# Лабораторна робота № 1

## Створення програми «Моя перша форма»

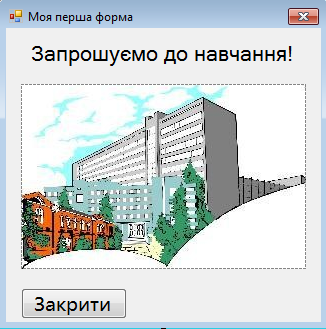
**Мета:** Модифікація форми. Робота з компонентами Label, PictureBox та Button. Програма закриття вікна.

**План**

* Об’єктно-орієнтоване програмування. Середовище програмування PascalABC.
* Створення проекту у PascalABC.

**Завдання:** Потрібно створити програму «Кольорове вікно з картинкою».

Приблизний вигляд вікна показано на малюнку 1.

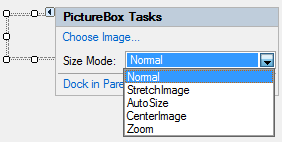


Процес створення програми відбувається за наступним алгоритмом:

1. Запустити PascalABC з пункту меню Windows або з «Рабочого столу»;
2. Зберегти проект в робочому каталозі командою «*Файл/Новий проект*».  **Рекомендація:** кожен проект краще зберігати в окремій папці; в іменах файлів можна використовувати тільки латинські букви без пробілів!;
3. Налаштування вікна форми:

* Заголовок вікна змінити на *Моя перша форма* 🡪 властивість **Text;**
* Змінити властивість **StartPosition** на *CenterScreen* для виводу вікна форми в центрі екрана;
* Змінити колір вікна на свій «улюблений колір», вибравши колір зі списку 🡪 властивість **BackColor;**
* Змінити іконки вікна програми 🡪 властивість **MaximizeBox** і **MinimizeBox**,Встановити всі підвластивості у значення *False* (в програмі відсутні усі системні кнопки вікна);

1. Помістити на форму картинку 🡪 об’єкт **Picture** на панелі Палітра *Стандартные элементы управления*, іконка значка . На формі з’явиться місце майбутньої картинки. Картинка може відображатися декількома способами: або займати місце, яке визначає користувач (рамка картинки) або змінити масштаб картинки під реальні розміри картинки. Для налаштування картинки потрібно змінити наступні властивості:



* Свойство **Choose Image** загружает «мастера» для выбора картинки из файла. Выберите любой файл с картинкой из имеющихся на жестком диске.
* Свойство **Size Mode –** при значении *Normal* значения установленные пользователем для картинки (свойства **Width** – ширина и **Height** – высота) будут изменены в соответствии с реальными размерами картинки;
* Свойство **Stretch Image** – при значении *true* размеры картинки будут адаптированы к размерам установленных пользователем (указанных в свойствах **Width** и **Height**);
* Свойство**AutoSize**приводит изображение к реальному размеру картинки (как в файле).

1. Поместить на форме кнопку для выхода из программы. Установить кнопку 🡪 объект **Button,** на панели Палитра *Стандартные элементы управления*, с иконкой объекта . Изменить свойства:
   * **Name** – изменить на *BQuit;*
   * **Text** – изменить на «Закрити»;
   * **Font –** шрифт текста, запускает мастера для настройки названия, цвета и стиля шрифта метки (установите любые параметры).
2. Поместить на форму метку для подписи к картинке 🡪 объект **Label**, на панели Палитра *Стандартные элементы управления*, значок . Изменить свойства:
   * **Text** – изменить на любой;
   * **TextAlign** – изменить на любой;
   * **Font –** шрифт текста, запускает мастера для настройки названия, цвета и стиля шрифта метки (установите любые параметры).
3. Запрограммировать выход при щелчке по кнопке:
   * Активный элемент, то есть элемент, свойства которого настраиваются, выделен рамкой с маркерами Ramka, если рамки нет ни на одном элементе, то активна форма. Выбрать элемент **кнопки** «Закрити» на форме (вокруг кнопки будет рамка);
   * Дважды щелкнуть по кнопке мышкой – будет выполнено событие *Click* на вкладке *Events* в **Инспектор Объектов** (это событие является событием «по умолчанием»). Станет активным окно **Редактора программ.**
   * В верхней части окна активной станет вкладка Код. Чтобы потом перейти снова к окну формы активируйте вкладку Дизайнер.



