Миколаївський обласний інститут післядипломної педагогічної освіти

Кафедра теорії та методики природничо-математичної освіти та інформаційних технологій

**Випускна робота на тему:**

**Вивчення окремих тем шкільного курсу за методикою технологічного навчання**

**Виконала**: слухач курсів

підвищення кваліфікації

Вердиш Ірина Іванівна

вчитель інформатики

Миколаївської загальноосвітньої

школи І-ІІІ ступенів № 35

Миколаївської міської ради

Миколаївської області

**Науковий керівник**:

Гапиченко Г.Є.

методист кафедри

Миколаїв

2016

# **Зміст**

Оглавление

[**Зміст** 2](#_Toc468830197)

[**Вступ. Технологічне навчання.** 3](#_Toc468830198)

[**Використання задачно-інструктивного підходу при вивченні програмування в Лазарусі** 5](#_Toc468830199)

[**Використання задачно-технологічного підходу при вивченні програмування в Лазарусі** 11](#_Toc468830200)

[**Використання проблемного при вивченні програмування в Лазарусі** 13](#_Toc468830201)

[**Додаток 1.** 18](#_Toc468830202)

[**Лазарус. Теоретична частина**. 18](#_Toc468830203)

[**1.** **TForm** 18](#_Toc468830204)

[**2.** **Компонент TLabel** 19](#_Toc468830205)

[**3.** **Компонент TEdit** 21](#_Toc468830206)

[**5.** **Компонент TMemo** 24](#_Toc468830207)

[**6.** **Компонент TRadioButton** 26](#_Toc468830208)

[**7.** **Компонент TCheckBox** 28](#_Toc468830209)

[**8.** **Компонент TListBox** 30](#_Toc468830210)

[**9.** **Компонент TComboBox** 32](#_Toc468830211)

[**10.** **Компонент TImage** 34](#_Toc468830212)

[**11.** **Компонент TTimer** 36](#_Toc468830213)

[**12.** **Компонент TBitBtn** 37](#_Toc468830214)

[**Основные свойства** 37](#_Toc468830215)

[**13.** **Компонент TRadioGroup** 39](#_Toc468830216)

**Вступ. Технологічне навчання.**

У зв’язку зі стрімким розвитком інформатичних технологій важливою складовою сучасної технологічної освіти стає інформаційно-технологічна освіта (**ІТ-освіта**). На відміну від фундаментальної освіти, заснованої на знаннєвій парадигмі та спрямованої на формування системи фундаментальних знань з інформатики, ІТ-освіта заснована на **компетентнісній парадигмі**. ЇЇ метою та результатом є сформованість інформаційно-комунікаційної компетентності учнів.

Однією з базових компетентностей, необхідних сучасній людині, є **технологічна компетентність**. Ця необхідність зумовлена високим рівнем розвитку технологій у будь-якій сфері життєдіяльності людини. Потреба сучасної людини в опануванні й оволодінні певними технологіями і привела до появи технологічної освіти. Метою технологічної освіти є навчання створення певних продуктів з необхідного матеріалу. Процес створення такого продукту саме і є **технологією**, яка, в свою чергу, складається з двох етапів: проектування й виготовлення.

Найбільш поширеними **методами технологічного навчання** є репродуктивні (відтворення способів дій, робота за зразком, інструкцією, тощо) та продуктивні (метод проектів, дослідницькі методи, творчі роботи).

**Основним видом роботи** при технологічному навчанні є практична робота.

Формування інформаційно-технологічних умінь передбачає застосування таких **методичних підходів**:

1. **Формально-операційних підхід.**

Це робота над відпрацюванням навичок роботи. Найменш ефективний для формування технологічних компетентностей, тому що учні просто заучують матеріал без належного рівня осмислення куди його можна потім застосувати.

1. **Задачно-інструктивний підхід**.

Це робота за технологічною картою. Тут присутнє спрямування на кінцевий результат, тобто учень з початку роботи націлений на створення будь-якого продукту, але за поданим зразком та описаною технологією.

1. **Задачно-технологічний підхід.**

Учні самостійно створюють інформаційно-технологічний продукт згідно з проектом, планом, з урахуванням поставлених вимог. Вони здатні обирати технологію виконання завдання й можуть застосовувати свої технологічні вміння в новій ситуації.

1. **Проблемний підхід.**

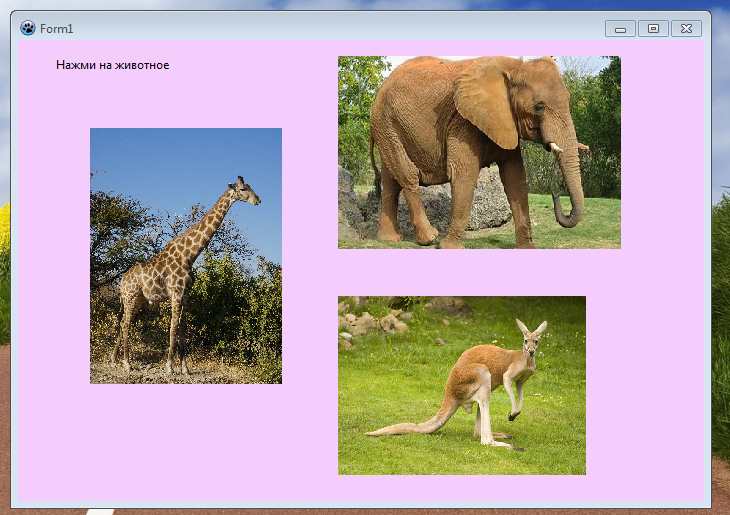
Це пошукова діяльність учнів. Але для неї вони повинні мати достатній рівень технологічних знань та вмінь. Самі сформулювали задачу, обговорили ідею з вчителем, намітили план виконання та самі задачу розв’язали.

