int X, int Y)  
{  
    click=true; // click- це індикатор натискання на клавішу миші  
    shiftX=X; // Запам'ятовує координату (X,Y) точки  
    shiftY=Y; // кляцання мишею всередині малюнка  
}
```

```
void __fastcall TForm1::Image1MouseMove(TObject *Sender,  
    TShiftState Shift, int X, int Y)  
{  
    if (click) { // Якщо натиснута клавіша миші, змінюємо  
                // координати малюнка на величину  
                // зміни координати  
        Image1->Left=Image1->Left + X – shiftX; // вказівника миші  
        Image1->Top=Image1->Top + Y – shiftY; // з урахуванням його  
                                                // зміщень у середині  
                                                // малюнка shiftX, shiftY;  
    }  
}
```

```
void __fastcall TForm1::Image1MouseUp(TObject *Sender,  
    TMouseButton Button, TShiftState Shift, int X, int Y)  
{  
    // Запам'ятовуємо, що клавіша миші відпущена  
    click = false;  
}
```

Зверніть увагу, що програма не перевіряє, на яку саме клавішу миші натиснув користувач, і тому перетягування можна здійснювати будь-якою клавішею. Перевірити клавіші миші можна, проаналізувавши аргумент Button (типу TMouseButton) наведених функцій: Button = {mbLeft (ліва клавіша), mbRight (права), mbMiddle



(середня)}. Координати вказівника миші у пікселях передаються у функції за допомогою аргументів X і Y цілого типу.

4.5.10. Аналогічно запрограмуйте відповідні події для перетягування двох інших малюнків.

4.5.11. Створіть функцію `ImageInShape` для перевірки розташування малюнка (об'єкта `Image`) в середині деякої геометричної фігури (об'єкта `Shape`).

```
bool ImageInShape(TImage*I,TShape*S)
```

```
{  
    if((I->Left>=S->Left)&&  
        (I->Left+I->Width<=S->Left+S->Width)&&  
        (I->Top>=S->Top)&&  
        (I->Left+I->Height<=S->Left+S->Height))  
        return true; // Малюнок є всередині фігури  
    else  
        return false; // Малюнок поза фігурою  
}
```

4.5.12. Запрограмуйте кнопку "Готово", яка перевіряє правильність розташування малюнків усередині фігур:

```
void __fastcall TForm1::Button1Click(TObject *Sender)  
{  
    // Якщо перший малюнок є у другій фігурі  
    if(ImageInShape(Image1,Shape2)&&  
        // другий малюнок у третій фігурі  
        ImageInShape(Image2,Shape3)&&  
        // третій малюнок у першій фігурі  
        ImageInShape(Image3,Shape1))  
        Result(right);  
    else  
        Result(wrong);  
}
```



4.5.13. Запрограмуйте другий таймер, який забезпечує індикацію часу виконання від 100 до 0% зі швидкістю, обернено пропорційною до числа в CSpinEdit1.

У випадку, коли користувач не встиг розташувати малюнки у прямокутниках і натиснути на кнопку "Готово" у відведений час, подається звук "ДІНГ" і відкривається інформаційне вікно з повідомленням "Спробуйте ще раз!".

```
void __fastcall TForm1::Timer2Timer(TObject *Sender)
{
    // Задаємо швидкість руху індикатора
    Timer2->Interval=CSpinEdit1->Value*50;
    // Працює індикатор часу
    ProgressBar1->Position=ProgressBar1->Position-1;
    // Якщо час вичерпано 'Спробуйте ще раз!'
    if(ProgressBar1->Position==0)
        Result(more);
}
```

4.5.14. Збережіть програму та проект у робочій папці.

4.5.15. Скопіюйте музичні файли ding.wav та tada.wav у свою робочу папку з папки C:\WINDOWS\MEDIA.

4.5.16. Запустіть програму. Перетягуйте мишею малюнки в середину відповідних фігур.

4.5.17. Забезпечте можливість перетягування малюнків лише лівою клавішею миші (див. п. 4.5.9).

4.5.18. Збільшіть удвічі частоту блимання тексту "*Обов'язково натисніть на кнопку "Готово" після розміщення рисунків!!!*".

4.5.19. Замініть у програмі індикатор стану типу ProgressBar на індикатор типу Gauge.

4.5.20. Підберіть найвдалішу швидкість руху індикатора часу і зробіть об'єкт SpinEdit1 недоступним для внесення змін.

4.5.21. Вилучіть об'єкт SpinEdit1 і задайте найвдалішу швидкість руху індикатора часу безпосередньо в проекті.

4.6. Вимоги до звіту з лабораторної роботи

Звіт повинен містити:



- Титульний лист із зазначенням номеру варіанту
- Мету роботи
- Завдання
- Текст програми
- Результат виконання програми, скопійований з монітора комп'ютера

4.7. Контрольні запитання

1. Для чого призначена Геометрична фігура Shape?
2. Які властивості має Індикатор стану CGauge?
3. Яке значення властивості Kind відповідає круговій діаграмі стану CGauge?
4. Які значення може приймати аргумент Button?

Список рекомендованої літератури

1. Інформатика : комп'ютерна техніка. Комп'ютерні технології. посіб. / за ред. О. І. Пушкаря. К. : Видавничий центр "Академія", 2001. 696 с.
2. Інформатика. Базовий курс / Симонович С.В. и др. СПб : Питер. 640 с. : ил.
3. Інформатика. Комп'ютерна техніка. Комп'ютерні технології : підручник. К. : Каравела, 2003. 464 с.
4. Мешков А., Тихомиров Ю. Visual C++ и MFC + дискета. / пер. с англ. 2-е перераб. И доп. Санкт-Петербург : БХВ-Петербург, 2002. 1040 с.
5. Макарова М. В., Карнаухова Г. В., Запара С. В. Інформатика та комп'ютерна техніка : навчальний посібник / за ред. к.е.н., доц. М. В. Макарової. Суми : ВТД "Університетська книга", 2003. 642 с.