* + Автоматически будет создана процедура для программирования события *Click.* текст. *Пример*, если была использована кнопка *BQuit,* текст будет следующим:

**procedure** FMain.BQuit\_Click (sender: Object; e: EventArgs);

**begin**

**end;**

* + В теле созданной процедуры (между операторными скобками **begin** и **end)** вписать команду закрытия окна – **Close**. Так как у нас одно окно, то закрытие окна приведет к выходу из программы. Текст программы станет таким:

**procedure** FMain.BQuit\_Click (sender: Object; e: EventArgs);

**begin**

Close;

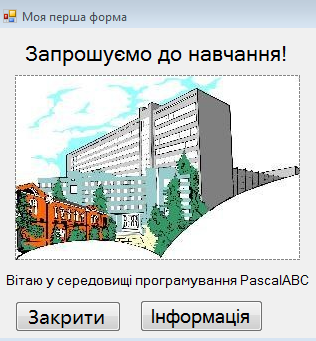
**end;**

**Обратите внимание!** После слова «Close» стоит знак «;», так как на языке Паскаль необходимо ставить знак «точка с запятой» – «;» после каждой команды.

1. Программа создана! Для запуска программы необходимо:
   * Сохранить программу 🡪 команда *Файл/Сохранить Все*, значок в меню **Главного окна PascalABC**;
   * Запустить программу 🡪 *Выполнить/Выполнить,* в меню **Главного окна PascalABC** или щелкнуть на значок  на *Панели инструментов* **в Главном окне PascalABC;**
2. Выход из оболочки **PascalABC** осуществляется выбором команды *Файл\Выход*.
3. Если при следующем запуске **PascalABC** потребуется загрузить проект, то это можно осуществить через пункт *Файл/Открыть проект*, значок Open ProjectM в **Главном окне PascalABC**.

### Самостійна робота

Додати до форми ще одну кнопку. При натиснені цієї кнопки на формі з’являється надпис – «Вітаю у середовищі програмування PascalABC».



**Вказівка:**

1. Встановити властивість **Visible** – False для напису; встановити ім’я напису – **Label2**;

2. Для кнопки «Інформація» при події *Click* змінити видимість текстового поля. Написати рядок програми для кнопки «Інформація»:

|  |
| --- |
| **if** Label2.Visible=True  **then** Label2.Visible:=False  **else** Label2.Visible:=True; |

1.2. Типи даних. Засоби введення та виведення

**Мета:** Робота з компонентами Label, textBox та Button.

**План**

* Поняття «тип даних». Основні типи даних середовища програмування PascalABC.
* Панель компонентів. Компоненти Label, textBox та Button.

**Практичне завдання:**

Написати програми з використанням розглянутих компонентів (Label, textBox та Button):

А) при запуску програми запускається форма 1 (з попереднього розділу) потім кнопкою відкривається форма 2.

Б) На формі 2 користувачеві пропонується ввести власне ім’я, вік та вагу. Крім цього на формі має бути кнопка «Введення». Після введення необхідної інформації користувач має натиснути на кнопку. Після чого праворуч від введеної інформації з’являються компоненти Label синього кольору із інформацією про типи даних (текст чи число), що відповідають змінним ім’я, вік та вага.

В) Удосконалити програму Б додавши на форму ще один компонент Button при натисненні на який виконувалося б очищення полів Ім’я та ін.

Приблизний вигляд вікна програми показано на рисунку 1.

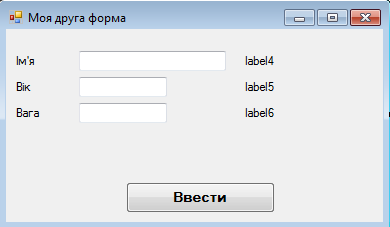
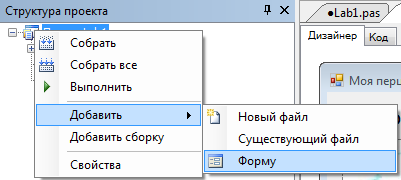


Рис. 1.

Створення програми відбувається за наступним алгоритмом:

1. Запустити PascalABC з пункту меню Windows або з «Рабочого столу»;
2. Відкрити проект1 в робочому каталозі командою «*Файл/Відкрити проект*».
3. Створення нового вікна форми:

* Викликати контекстне меню на назві проекту у вікні «Структура проекту». Вибрати команду Додати 🡪 Форму. З’явиться ще вікно ще однієї форми. Дати назву (або залишити за змовчуванням).