Найбільш важливими та корисними з технологічної точки зору є задачно-інструктивний та задачно-технологічний підхід. Саме їм повинна віддаватися перевага у старшій школі. Проблемний підхід корисно використовувати для формування творчого мислення учнів, а також, як мотиваційний аспект для самостійного бажання здобувати знання.

# **Використання задачно-інструктивного підходу при вивченні програмування в Лазарусі**

**Задача**

Створити форму з 3 малюнками звірів. Зробивши клік на будь-якому з малюнків, ми повинні отримати вікно з додатковою інформацією про цю тварину. Також на формі, що відкриється повинна бути передбачена кнопка закриття цього вікна.



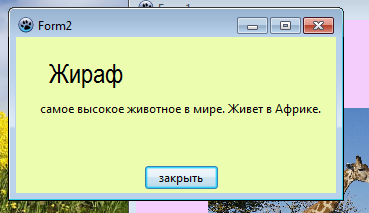
**Технологія виконання.**

Нам знадобляться: Form1 (Image1-3, Label1), Form2 (Label1-2, Button1), Form3 (Label1-2, Button1), Form4 (Label1-2, Button1).

Form1:

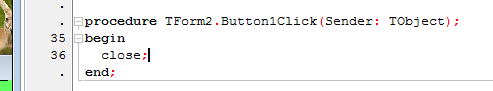
|  |  |  |
| --- | --- | --- |
| Компонент | Властивість | Значення |
| Form1 | Color | Pink |
| Image1 | Picture  Stretch | Загрузить  True |
| Image2 | Picture  Stretch | Загрузить  True |
| Image3 | Picture  Stretch | Загрузить  True |
| Label1 | Caption | Нажми на животное |

Form2:

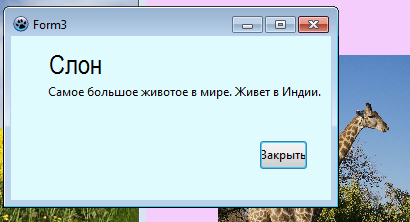


|  |  |  |
| --- | --- | --- |
| Компонент | Властивість | Значення |
| Form2 | Color | Yellow |
| Label1 | Caption  Font | Жираф  Arial 16 |
| Label2 | Caption | Самое высокое… |
| Button1 | Caption | закрыть |

Двічі нажимаємо на кнопку та прописуємо код:



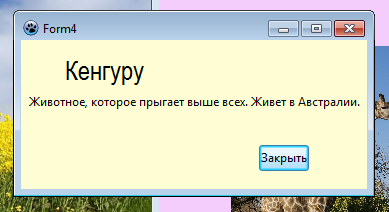
Form3:



|  |  |  |
| --- | --- | --- |
| компонент | Властивість | Значення |
| Form3 | Color | Blue |
| Label1 | Caption  Font | Слон  Arial 24 |
| Label2 | Caption | Самое большое… |
| Button1 | Caption | Закрыть |

Двічі нажимаємо на кнопку та пишемо: ‘close;’.

Form4:

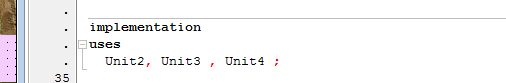


|  |  |  |
| --- | --- | --- |
| Компонент | Свойство | Значение |
| Form4 | Color | Orange |
| Label1 | Caption  Font | Кенгуру  Arial 24 |
| Label2 | Caption | Животное… |
| Button1 | Caption | Закрыть |

Двічі нажимаємо на кнопку та пишемо: ‘close;’.

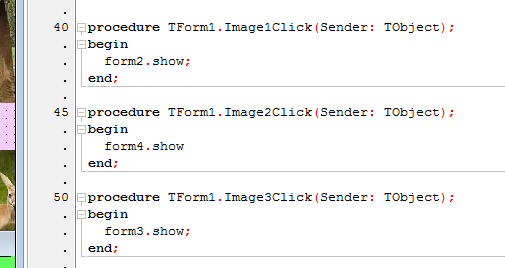
Тепер нам потрібно з’єднати вікна та налагодити їх відкриття. Для цього:

Вводимо код

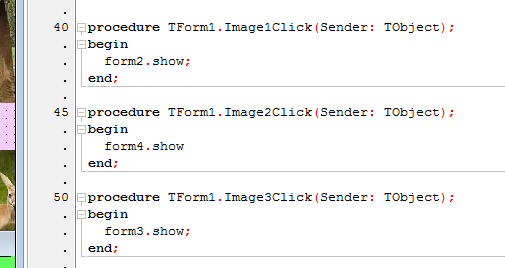


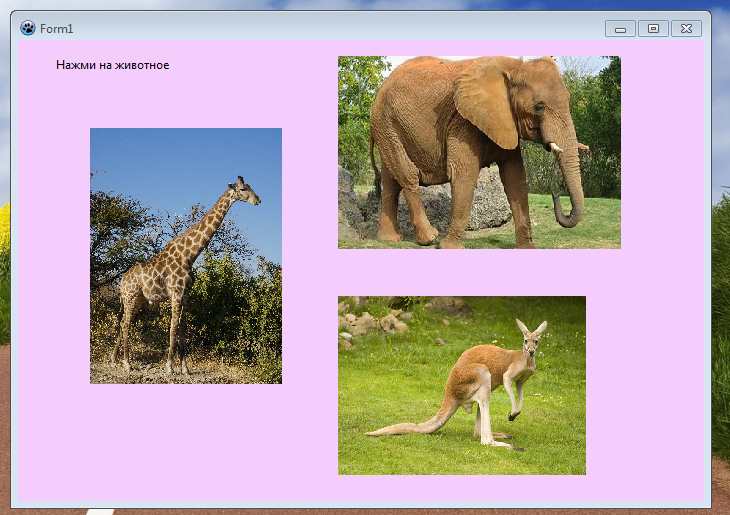
Потім натискаємо на Image1 і у подіях вибираємо OnClick.

