

**Лекція 9.** Перевантаження методів. Перевантаження бінарних та унарних операторів. Стандартний клас рядок. Конструктори та методи класу рядок. Приклади використання рядків.

Наслідування. Типи наслідування в Cі++.

Наслідування та абстрактні класи. Віртуальні методи. Множинне наслідування та проблеми з ним пов'язані. Віртуальне наслідування.

#### **Питання.**

- 1) Що таке перевантаження методів? Чому воно зручно в мовах зі строгою типізацією?
- 2) Чим перевантаження операторів відрзняється від перевантаження інших методів?
- 3) Які оператори не можна перегружати? Коли перевантаження операторів може бути небезпечним?
- 4) Чому при перевантаженні операторів вводу-виводу нам потрібно ключове слово friend?
- 5) В файлі string.hpp приведений код, що реалізує інтерфейс класу рядок Cі++. Скільки конструкторів в цьому коді? Скільки копиконструкторів? Скільки та які оператори є перевантаженими?
- 6) Як видалити підрядок використовуючі методи класу String?
- 7) Які типи наслідування є на Cі++ та яка між ними різниця?
- 8) Поясніть на прикладі, що таке раннє та пізнє зв'язування
- 9) Що таке чисто віртуальний клас та чисто віртуальний метод? Коли вони потрібні?
- 10) Як реалізувати множинне наслідування на Cі++?
- 11) Що робити та які шляхи правильного множинного наслідування якщо й класи батьки й клас-син мають метод з однаковою назвою? Що зміниться, якщо це не метод, а перевантажений оператор?

#### **Вправи:**

- 1) В класі Рациональний дріб з попередньої лекції напишіть методи введення, виведення (cin>>, cout<<) та оператори віднімання, ділення як перевантажені оператори. Тобто з типом Рациональний дріб можна тепер працювати як зі стандартним типом. Чому краще перевантажити два оператори віднімання?
  - 2) Напишіть функцію часткового сплітінгу рядку. Тобто функція, що приймає рядок та повертає перше слово з рядку (роздільник – задається як аргумент функції)
  - 3) Напишіть функцію, що приймає рядок та повертає масив (як аргумент-змінний) всі дійсні числа, що містяться в рядку (роздільник – задається як аргумент функції)
  - 4) Створіть клас Людина (члени: ПІБ, стать, вік) та його наслідники Студент (додано: курс, група, ВУЗ), Викладач (додано: ВУЗ, посада, з.п.). Методи введення, виведення, конструктори для різної кількості вхідних даних.  
Створіть клас Аспірант, що є наслідником і студента і викладача. Коректно визначте член ВУЗ для нього.

Наслідування та віртуальні методи

- 5) Реалізувати наступні класи:

Описати клас **Прямокутник**. Сторони прямокутника паралельні осям координат. Для прямокутника задані лівий верхній кут та довжини сторін. Описати методи отримання довжини кожної з сторін, площі прямокутника, периметру, метод знаходження перетину двох прямокутників. Методи переміщення прямокутника. Скласти програму створення заданої кількості прямокутників та знаходження їх спільного перетину.

Описати клас **Трикутник**. Основа трикутника паралельна осі  $x$  координат. Для трикутника задані лівий нижній кут (координати) та довжини сторін. Описати методи отримання довжини кожної зі сторін. Описати методи отримання довжини кожної з сторін, площі прямокутника, периметру, метод знаходження перетину двох прямокутників. Методи переміщення прямокутника. Скласти програму створення заданої кількості прямокутників та знаходження їх спільного перетину.

Описати клас **Трикутник**. Основа трикутника паралельна осі  $x$  координат. Для трикутника задані лівий нижній кут (координати) та довжини сторін. Описати методи отримання довжини кожної зі сторін. Описати методи отримання довжини кожної з сторін, площі, периметру, метод знаходження перетину двох трикутників. Методи переміщення. Скласти програму створення заданої кількості трикутників та знаходження їх спільного перетину.

Описати клас **Еліпс**. Для нього задані фокуси та радіуси. Описати методи отримання геометричних характеристик. Описати методи отримання довжини радіусів, площі, периметру, метод знаходження площі перетину двох еліпсів. Методи переміщення та повороту. Скласти програму створення заданої кількості еліпсів та знаходження їх спільного перетину.

Створити клас **Фігура**, який є базою.

Опишіть класи для таких геометричних фігур та реалізуйте зазначені методи:

- a) Клас **Трапеція**. У цьому класі реалізуйте операції знаходження периметра і площі;
- b) Клас **Паралелограм**. У цьому класі реалізуйте операції знаходження периметра і площі.

