

# Реалізація та використання перерахувань у програмуванні мовами С та С++

Дідик Олексій, Комп-мат2

## Вступ

Перерахування (`enum`) є важливим інструментом для розробників, які працюють із мовами програмування С та С++. Вони дозволяють створювати читаємий і структурований код, замінюючи числові значення на зрозумілі імена. Це знижує ризик помилок і підвищує ефективність розробки програм.

Цей реферат розглядає програму, яка реалізує задачі на основі перерахувань, структур, текстових і числових обчислень. Програма реалізована у двох версіях: на С та С++. Мова С використовується для створення базового функціоналу, тоді як мова С++ демонструє її розширення та оптимізацію.

## Перша частина: Реалізація програми мовою С

### Функціональні можливості програми

Перша версія програми, написана мовою С, вирішує такі завдання:

1. **Друк значень перерахувань.** Здійснюється виведення назв днів тижня, місяців, кольорів спектру та шахових фігур. Наприклад, введення числа 0 для днів тижня поверне *Понеділок*.
2. **Робота зі структурою Card.** Структура представляє ігрову карту з полями `Suit` (масть) та `Rank` (ранг). Реалізовані функції для введення даних карти, їх виведення, збереження у файл і зчитування з файлу.
3. **Порівняння карт.** Функція визначає, чи одна карта може побити іншу, зважаючи на їх масть і ранг. Якщо карти однієї масті, порівняння виконується за рангом. У випадку наявності козирної масті, козирна карта автоматично перемагає.
4. **Конвертація довжин.** Значення довжин у міліметрах, сантиметрах, дециметрах, метрах і кілометрах переводяться у метри.
5. **Робота з текстами.** Здійснюється відмінювання слів за граматичними відмінками. Наприклад, слово "книга" у родовому відмінку набуде вигляду "книга (Родовий)".
6. **Підрахунок складів у тексті.** Функція рахує кількість голосних у реченні, введеному користувачем англійською мовою.
7. **Рух корабля.** Моделюється траєкторія руху корабля на основі введених команд (вперед, назад, ліворуч, праворуч). Також обчислюється загальна пройдена відстань за заданими швидкостями та часами.

## Технічна реалізація

Програма складається з трьох основних файлів:

- `Enumeration.h` — містить оголошення перерахувань (`enum`), структур (`Card`) та прототипів функцій.
- `Enumeration.c` — реалізує функціональність для роботи з перерахуваннями, картами та іншими задачами.
- `Enumeration_test.c` — тестовий файл для перевірки працездатності функцій.

Приклад реалізації функції друку:

```
void printDay(DaysOfWeek day) {  
    const char* days[] = {"", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"};  
    printf("%s\n", days[day]);  
}
```

## Друга частина: Реалізація програми мовою C++

### Переваги реалізації у C++

Версія програми на C++ зберігає основні можливості, але використовує сучасні інструменти цієї мови:

- `enum class`: підвищує типобезпеку перерахувань.
- `std::vector`: дозволяє працювати з динамічними масивами.
- `std::string`: полегшує роботу з текстовими даними.
- **Посилання** (`const &`): підвищують ефективність роботи зі структурами, зменшуючи копіювання.

### Основні задачі

1. **Друк значень перерахувань.** Функції виводять текстові назви днів, місяців, кольорів спектра та шахових фігур через масиви `std::string`.
2. **Порівняння карт.** Функція `beat` приймає карти як параметри-посилання, що знижує навантаження на пам'ять.
3. **Конвертація довжин.** Значення переводяться у метри через оператор `switch`.
4. **Робота з текстами.** Відмінювання слів реалізовано через конкатенацію рядків, а підрахунок складів — через ітерацію по символах у рядку `std::string`.
5. **Рух корабля.** Траєкторія моделюється через вектор команд, який можна динамічно змінювати.
6. **Обчислення відстані.** Використовується вектор швидкостей і часів для обчислення загальної відстані.

### Приклад функції для підрахунку складів:

```
int countSyllables(const string& sentence) {  
    int count = 0;  
    for (char ch : sentence) {  
        if (string("AEIOUYaeiouy").find(ch) != string::npos) {  
            ++count;  
        }  
    }  
    return count;  
}
```

## Висновок

Реалізація програми демонструє, як використання перерахунку дозволяє створювати структурований і зрозумілий код. У версії на C++ програма стала більш ефективною завдяки використанню сучасних можливостей мови.

### Основні переваги:

- **Чіткість:** зрозуміле використання перерахунку та структур.
- **Гнучкість:** можливість легко додавати нові функції.
- **Продуктивність:** оптимізація коду через об'єктно-орієнтовані можливості.

Програма може бути розширена шляхом впровадження об'єктно-орієнтованого підходу (наприклад, через створення класів для карти, корабля тощо). Це дозволить ще більше підвищити її функціональність.