Прописуємо такий код:

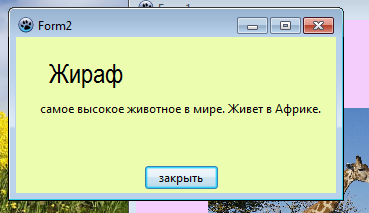


Для малюнка 2 та 3 робимо аналогічно, змінюючи form2 на form3 та form4:

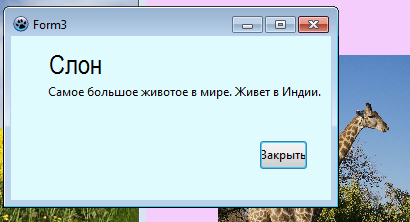


Компілюємо та користуємося програмою)

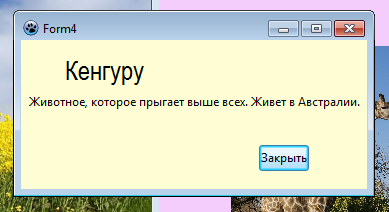
Натискаємо на жирафа. З’являється вікно:



Натискаємо Закрити та натискаємо на слона:



Натискаємо Закрити та натискаємо на кенгуру:



# **Використання задачно-технологічного підходу при вивченні програмування в Лазарусі**

**Задача.**

Створити програму за поданим зразком:

В текстовому полі набираємо назву місяця. Натискаємо кнопку УЗНАТЬ. У формі повинен відтворитися один з 12 малюнків, де показані свята цього місяця.

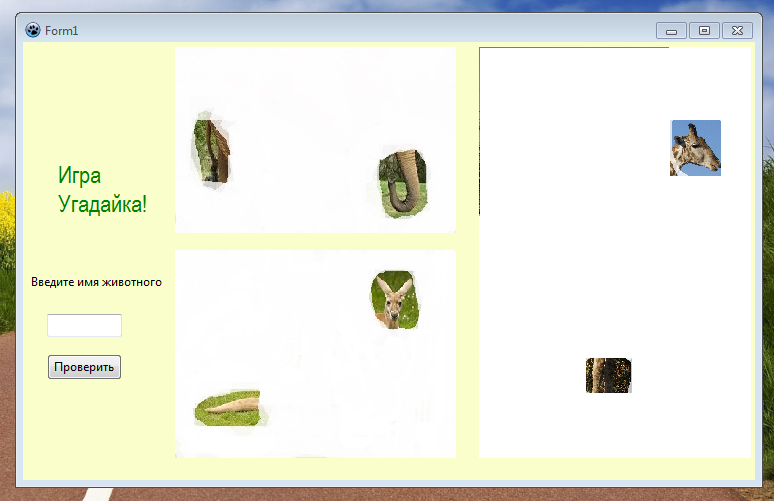
# **Використання проблемного при вивченні програмування в Лазарусі**

**Задача.**

З огляду на попереднє завдання потрібно придумати задачу, сформулювати її умову, скласти структуру своєї задачі та реалізувати її. Використати текстове поле, кнопку, мітку з поясненням, що потрібно робити користувачеві та поле для малюнка.

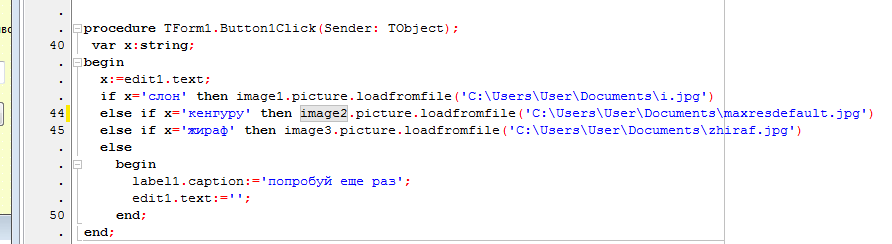
**Варіант розробки проекту та технологічної карти учнем:**