- с) Клас Круг. Реалізуйте методи відшукування площі круга, довжини кола, цього круга.
- д) Клас Піраміда. Реалізуйте методи пошуку площі бічної поверхні і об'єму;
- е) Клас П'ятикутник, що містить масив вершин. Реалізуйте метод перевірки чи є цей п'ятикутник опуклим.
- ф) Клас Багатокутник. Реалізуйте метод перевірки чи є цей багатокутник опуклим.

Дано список фігур вищенаведених класів. Серед фігур, що належать до перших трьох класів знайдіть фігуру, що має найбільшу площу та периметр (довжину кола). Також знайдіть всі опуклі багатокутники

#### 6) Опишіть класи

1. **Гість**, що містить всю необхідну інформацію про жителя деякого готелю: ім'я, період проживання тощо.
2. **Кімната**, що містить інформацію про кімнату готелю у тому числі вартість проживання за добу.
3. **Готель**, що містить список кімнат цього готелю, інформацію про те ким і коли вони зайняті, а також методи на кшталт тощо.

Використовуючи вищенаведені класи розв'язати задачі:

- а) Вивести відомість про кількість вільних кімнат у готелі;
- б) Пошуку вільної кімнати у зазначений період;
- в) Поселити жителя на вказаний термін;
- г) Вартості проживання жителя у зазначений період;
- д) Прибутку, який отримає готель за вказаний період;
- е) Пошуку гостя у готелі (у заданий період);

7) Опишіть клас Фігура, що інкапсулює основні геометричні характеристики та методи. Для фігури визначено методи:

1. `calculateVolume()` – віртуальний метод, що обчислює міру фігури (для плоскої фігури – площу, для об'ємної – відповідно об'єм).
2. `getVolume()` – що повертає міру фігури.

Від класу Фігура наслідуються такі класи

- Трикутник
- Прямокутник
- Трапеція
- Паралелограм
- Круг
- Куля
- Трикутна Піраміда (який наслідується від класу Трикутник)
- Чотирикутна піраміда (який наслідується від класу Прямокутник)
- Паралелепіпед (який наслідується від класу Прямокутник)

Нехай дано список фігур. Серед заданих фігур, знайдіть фігуру, що має найбільшу міру якої є найбільшою

8) Опишіть клас `Pet` – домашня тварина, що має метод `to_feed(feed, count)` – годувати (`feed` – тип корму, `count` – кількість).

Клас `Pet` має віртуальні методи

1. `to_sniff()` («нюхати» – визначає, чи може їсти тварина заданий тип корму),
2. `to_ask()` («просити» – метод повертає `True`, якщо тип корму не підходить або тварина ще хоче їсти і виводить на екран прохання «тваринною мовою», наприклад, «Мяв...» для кота),
3. `to_eat()` (їсти, якщо тип корму підходить).

Клас `Pet` має нащадки – `Cat`, `Dog`, `Parrot` (папуга), у яких перевизначено вищезгадані віртуальні методи.

Задано список тварин та список кормів (тип та загальна вага). Пропонуючи по черзі кожній тварині порцію їжі, потрібно нагодувати всіх тварин. Якщо корму не вистачить – вивести відповідне повідомлення.

9) Опишіть клас Car, що має метод go(distance), який змінює пройдений кілометраж автомобілем та залишок пального. Метод go(...) залежить від віртуального методу fuelPerKm(), який визначає скільки потрібно пального автомобілю для проїзду одного кілометра. Нехай Personal (легковий автомобіль) і Truck (вантажівка) – класи, що наслідують клас Car і перевизначають метод fuelPerKm(). При цьому потрібно врахувати, що цей метод залежить від кількості пасажирів (+10% на кожного пасажирів) для авто класу Personal або ваги вантажу для Truck (+25% на кожен тону вантажу). Визначити чи зможе задане авто проїхати задану відстань.

10) Задано клас Flower, що має нащадками конкретні класи квітів (напр., тюльпан, троянд, тощо). Ви зайшли у квітковий магазин у якому продаються різні типи квітів. Необхідно зібрати букет з квітів (букет може містити квітки одного класу) та визначити:

1. Його вартість.
2. Скільки часу зможе тішити букет очі (до моменту поки не зів'яне перша квітка).
3. Колір, що домінує у цьому букеті.
4. Чи припустимий цей букет за інтенсивністю запаху.