

Об'єктно-орієнтоване програмування

Виконайте ОДНУ (на Ваш вибір) лабораторну роботу мовою програмування (на Ваш вибір) C++, JAVA або C#

1 ПРОСТІ КЛАСИ

2 Успавдкування класів

1 ПРОСТІ КЛАСИ

Мета роботи – навчитись будувати класи та використовувати їх при створенні програм.

1.1 Порядок виконання лабораторної роботи

1. Виконати завдання відповідно до номера варіанта. При виконанні програм слід використовувати конструктори та деструктори.
2. Оформити звіт.
3. Зробити висновки.

1.2 Варіанти завдання

Варіант 1. Створити клас для роботи з трьох вимірними векторами. Передбачити функції для виконання наступних операцій: консольне введення і виведення значень вектора; ініціалізація вектора; складання та віднімання векторів; обчислення довжини вектора.

Варіант 2. Створити клас *Person*. Змінна типу *Person* повинна вміщувати наступну інформацію про деяку людину: прізвище, ім'я та по батькові; адреса; рік народження; телефон; зріст, вага, стать. Передбачити функції для виконання наступних операцій: консольне введення і виведення значень типу *Person*; ініціалізація. Створити масив, що містить інформацію про деяку групу людей. Підрахувати кількість людей за статтю (наприклад - *friend*-функція).

Варіант 3. Створити клас *Alfa* таким чином, щоб при створенні першого і знищенні останнього об'єкта цього типу на екран видавалися відповідні повідомлення. Вказівка: застосувати статичні компоненти класу.

Варіант 4. Створити клас *Beta* таким чином, щоб при знищенні останнього об'єкта на екран видавалося повідомлення про найбільшу кількість об'єктів типу *Beta*, що існували та час існування популяції об'єктів. Вказівка: застосувати статичні компоненти класу.

Варіант 5. Створити клас *Delta* таким чином, щоб кожний об'єкт вміщував свій персональний номер (дескриптор об'єкта) та функцію, яка повертає його значення. Дескриптор об'єкта – унікальне для об'єктів даного типу ціле число.

Варіант 6. Створити клас таким чином, щоб він відстежував кількість існуючих у даний момент об'єктів та об'єм оперативної пам'яті, який вони займають. Вказівка: застосувати статичні компоненти класу.

Варіант 7. Створити клас для виконання операцій з комплексними числами. Передбачити операції: складання та віднімання; множення та

ділення; обчислення модулю; консольне введення та виведення; ініціалізацію.

Варіант 8. Створити клас *Organization*. Змінна типу *Organization* повинна вміщувати наступну інформацію: назва; адреса; директор; телефон. Передбачити функції для виконання наступних операцій: консольне введення і виведення значень типу *Organization*; ініціалізація. Створити каталог організацій міста та відсортувати за назвою (*friend*- функція).

Варіант 9. Створити клас *Book*. Зміна типу *Book* повинна містити наступні поля: назва; кількість авторів; автори; рік видання; кількість сторінок. Передбачити функції для виконання наступних операцій: ініціалізація книги, додавання авторів, консольне введення/виведення інформації про книгу. Створити каталог книг впорядкувати елементи по назві (*friend*- функція).

Варіант 10. Створити клас *Film*. Зміна типу *Film* повинна містити наступні поля: назва; рік створення; режисер; кількість акторів; актори. Передбачити функції для виконання наступних операцій: ініціалізація інформації, додавання акторів, консольне введення/виведення інформації про стрічку. Створити каталог стрічок, впорядкувати елементи по назві (*friend*-функція).

Варіант 11. Створити клас *House*. Зміна типу *House* повинна містити наступні поля: вулиця; номер; кількість квартир, поверхів, під'їздів, кількість квартир на поверсі; дата початку будівництва; дата закінчення будівництва; назва будівничої фірми. Передбачити функції для виконання наступних операцій: ініціалізація інформації, консольне введення/виведення інформації про будинок, розрахунок кількості квартир у під'їзді, у будинку з використанням *friend*-функцій.

Варіант 12. Створити клас *Train*. Зміна типу *Train* повинна містити наступні поля: номер; кількість зупинок; кількість вагонів; час відправлення; час прибуття. Передбачити функції для виконання наступних операцій: ініціалізація інформації, консольне введення та виведення інформації про потяг, розрахунок часу, що витрачається на маршрут. Створити реєстр потягів (динамічний масив).

Варіант 13. Створити клас вибірка *Sample* розмірності N . Передбачити функції для виконання наступних операцій: консольне введення/виведення значень вибірки, розрахунок середнього, дисперсії, розмаху,

середньоквадратичного відхилення. Розробити дружню функцію для розрахунку критерію Кохрена ($G=S_{\max}/G=S_{\max}/\text{SUM}(S_i) \sum S_i$).

Варіант 14. Скласти опис класу багаточленів від однієї змінної, що задаються ступенем багаточлена і масивом коефіцієнтів. Передбачити методи для обчислення значення багаточлена для заданого аргументу, операції складання, віднімання та множення багаточленів з отриманням нового об'єкта – багаточлена, виведення на екран опису многочлена.

Варіант 15. Створити клас вибірка *Sample* розмірності N. Передбачити функції для виконання наступних операцій: консольне введення/виведення значень вибірки, розрахунок середнього, дисперсії, розмаху, середньоквадратичного відхилення. Розробити дружню функцію для розрахунку критерію Фішера ($F=\sigma_1/\sigma_2$).

2 Успадкування класів

1 Створити базовий клас **Car** (машина), що характеризується торговою маркою (рядок), числом циліндрів, потужністю. Визначити методи призначення та змінення потужності. Створити похідний клас **Lorry** (вантажівка) який характеризується також вантажопідйомністю кузова. Визначити функції призначення марки та зміни вантажопідйомності.