Гра Угадайка

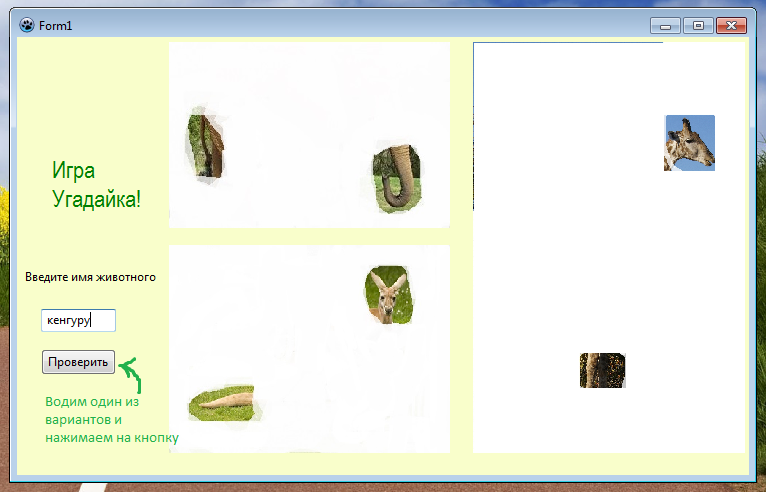


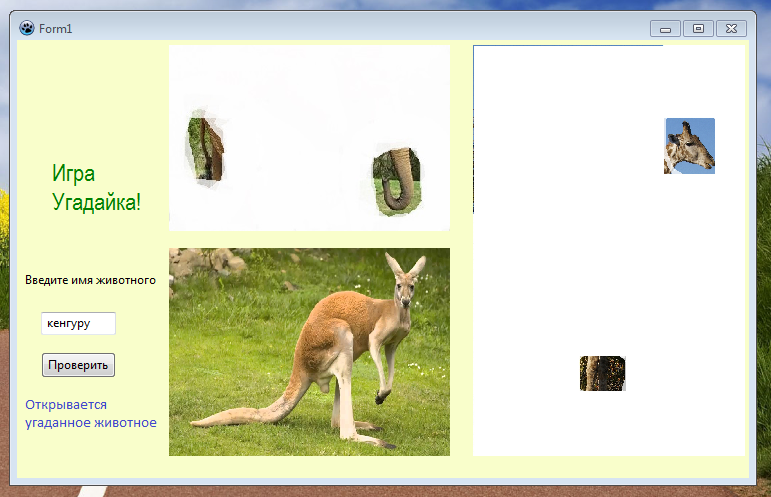
Нам знадобляться: 3 малюнки, оброблені в Paint, Image1-3, Button1, Label1-2, Edit1.

|  |  |  |
| --- | --- | --- |
| Компонент | Властивість | Значення |
| Form1 | Color | Orange |
| Image1 | Picture  Stretch | загрузить  True |
| Image2 | Picture  Stretch | Загрузить  True |
| Image3 | Picture  Stretch | Загрузить  True |
| Label1 | Caption  Font | Игра Угадайка!  Зеленый 24 Arial |
| Label2 | Caption | Введите имя… |
| Button1 | Caption | Проверить |
| Edit1 | Text | пусто |

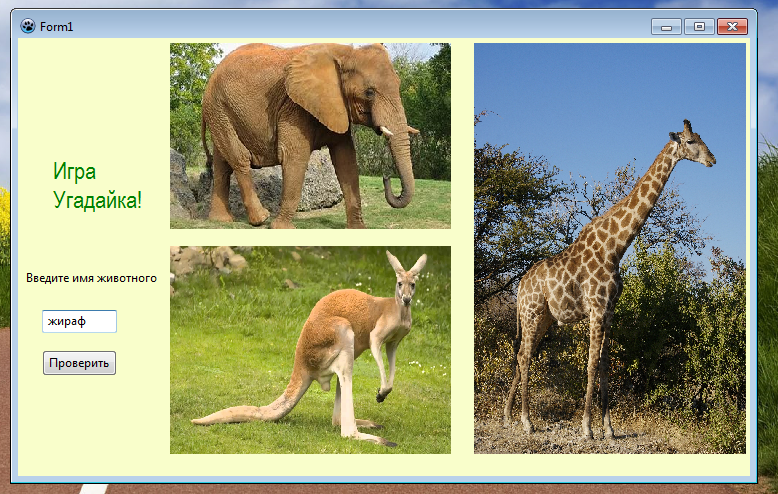
Двічі натискаємо на кнопку та програмуємо код:

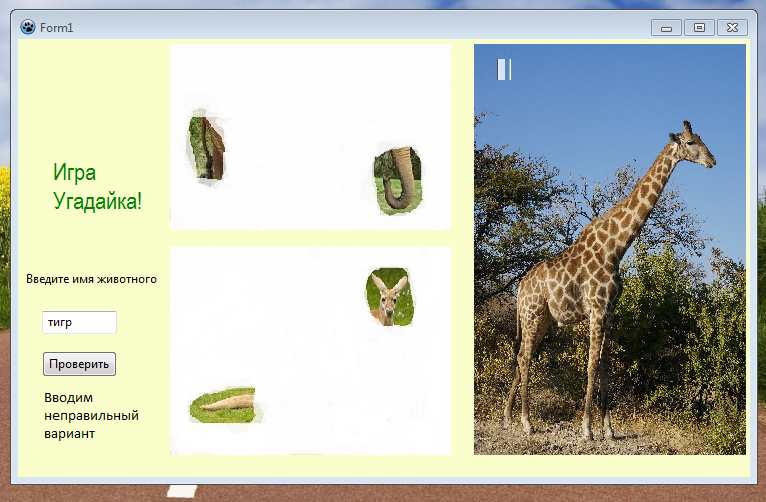
Компілюємо та граємо))

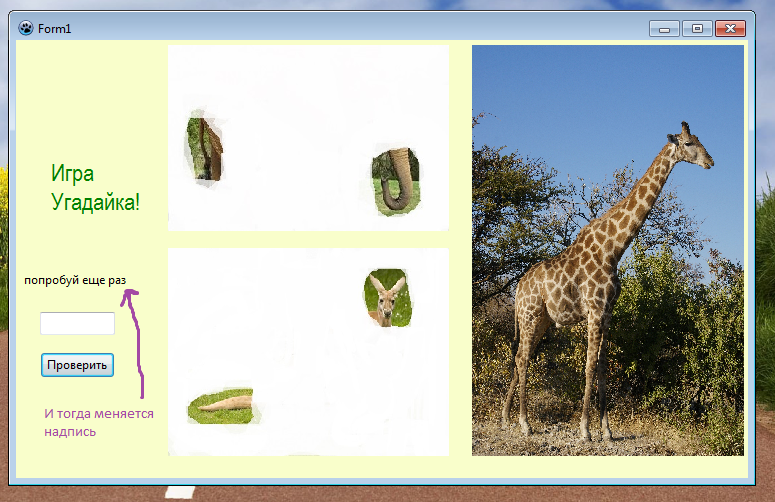




Коли ви вгадаєте всіх тварин, відкриються усі малюнки.







**Заключна частина.**

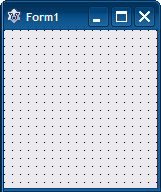
На уроках, на яких використовуються методи технологічного навчання, роль вчителя – це роль консультанта. Майже всю роботу діти виконують самостійно. Вчитель лише підбиває підсумки роботи за урок, узагальнює знання та вміння учнів, формуючи тим самим сталі навички роботи.

