

# КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БУДІВНИЦТВА І АРХІТЕКТУРИ

## ДІАЛОГОВІ СИСТЕМИ І ЛІНГВІСТИЧНЕ ЗАБЕЗПЕЧЕННЯ

---

кафедра прикладної математики



Методичні вказівки  
до індивідуальних занять для студентів спеціальності  
6.080402 «Інформаційні технології проектування»

# Зміст

1. Випадкові фігури .....	3
ЕТАПИ РОБОТИ.....	3
РЕКОМЕНДАЦІЇ ДО ВИКОНАННЯ .....	3
2. Анкета-тест .....	4
ЕТАПИ РОБОТИ.....	4
РЕКОМЕНДАЦІЇ ДО ВИКОНАННЯ .....	4
3. Вирішення квадратного рівняння .....	5
ЕТАПИ РОБОТИ.....	5
РЕКОМЕНДАЦІЇ ДО ВИКОНАННЯ .....	5
4. Пошук у тексті .....	6
ЕТАПИ РОБОТИ.....	6
РЕКОМЕНДАЦІЇ ДО ВИКОНАННЯ .....	6
5. Конвертер валют .....	7
ЕТАПИ РОБОТИ.....	7
РЕКОМЕНДАЦІЇ ДО ВИКОНАННЯ .....	7
6. Простий XML-PAD .....	8
ЕТАПИ РОБОТИ.....	8
РЕКОМЕНДАЦІЇ ДО ВИКОНАННЯ .....	8
7. Побудова графіків функцій .....	9
ЕТАПИ РОБОТИ.....	9
РЕКОМЕНДАЦІЇ ДО ВИКОНАННЯ .....	9
8. Букмекерська контора F1 .....	10
ЕТАПИ РОБОТИ.....	10
РЕКОМЕНДАЦІЇ ДО ВИКОНАННЯ .....	10
9. Площа плоских фігур .....	11
ЕТАПИ РОБОТИ.....	11
РЕКОМЕНДАЦІЇ ДО ВИКОНАННЯ .....	11
10. Визначення детермінанту матриці .....	12
ЕТАПИ РОБОТИ.....	12
РЕКОМЕНДАЦІЇ ДО ВИКОНАННЯ .....	12
11. Складний текстовий редактор.....	13
ЕТАПИ РОБОТИ.....	13
РЕКОМЕНДАЦІЇ ДО ВИКОНАННЯ .....	13

# 1. ВИПАДКОВІ ФІГУРИ

**Мета:** здобути базові навички роботи з GDI-графікою. Розробити програму, яка буде виводити задану кількість випадкових фігур.

**Максимальна оцінка:** 3–4 бали.

## ЕТАПИ РОБОТИ

1. Створити застосування на C++. Розробити оригінальний інтерфейс.
2. Забезпечити можливість вводу числа бажаних фігур.
3. На полотні побудувати вказане число випадкових фігур.

## РЕКОМЕНДАЦІЇ ДО ВИКОНАННЯ

1. Розмістити у вікні компонент для вводу числа фігур.
2. Розмістити кнопку, при натисканні на яку буде запущено процес побудови фігур.
3. Використовуючи в циклі функцію `Random` отримати спочатку тип фігури, а потім координати її центру. Наприклад `Random(2)` – 0 – коло, 1 – квадрат. `Random(300)` – координата X. Координата Y – аналогічно.
4. Для отримання більш високої оцінки необхідно розробити клас/класи і реалізувати програму з використанням об'єктно-орієнтованого програмування.

## 2. АНКЕТА-ТЕСТ

**Мета:** здобути базові навички розробки програм з діалоговою взаємодією. Розробити програму, яка буде проводити тестування на знання певної категорії інформації.

**Максимальна оцінка:** 3–4 бали.

### ЕТАПИ РОБОТИ

1. Створити застосування на C++. Розробити оригінальний інтерфейс.
2. Забезпечити можливість вибору варіанту відповіді на питання.
3. Підрахувати кількість правильних відповідей і видати кінцевий результат.