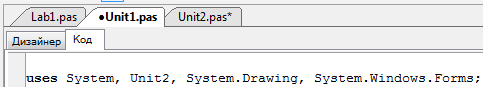


* Заголовок вікна змінити на *Моя друга форма* 🡪 властивість **Text;**
* Змінити властивіть **Name** форми на *Form2* , для звернення до неї в програмі.
* Змінити властивість **StartPosition** на *CenterScreen* для виводу вікна форми в центрі екрана;
* Змінити колір вікна на свій улюблений, властивість **BackColor;**
* Змінити іконки вікна програми **MaximizeBox** і **MinimizeBox**;

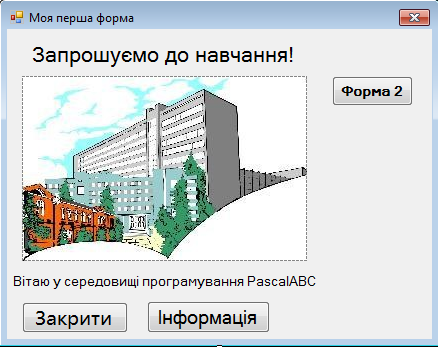
1. Активізувати вкладку основної програми (назва проекту.pas) і в рядок uses дописати назву нового юніта (нової форми).



1. Зробимо перехід з форми 1 у форму 2 по кнопці. Для цього під’єднаємо форму 2 у першу форму:
   * На вкладці Unit1 – Код допишемо у розділ Uses назву нової форми Unit2.



* + На вкладці Дизайнер поставити нову кнопку 🡪 об’єкт **Button**, на панелі Палітра *Стандартні елементи керування* . Змінити текст кнопки на «Форма 2».



* + Подвійним кліком на кнопці викликати обробку події *Click* і ввести текст програми:

|  |
| --- |
| **procedure** Form1.button2\_Click(sender: Object; e: EventArgs);  **begin**  Form(**new** Form2).show; // відкриття Form2  **end**; |

Створення форми 2.

1. Перейти до вкладки Unit2. Помістити на форму Підписи даних «Ім’я», «Вік» та «Вага» 🡪 объект Label, на панелі Палітра *Стандартні елементи керування*, значок . Змінити властивості: **Text; TextAlign; Font**.
2. Помістити на форму Вікна введення даних для «Ім’я», «Вік» та «Вага» 🡪 об’ект **TextBox** на панелі Палітра *Стандартні елементи керування*, іконка значка . На формі повинні з’явитися текстові поля.
3. Оскільки біля кожного з вікон має в майбутньому з’явитися напис, то для них потрібно підготувати місце: вставити напроти кожного з правої сторони ще по одному об’єкту типу **Label**. Встановити для них властивість **Visible** – False.



**Рекомендація**: можна створити одну трійку (вікна введення та написів до нього), встановити властивості, а потім виділити разом і утримуючи кнопку Ctrl перетягнути мишею по формі – скопіювати потрібну кількість разів. Змінити назви підписів.

1. На форму встановити кнопку «Введення».
2. Для кнопки «Введення» написати програму для події *Click*:

|  |
| --- |
| **procedure** Form2.button1\_Click(sender: Object; e: EventArgs);  **var**  s: string;  i: integer;  r: real;  **begin**  **try**  s:=(textBox1.Text);  i:=StrToInt(textBox2.Text); // перевід значення Box2 в ціле число  r:=StrToFloat(textBox3.Text); // перевід значення Box3 в дробове число  label4.Text:='Текст';  label5.Text:='Ціле число';  label6.Text:='Дробове число'  **except**  MessageBox.Show('Введіть інші дані', 'Помилка', MessageBoxButtons.OK);  **end**;  **end**; |

Команда **try ... except** робить спробу виконати усі команди, які стоять після слова **try**. Якщо це не вдається – формується виключення (**except**) і виконуються усі команди з цього розділу. Ця команда не дає програмі закінчитися в аварійному режимі, а дозволяє виправити дані.

### Самостійна робота

1. Зробіть компоненти Label, які з’являються на екрані, синього кольору.

2. Удосконалити програму, додавши на форму ще один компонент Button при натисненні на який виконувалося б очищення полів Ім’я та ін.

**Контрольні питання:**

1. Який тип даних ми називаємо *integer*?
2. Чи можна виконати операцію ділення над цілим типом даних?
3. До якого типу даних відносяться:

А) кількість книжок

Б) вартість товару

В) назва вулиці

Г) правдивість інформації

Д) об’єм кімнати

1. Які властивості компонента **Button** ви знаєте?

## Висновки

В першій темі були вивчені: о*болонка PascalABC; структура програми; команди редактора.*

При створенні програми були отримані навички роботи з наступними командами і візуальними компонентами: вікном форми **Form;** кнопкою **Button;** малюнком **Image;** міткою **Label;** командою закриття вікна **Close.**