**Додаток 1.**

**Лазарус. Теоретична частина**.

1. **TForm**

Форма (об’єкт типа TForm) є основою програми. Властивості форми зумовлюють зовнішній вигляд вікна.

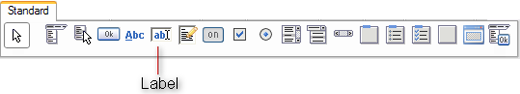


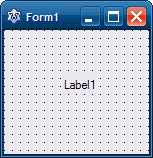
Основні властивості форми

|  |  |
| --- | --- |
| ***Властивість*** | ***Опис*** |
| **Name** | Им’я форми. Використовується для керування формою та доступу до компонентів форми. |
| **Caption** | Текст заголовка вікна. |
| **Top** | Відстань від верхньої межі форми до верхньої межі екрана. |
| **Left** | Відстань від лівої межі форми до лівої межі екрана. |
| **Width, Height** | Ширина, висота форми. |
| **Icon** | Значок у заголовку діалогового вікна, який позначає кнопку виводу системного меню. |
| **Color** | Колір фону. |
| **Font** | Шрифт. Шрифт, що використовується по «замовченню» для компонентів на поверхні форми. |
| **Canvas** | Поверхня, на яку виводять графіку. |

1. [**Компонент TLabel**](http://gospodaretsva.com/komponent-tlabel.html)

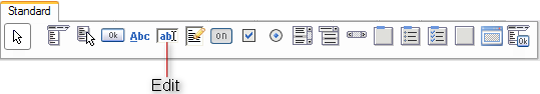
Компонент Label (Напис) використовують для виводу на форму тексту, який користувач не має можливості змінити під час виконання програми.

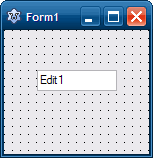


[](http://gospodaretsva.com/wp-content/uploads/2013/08/Label-2.png)

|  |  |
| --- | --- |
| *Властивість* | *Опис* |
| Name | *Им’я компонента. Використовується для керування формою та доступу до компонентів форми.* |
| Caption | *Текст, що відображується в полі напису.* |
| Left | *Відстань від лівої межі поля виводу до лівої границі форми.* |
| Top | *Відстань від верхньої межі поля виводу до верхньої межі форми.* |
| Width,Height | *Ширина, висота поля виводу.* |
| AutoSize | *Ознака того, що розмір поля визначається його вмістом.* |
| WordWrap | *Ознака того, що слова, які не вміщуються у поточному рядку, автоматично переносяться на наступний рядок (значення властивості AutoSize повинно бути False).* |
| Alignment | *Задає спосіб вирівнювання тексту всередині поля: taLeftJustify - вирівнювання по лівому краю; taCenter – вирівнювання по центру; taRightJustify –  Вирівнювання по правому краю* |
| Font | *Параметри шрифту, що використовуються для відображення текста: Font.Name – вид шрифту; Font.Size – розмір шрифту; Font.Color – колір шрифту.* |
| ParentFont | *Ознака наслідування компонентом характеристик шрифту форми, на якій знаходиться компонент. Якщо значення властивості дорівнює True, то текст виводиться шрифтом, що установлений для форми.* |
| Color | *Колір фону області виводу тексту.* |
| Transparent | *Керує відображенням фону області виводу тексту. Значення True робить область виводу тексту прозорою, (область не зафарбовується кольором, що заданий властивістю Color).* |
| Visible | *Дозволяє приховати текст (False) або зробити його видимим (True).* |

1. [**Компонент TEdit**](http://gospodaretsva.com/komponent-tedit.html)



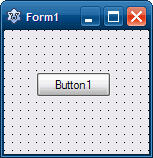


Компонент (TEdit) являє собою поле вводу-редагування рядка символів.

|  |  |
| --- | --- |
| *Властивість* | *Опис* |
| Name | *Ім’я компоненту. Використовується в програмі для доступу до компонента та його властивостям, а саме для доступу до тексту, що введений в полі редагування.* |
| Text | *Текст, що знаходиться в полі вводу та редагування.* |
| Left | *Відстань від лівої межі компонента до лівої межі форми.* |
| Top | *Відстань від верхньої границі компонента до верхньої границі форми.* |
| Width, Height | *Ширина, висота поля.* |
| Font | *Шрифт, що використовується для відображення тексту, який ми вводимо.* |
| ParentFont | *Ознака наслідування компонентом характеристик шрифту форми, на якій знаходиться компонент. Якщо ознака властивості дорівнює* True*, то при зміні властивості* Font *форми автоматично змінюється значення властивості* Font *компонента.* |
| Enabled | *Використовується для обмеження можливості змінити текст у полі редагування. Якщо значення властивості дорівнює* False *, то текст в полі редагування змінити не можна.* |
| Visible | *Дозволяє приховати текст (*False*) або зробити його видимим (*True*).* |

1. [**Компонент TButton**](http://gospodaretsva.com/komponent-tbutton.html)

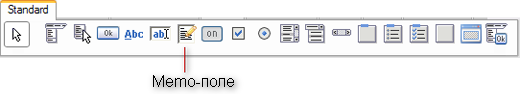


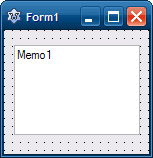
[](http://gospodaretsva.com/wp-content/uploads/2013/08/button-2.png)

Компонент Button (Кнопка) – командна кнопка, з допомогою якої користувач має змогу визвати виконання будь-якої дії.