### РЕКОМЕНДАЦІЇ ДО ВИКОНАННЯ

1. Для кожного питання розробити фрейм, на якому буде саме питання і варіанти відповіді для нього у вигляді перемикачів.
2. Головне вікно оформити у вигляді майстра — кнопки назад, вперед. При натисканні кнопок здійснювати перехід між фреймами.
3. Перехід між фреймами організувати шляхом увімкнення/вимкнення їх видимості.
4. Бажано мати групу перемикачів для швидкого переходу до будь-якого попереднього питання.
5. В якості кінцевого результату привести всі питання і правильні відповіді на них, а також відповіді дані користувачем.
6. Для отримання більш високої оцінки необхідно розробити клас/класи і реалізувати програму з використанням об'єктно-орієнтованого програмування.

## 3. ВИРІШЕННЯ КВАДРАТНОГО РІВНЯННЯ

**Мета:** здобути базові навички розробки програм з діалоговою взаємодією. Розробити програму, яка буде виконувати розв'язок квадратного рівняння.

**Максимальна оцінка:** 3–4 бали.

### ЕТАПИ РОБОТИ

1. Створити застосування на C++. Розробити оригінальний інтерфейс.
2. Забезпечити можливість вводу коефіцієнтів рівняння.
3. Знайти всі корені рівняння, якщо вони існують.

### РЕКОМЕНДАЦІЇ ДО ВИКОНАННЯ

1. Для вводу коефіцієнтів квадратного рівняння використати компоненти `TEdit`.
2. Для взяття квадратного кореня використати функцію `sqrt` з модуля `Math.hpp`.
3. Всі основні математичні функції знаходяться в модулях `math.h` та `Math.hpp`.
4. Для отримання більш високої оцінки необхідно розробити клас/класи і реалізувати програму з використанням об'єктно-орієнтованого програмування.

## 4. ПОШУК У ТЕКСТІ

**Мета:** закріпити навички роботи з рядками. Розробити застосування, що реалізує пошук заданого рядка в тексті.

**Максимальна оцінка:** 4-5 балів.

### ЕТАПИ РОБОТИ

1. Створити застосування на C++. Розробити оригінальний інтерфейс.
2. Забезпечити можливість відкриття текстових файлів та їх перегляд.
3. Забезпечити можливість виконання наступних операцій:
  - пошук тільки цього слова (виразу) у тексті – режим пошук виразу;
  - пошук входжень цього слова (виразу) у інші вирази – режим повний пошук.
4. Відобразити результат пошуку.

### РЕКОМЕНДАЦІЇ ДО ВИКОНАННЯ

1. При виконанні роботи в середовищі *CodeGear RAD Studio 2009* використати для пошуку функції роботи з рядками `Pos` або `AnsiPos`.
2. В режимі пошуку тільки цього слова (виразу) перевірити чи до і після знайденого фрагменту присутні припустимі знаки розриву слова (пробіл, крапка, кома, тире, крапка з комою).
3. Можна застосувати стандартний компонент *CodeGear RAD Studio 2009*— `TFindDialog`.
4. Для отримання більш високої оцінки необхідно розробити клас/класи і реалізувати програму з використанням об'єктно-орієнтованого програмування.

## 5. КОНВЕРТЕР ВАЛЮТ

**Мета:** здобути базові навички розробки програм з діалоговою взаємодією. Розробити програму, яка буде виконувати функцію конвертації валют за заданим курсом.

**Максимальна оцінка:** 3–4 бали.

### ЕТАПИ РОБОТИ

1. Створити застосування на C++. Розробити оригінальний інтерфейс.
2. Забезпечити можливість вводу курсу валют.
3. Здійснювати не тільки пряму і зворотну конвертацію, а і крос-курсіву.