|  |  |
| --- | --- |
| *Властивість* | *Опис* |
| Name | *Им’я компонента. Використовується для доступу до компонента та його властивостей.* |
| Caption | *Текст на кнопці.* |
| Left | *Відстань від лівої границі кнопки до лівой границі форми.* |
| Top | *Відстань від верхньої границі кнопки до верхньої границі форми.* |
| Width, Height | *Ширина, висота кнопки.* |
| Enabled | *Ознака досяжності кнопки.* True – *кнопка досяжна* False – *кнопка недосяжна. Наприклад, в наслідок клацання на кнопці подія* Click *не наступає.* |
| Visible | *Дозволяє приховати текст. False – текст бачимо. True – текст не бачимо.* |
| Hint | *Контекстна підказка – текст, який з’являеться поряд з покажчиком мишки при наведенні покажчика (для того, щоб текст з’явився, потрібно щоб значення властивості ShowHint було True).* |
| ShowHint | *Дозволяє (*True*) або не дозволяє (*False*) відображення підказки при наведенні покажчика на кнопку.* |

1. [**Компонент TMemo**](http://gospodaretsva.com/komponent-tmemo.html)



[](http://gospodaretsva.com/wp-content/uploads/2013/08/memo-2.png)

Основне призначення компонента TMemo – робота з великою кількістю рядків (ввод, відображення та редагування текстового материалу).

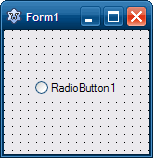
Для роботи с буфером обміну можливо використовувати загальноприйняті гарячі клавіши: Ctrl-X — вирізати, Ctrl-C — копіювати; Ctrl-V — вставити.

|  |  |
| --- | --- |
| **Властивість** | **Опис** |
| Name | *Ім’я компоненту. Використовується в програмі для доступу до компонента та його властивостей, а саме для доступу до тексту, що введений в полі редагування.* |
| Text | *Текст, що знаходиться в полі Memo. Разглядається як одне ціле.* |
| Lines | *Масив рядків, що відповідає вмісту поля. Доступ до рядка здійснюється за номером. Нумерація рядків здійснюється з нуля.* |
| Lines.Count | *Кількість рядків тексту в полі Memo.* |
| Left | *Відстань від лівої границі поля до левої границі форми.* |
| Top | *Відстань від верхньої границі поля до верхньої границі форми.* |
| Width, Height | *Ширина, висота поля.* |
| Font | *Шрифт, що використовується для відображення тексту, який ми вводимо.* |
| ParentFont | *Ознака наслідування властивостей шрифту батьківської форми* |
| WantReturns | *Клавиша для вводу кінця рядку: TRUE –  клавіша ENTER; FALSE –клавіші CTRL + ENTER.* |
| WordWrap | *Перехід на початок наступного рядка при вводі довгих рядків: TRUE – робиться автоматично; FALSE – не робиться. При включеній горизонтальній полосі прокрутки ця властивість игнорируєтся.* |
| ScrollBar | *Використання полоси прокрутки, якщо текст великий і не вміщується в компонент Memo: ssNone – Немає полоси прокрутки; ssHorizontal – Встановлена горизонтальна прокрутка; ssVertical – Встановлена вертикальна прокрутка; ssBoth – Встановлені дві полоси прокрутки.* |
| ReadOnly | *Дозволяє або не дозволяє редагування тексту. (Однак, програмно текст можна додавати).* |

Для збереження вмісту текстового поля Memo в файл використовується функція SaveToFile(‘mytetxt.txt’), а для витягу -LoadFromFile(‘mytetxt.txt’), де mytetxt.txt – текстовий файл, що розташований в каталозі програми.

1. [**Компонент TRadioButton**](http://gospodaretsva.com/komponent-tradiobutton.html)





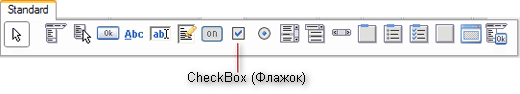
Компонент TRadioButton (перемикач) на відміну від прапорця (CheckBox), дозволяє вибрати лише один з можливих варіантів.

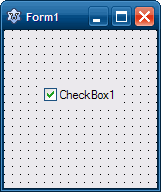
Коли користувач встановлює один з перемикачів, усі інші перемикачі групи автоматично знімаються.

Якщо потрібно организувати декілька груп перемикачів, то кожну групу потрібно представити компонентом TRadioGroup.

|  |  |
| --- | --- |
| *Властивість* | *Опис* |
| Name | *Ім’я компоненту. Використовується в програмі для доступу до компонента та його властивостей* |
| Caption | *Текст, який знаходиться зправа від кнопки.* |
| Checked | *Стан, зовнішній вигляд кнопки: якщо кнопка обрана, то значення* True *, якщо ні – значение* False*.* |
| Left | *Відстань від лівої границі прапорця до левої границі форми.* |
| Top | *Відстань від верхньої границі прапорця до верхньої границі форми.* |
| Width, Height | *Ширина, висота поля виводу пояснюючого тексту* |
| Font | *Шрифт, що використовується для відображення поясняюючого тексту.* |
| ParentFont | *Ознака наслідування характеристик шрифту батьківської форми.* |

1. [**Компонент TCheckBox**](http://gospodaretsva.com/komponent-tcheckbox.html)

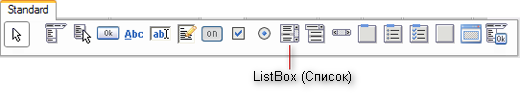


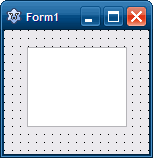
[](http://gospodaretsva.com/wp-content/uploads/2013/08/CheckBox-2.png)

*Компонент Прапорець (*TCheckBox*) дає користувачу два варіанти вибору – його можливо встановити або зняти. Встановлений прапорець відмічається галочкою. Коли прапорці поєднані в групу, користувач має можливість встановити або зняти будь-які прапорці групи. Одночасно може бути включено декілька прапорців.*

|  |  |
| --- | --- |
| *Властивість* | *Опис* |
| Name | *Ім’я компоненту. Використовується в програмі для доступу до компонента та його властивостей* |
| Caption | *Текст, що знаходиться справа від прапорця.* |
| Checked | *Стан, зовнішній вигляд прапорця: якщо прапорець встановлений (в квадратику є «галочка»), то значення* True; *якщо прапорець скинуто (немає«галочки»), то* False*.* |
| State | *Стан прапорця. На відміну від властивості* Checked*, дозволяє розрізняти встановлений, скинутий і проміжний стан. Стан прапорця визначає одна з констант:* cbChecked *(встановлений);* cbGrayed *(сірий, невизначений стан);* cbUnChecked *(скинутий).* |
| AllowGrayed | *Властивість визначає, чи має прапорець можливість бути у невизначеному стані: якщо* AllowGrayed = False*, то прапорець може бути тільки встановленим або скинутим, якщо* AllowGrayed = True, *то допускається проміжний стан.* |
| Left | *Відстань від лівої границі прапорця до левої границі форми.* |
| Top | *Відстань від верхньої границі прапорця до верхньої границі форми.* |
| Width, Height | *Ширина, висота поля виводу пояснюючого текста* |
| Font | *Шрифт, що використовується для відображення пояснюючого тексту.* |
| ParentFont | *Ознака наследування характеристик шрифту батьківської форми.* |

1. [**Компонент TListBox**](http://gospodaretsva.com/komponent-tlistbox.html)

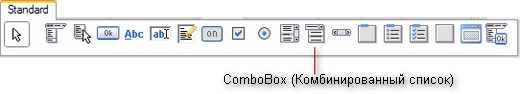


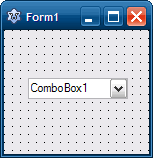


Компонент (TListBox) являє собою список, в якому можнаобрать потрібний елемент.

|  |  |
| --- | --- |
| *Властивість* | *Опис* |
| Name | *Ім’я компоненту. Використовується в програмі для доступу до компонента та його компоненту и его свойствам.* |
| Items | *Елементи списку – масив рядків* |
| Count | *Кількість елементів списку* |
| Sorted | *Ознака необхідності автоматичної сортировки (*True*) післе додавання чергового елемента.* |
| ItemIndex | *Номер вибраного елемента. Елементи списку нумеруються з нуля. Якщо в списку жоде з елементів не обрали, то значення дорівнює минус 1.* |
| Left | *Відстань від лівой границі списка до лівої границі форми.* |
| Top | *Відстань від верхньої границі списка до верхньої границі форми.* |
| Width | *Ширина поля списку.* |
| Height | *Висота поля списку.* |
| Font | *Шрифт, що використовується для відображення елементів списку.* |
| ParentFont | *Ознака наслідування властивостей шрифту батьківської форми.* |

1. [**Компонент TComboBox**](http://gospodaretsva.com/komponent-combobox.html)

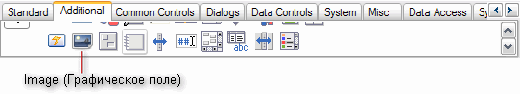


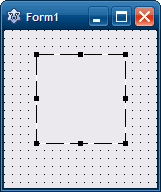


Компонент (TComboBox) дає можливість ввести дані в поле редагування шляхом набору на клавиатурі або вибором із списку.

|  |  |
| --- | --- |
| *Властивість* | *Опис* |
| Name | *Ім’я компоненту. Використовується в програмі для доступу до компонента та його властивостей* |
| Text | *Текст, що знаходиться в полі ввода-редагування* |
| Items | *Елементи списку – масив рядків* |
| Count | *Кількість елементів списку* |
| Sorted | *Ознака необхідності автоматичної сортировки (*True*) після додавання чергового елементу.* |
| ItemIndex | *Номер обраного елемента. Елементи списку нумеруються з нуля. Якщо в списку жоден з елементів не був обраний, то значення дорівнює мінус 1.* |
| DropDownCount | *Кількість елементів, що відтворюються в розкритому списку. Якщо кількість елементів списка більша, ніж* DropDownCont *, то з’являється вертикальна полоса прокрутки.* |
| Left | *Відстань від лівої границі компонента до левої границі форми.* |
| Top | *Відстань від верхньої границі компонента до верхньої границі форми.* |
| Width | *Ширина компоненту.* |
| Height | *Висота компоненту (поля ввода-редагування).* |
| Font | *Шрифт, що використовується для відображення елементів списку.* |
| ParentFont | *Ознака наслідування властивостей шрифту батьківської форми.* |

1. [**Компонент TImage**](http://gospodaretsva.com/komponent-timage.html)

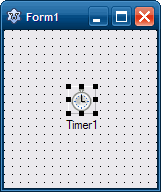




*Компонент (*TImage*) забезпечує вивод на поверхню форми ілюстрацій, що представлені в* bmp*–форматі (щоб компонент можливо було використовувати для відображення ілюстрацій у форматі* JPG*, потрібно підключити модуль* JPEG *– вказати им’я модуля в директиві* uses*).*

|  |  |
| --- | --- |
| ***Властивість*** | ***Опис*** |
| Picture | Ілюстрація, яка відображається в полі компонента. |
| Width, Height | Розмір компонента. Якщо розмір компонента менше, ніж розмір ілюстрації, та значення властивостей AutoSize, Strech и Proportional дорівнює False, то відображується частина ілюстрації. |
| Proportional | Ознака автоматичного масштабування картинки без спотворення. Щоб масштабування було виконано, значення властивості AutoSize повинно бути False. |
| Strech | Ознака автоматичного масштабування (стиснення обо розтягнення) ілюстрації відповідно до реального розміру компонента. Якщо розмір компонента не пропорціональний розміру ілюстрації, то ілюстрація буде спотворена.  Увага: властивість Stretch не впливає на файли малюнків типа .ico. |
| AutoSize | Ознака автоматичної зміни розміру компонента відповідно до реального розміру ілюстрації. |
| Center | Ознака визначає розташування малюнка в полі компонента по горизонталі, якщо ширина малюнка менше ширини поля компонента. Якщо значення властивості дорівнює False, то малюнок притиснутий до правої границі компонента, якщо True – то малюнок розташовується по центру. |
| Visible | Чи відображається компонент і, відповідно, ілюстрація на поверхні форми. |
| Canvas | Поверхня, на яку можливо вивести графіку. |