### РЕКОМЕНДАЦІЇ ДО ВИКОНАННЯ

1. Для вводу курсу валют використати компоненти `TEdit`. Курс задається тільки для гривні відносно всіх інших валют.
2. Для конвертації сумми грошей використати математичні функції модулів `math.h` та `Math.hpp`.
3. При розрахунку крос-курсу спочатку зводити перетворення до гривні, а потім в нову валюту.
4. Для отримання більш високої оцінки необхідно розробити клас/класи і реалізувати програму з використанням об'єктно-орієнтованого програмування.

## 6. ПРОСТИЙ XML-PAD

**Мета:** навчитись аналізувати текстову інформацію, та виконувати її форматування в залежності від змісту.

**Максимальна оцінка:** 4-5 балів.

### ЕТАПИ РОБОТИ

1. Знайти XML-документ, або створити його самому в будь-якому текстовому редакторі.
2. Створити застосування на C++. Розробити оригінальний інтерфейс.
3. Забезпечити можливість відкриття файлів XML та їх перегляд.
4. Забезпечити форматне відображення елементів XML-документу в залежності від їх призначення. Елементами, які слід виділяти спеціальним форматуванням являються:
  - символи «<» та «>»;
  - текст, що знаходиться між символами «<» та «>»;
  - цифри;
  - символи «"» та «'»;
  - текст, що знаходиться між символами «"» та «'»;
  - символ «=».
5. Забезпечити можливість користувачу задавати способи форматування того чи іншого елемента (колір, шрифт та його написання).

### РЕКОМЕНДАЦІЇ ДО ВИКОНАННЯ

1. При виконанні даної роботи рекомендується використати для відображення вмісту XML-документу компонент `TRichEdit`.
2. Спочатку XML-документ слід завантажити в компонент `TRichEdit`, а потім виконувати його форматування.
3. Для форматування використовувати пошук в тексті необхідних елементів, їх програмне виділення та застосування до них форматування.
4. Для виділення тексту використовувати властивості `SelStart` та `SelLength`.
5. Для форматування виділеного фрагменту використовувати властивість `SelAttributes`.
6. Властивість `SelStart` використовує індекс символу, з якого треба починати виділення тексту. Не забувати, що індекси символу в рядках мають наскрізну нумерацію, тобто для визначення позиції символу в  $n$ -ному рядку необхідно додати до його індексу в поточному рядку довжини всіх попередніх рядків. Причому до довжини рядка отриманої оператором `Length` треба додавати 2 (це недруковані символи кінця рядку та переводу каретки).
7. Для отримання більш високої оцінки необхідно розробити клас/класи і реалізувати програму з використанням об'єктно-орієнтованого програмування.



## 7. ПОБУДОВА ГРАФІКІВ ФУНКЦІЙ

**Мета:** закріпити навички роботи з GDI-графікою. Згадати курс вищої математики.

**Максимальна оцінка:** 3-4 бали.

### ЕТАПИ РОБОТИ

1. Створити застосування на C++. Розробити оригінальний інтерфейс.
2. Надати список типових математичних функцій ( $\sin x$ ,  $\cos x$ ,  $\operatorname{tg} x$ ,  $y=x^2$ ,  $y=x^3$ ,  $y=1/x^2$  і т. д.).
3. Дозволити користувачу задавати коефіцієнти для функцій (наприклад  $y=3x^2$ ).

### РЕКОМЕНДАЦІЇ ДО ВИКОНАННЯ

1. Можна скористатися засобами GDI-графіки для побудови графіків функцій. В циклі розраховувати положення точок і наносити їх на полотно.
2. При виконанні даної роботи в *CodeGear RAD Studio 2009* можна спробувати використати компонент `TTreeChar`.
3. Для отримання більш високої оцінки необхідно розробити клас/класи і реалізувати програму з використанням об'єктно-орієнтованого програмування.

## 8. БУКМЕКЕРСЬКА КОНТОРА F1

**Мета:** здобути базові навички розробки програм з діалоговою взаємодією. Розробити програму, яка буде виконувати розрахунок виграшу букмекерських ставок.

**Максимальна оцінка:** 4-5 балів.

### ЕТАПИ РОБОТИ

1. Створити застосування на C++. Розробити оригінальний інтерфейс.
2. Забезпечити можливість вибору прізвища пілота із списку.
3. Ввести кінцеві результати гонки.
4. Здійснити підрахунок виграних грошей кожним гравцем за правильно вгадані позиції пілотів на фініші.

### РЕКОМЕНДАЦІЇ ДО ВИКОНАННЯ

1. Для вводу прізвища пілота на конкретній позиції використовувати компонент `TComboBox`.
2. При виконанні ставки необхідно зберігати ім'я гравця, що зробив її.
3. Ставки всіх гравців показувати в окремому компоненті — (`TMemo`, `TRichEdit` або `TStringGrid`).
4. Для отримання більш високої оцінки необхідно розробити клас/класи і реалізувати програму з використанням об'єктно-орієнтованого програмування.

## 9. ПЛОЩА ПЛОСКИХ ФІГУР

**Мета:** здобути базові навички розробки програм з діалоговою взаємодією та роботи з GDI графікою. Розробити програму, яка буде виконувати розрахунок площі будь-яких плоских фігур.

**Максимальна оцінка:** 4-5 балів.

### ЕТАПИ РОБОТИ

1. Створити застосування на C++. Розробити оригінальний інтерфейс.
2. Забезпечити можливість вибору типу плоскої фігури (трикутник, трапеція, коло, прямокутник, ромб, сектор, сегмент).
3. Забезпечити можливість вводу всіх необхідних розмірів і параметрів. Зобразити графічно задану фігуру.
4. Здійснити розрахунок площі вибраної плоскої фігури та вивести результат.

### РЕКОМЕНДАЦІЇ ДО ВИКОНАННЯ

1. Для вибору типу фігури використати компонент `TSpeedButton` з нанесеними попередньо підготовленими картинками з позначенням параметрів, що необхідно ввести.
2. Для кожної фігури створити окремий фрейм на якому будуть перераховані всі необхідні параметри для розрахунку та формули розрахунку площі.
3. Після введення всіх необхідних параметрів здійснити розрахунок площі і зобразити фігуру на будь-якому компоненті, що має властивість `Canvas`.
4. Для отримання більш високої оцінки необхідно розробити клас/класи і реалізувати програму з використанням об'єктно-орієнтованого програмування.

## 10. ВИЗНАЧЕННЯ ДЕТЕРМІНАНТУ МАТРИЦІ

**Мета:** здобути навички об'єктно-орієнтованого програмування. Навчитися працювати з динамічними масивами.

**Максимальна оцінка:** 4-5 балів.

### ЕТАПИ РОБОТИ

1. Створити застосування на C++. Розробити оригінальний інтерфейс.
2. Забезпечити можливість вводу матриць будь-якої розмірності.
3. Розрахувати детермінант заданої матриці і вивести його значення.

### РЕКОМЕНДАЦІЇ ДО ВИКОНАННЯ

1. Створити клас матриці, який буде розподіляти потрібну кількість пам'яті під елементи (див. лекцію по C++).
2. Для введення елементів матриці використати компонент `TStringGrid`.
3. Створити метод класу, який буде за допомогою рекурсії розраховувати детермінант матриці методом викреслень.

## 11. СКЛАДНИЙ ТЕКСТОВИЙ РЕДАКТОР

**Мета:** закріпити навички роботи з рядками та методами їх форматування. Розробити застосування, що виконує функції текстового редактора.

**Максимальна оцінка:** 4-5 балів.

### ЕТАПИ РОБОТИ

1. Створити застосування на C++. Розробити оригінальний інтерфейс.
2. Забезпечити можливість відкриття і збереження текстових файлів.
3. Забезпечити можливість набору та форматування тексту (створення заголовків, зміна шрифту виділеного тексту, зміна написання шрифту, зміна кольору тексту).
4. Передбачити можливість пошуку та заміни слів і виразів.

### РЕКОМЕНДАЦІЇ ДО ВИКОНАННЯ

1. Використати компонент TRichEdit. Розібратися з його властивостями.
2. Дивитися рекомендації до робіт №4 та №6.