## [**Компонент TTimer**](http://gospodaretsva.com/komponent-ttimer.html)

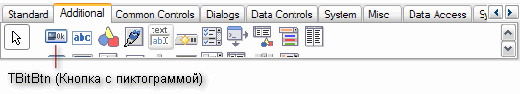
  


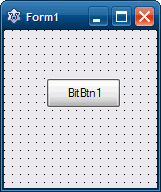
За допомогою таймера (Timer) можливо запрограмувати виконання будь-якого кода через рівні проміжки часу. Коли таймер встановлений на формі, система періодично генерує подію OnTimer. Для користувача таймер невидимий.

|  |  |
| --- | --- |
| *Властивість* | *Опис* |
| Name | *Ім’я компоненту. Використовується в програмі для доступу до компонента та його властивостей* |
| Interval | *Інтервал часу між генерацією подій OnTimer, він виражений в мілісекундах (мс). Відлік часу починається з моменту встановлення властивості*Enabled *в* True*.* |
| Enabled | *Дозвіл роботы. При значенні* True *таймер включається,* False *– виключається.* |

Щоб відключити таймер, потрібно присвоїти властивості Enabled значення False або властивості Interval  – значення 0

1. [**Компонент TBitBtn**](http://gospodaretsva.com/komponent-tbitbtn.html)





Компонент Кнопка (TBitBtn) – різновид кнопки, відрізняється від TButton тим, що на ньому можливо відображати піктограми.

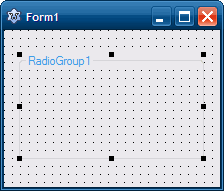
Властивості для TButton мають місце і для TBitBtn. Крім того, цей компонент має і свої особливі властивості.

|  |  |
| --- | --- |
| **Властивість** | **Опис** |
| Kind | Задає тип кнопки. Є декілька зумовлених типів кнопки з готовой піктограмою та текстом:⎫ bkAbort – з текстом “Прервати”.⎫ bkAll – з текстом “Все”.⎫ bkCancel – з текстом “Відміна”.  ⎫ bkClose – з текстом “Закрити”.  ⎫ bkCustom – будь-який текст, що задає програмувальник.  ⎫ bkHelp – з текстом “Довідка”.  ⎫ bkIgnore – з текстом “Пропуск”.  ⎫ bkNo – з текстом “Ні”.  ⎫ bkNoToAll – з текстом “Ні для всіх”.  ⎫ bkOK – з текстом “ОК”.  ⎫ bkRetry – з текстом “Повторення”.  ⎫ bkYes – з текстом “Так”.  ⎫ bkYesToAll – з текстом “Так для Всіх”. |
| Glyph | Якщо вас не задовольняють допустимі малюнки, вы маєте можливість обрати інші. Буде відкрите діалогове вікно, необходно вказати путь до цього малюнку. |
| Margin | Задає відстань від края кнопки до малюнка (в пікселах). По замовченню -1. В цьому випадку малюнок та текст розташовуються по центру. |
| Layout | Задає положення малюнка на кнопці. Можна выбрати:  ⎫ blGlyphLeft – зліва.  ⎫ blGlyphRight – справа.  ⎫ blGlyphBottom – знизу.  ⎫ blGlyphTop – зверху. |
| Spacing | Задає відстань в пикселах між малюнком та текстом кнопки. |

1. [**Компонент TRadioGroup**](http://gospodaretsva.com/komponent-tradiogroup.html)

Компонент TRadioGroup (Група перемикачів) знаходиться на вкладці Standard





 Компонент TRadioGroup (Группа перемикачів) представляє собою групу перемикачів, в якій одночасно може бути виділений тільки один перемикач.  Коли користувач встановлює один з перемикачів групи, всі останні автоматично скидуються.

Таким чином, будь-які два перемикачі можуть бути встановлені одночасно, тільки якщо вони розташовані в різних контейнерах, наприклад в різних групах перемикачів.

Радіокнопки в контейнері TRadioGroup створюються через властивість Items. В групі перемикачів кожний рядок властивості Items відносяться до окремого перемикача і є його заголовком.

|  |  |
| --- | --- |
| Властивість | Опис |
| Caption | Заголовок групи |
| Columns | Кількість столбців у групі перемикачів. По замовченню дорівнює 1. |
| ItemIndex | Визначає номер (починаючи з 0) перемикача, який є виділеним у групі. Якщо спочатку не виділений жодний перемикач, то значення властивості дорівнює -1. |
| Item | Містить перелік заголовків перемикачів групи. Для їх ввода передбачений спеціальний редактор, який викликається кліком на специальну кнопку Кнопка для формирования списка в редакторе строк, розташовану справа в рядку, що описує властивість Items. |

**Процес створення будь-якого додатка можливо розділити на наступні етапи:**

* Створення проекту. В результаті на екрані з’являється пуста форма (вікно майбутнього додатку).
* Створення графичного інтерфейса проекта – розташування необхідних елементів, задання розмірів, змінення властивостей;
* Написання програмного коду, який визначить, що буде робити ваша  програма.
* Отладка програми.