2. Створити клас **Pair** (пара чисел); визначити методи зміни полів і порівняння пар: пара p1 більше пари p2, якщо p1.first > p2.first або p1.first = p2.first й p1.second > p2.second. Визначити похідний клас **Fraction** з полями: ціла та дробова частина числа. Визначити повний набір методів порівняння.

3. Створити клас **Liquid** (рідина), що має поля назви і щільності. Визначити методи перепризначення і зміни щільності. Створити похідний клас **Alcohol** (спирт), що має міцність. Визначити методи отримання та зміни міцності.

4. Створити клас **Pair** (пара чисел); визначити методи зміни полів і обчислення добутку чисел. Визначити похідний клас **Rectangle** (прямокутник) з полями-сторонами. Визначити методи обчислення периметра і площі прямокутника.

5. Створити клас **Man** (людина), з полями: ім'я, вік, стать і вага. Визначити методи перепризначення імені, зміни віку і зміни ваги. Створити похідний

клас **Student**, що має поле року навчання. Визначити методи перепризначення та збільшення року навчання.

6. Створити клас **Triad** (трійка чисел) й визначити методи зміни полів і обчислення суми чисел. Визначити похідний клас **Triangle** з полями-сторонами. Визначити методи обчислення кутів і площі трикутника.
7. Створити клас **Triangle** з полями-сторонами. Визначити методи зміни сторін, обчислення кутів, обчислення периметра. Створити похідний клас **Equilateral** (рівносторонній), що має поле площі. Визначити метод обчислення площі.
8. Створити клас **Triangle** з полями-сторонами. Визначити методи зміни сторін, обчислення кутів, обчислення периметра. Створити похідний клас **RightAngled** (прямокутний), що має поле площі. Визначити метод обчислення площі.
9. Створити клас **Pair** (пара чисел); визначити методи зміни полів і обчислення добутку чисел. Визначити похідний клас **RightAngled** з полями-катетами. Визначити методи обчислення гіпотенузи і площі трикутника.
10. Створити клас **Triad** (трійка чисел) та визначити метод порівняння **Triad** (див. завдання 2). Визначити похідний клас **Date** з полями: рік, місяць і день. Визначити повний набір методів порівняння дат.
11. Створити клас **Triad** (трійка чисел) й визначити метод порівняння тріад (див. завдання 2). Визначити похідний клас **Time** з полями: година, хвилина і секунда. Визначити повний набір методів порівняння моментів часу.
12. Реалізувати клас **Number** для числового типу float. Реалізувати методи додавання і ділення. Створити похідний клас **Real**, в якому реалізувати метод піднесення до довільного ступеня та метод для обчислення логарифма числа.
13. Створити клас **Triad** (трійка чисел) й визначити методи збільшення полів на 1. Визначити похідний клас **Date** з полями: рік, місяць і день. Перевизначити методи збільшення полів на 1 і визначити метод збільшення дати на n днів.
14. Реалізувати клас-оболонку **Number** для числового типу double. Реалізувати методи множення і віднімання. Створити похідний клас **Real**, в якому реалізувати метод, що обчислює корінь довільного ступеня та метод для обчислення числа π в цьому ступені.
15. Створити клас **Triad** (трійка чисел); визначити методи збільшення полів на 1. Визначити клас-спадкоємець **Time** з полями: година, хвилина, секунда. Перевизначити методи збільшення полів на 1 і визначити методи збільшення на n секунд і хвилин.

16. Створити базовий клас **Pair** (пара цілих чисел) з операціями перевірки на рівність та перемноження полів. Реалізувати операцію віднімання пар по формулі $(a, b) - (c, d) = (a - b, c - d)$. Створити похідний клас **Rational** і визначити нові операції складання $(a, b) + (c, d) = (ad + be, bd)$ і ділення $(a, b) / (c, d) = (ad, be)$. Перевизначити операцію різниці $(a, b) - (c, d) = (ad - be, bd)$.

17. Створити клас **Pair** (пара чисел) та визначити метод множення і операцію додавання пар $(a, b) + (c, d) = (a + b, c + d)$. Визначити похідний клас **Complex** з полями: дійсна і уявна частини числа. Визначити методи добутку $(a, b) \times (c, d) = (ac - bd, ad + bc)$ та різниці $(a, b) - (c, d) = (a - c, b - d)$.

18. Створити клас **Pair** (пара цілих чисел) й визначити методи зміни полів і операцію складання пар $(a, b) + (c, d) = (a + b, c + d)$. Визначити клас-спадкоємець **Long** з полями: старша частина числа і молодша частина числа. Перевизначити операцію складання і визначити методи добутку та різниці.

19. Створити базовий клас **Triad** (трійка чисел) та визначити операції складання з числом, множення на число, перевірки на рівність. Створити похідний клас **Vector3D**, що задається трійкою координат. Мають бути реалізовані: операції складання векторів, скалярний добуток векторів.

20. Створити клас **Pair** (пара цілих чисел); визначити метод множення на число і операцію додавання пар $(a, b) + (c, d) = (a + b, c + d)$. Визначити клас-спадкоємець **Money** з полями: рублі й копійки. Перевизначити операцію додавання і визначити методи різниці та ділення грошових сум.

Контрольні запитання

1. Чим відрізняються поняття класу та структури у C++ та C#? Дайте загальне визначення класу?
2. Наведіть загальну структуру класу.
3. Що таке інтерфейс класу та його реалізація?
4. У чому полягає роль конструкторів та деструкторів у класі?
5. Які способи можна застосувати для ініціалізації об'єкта класу? Як для цього застосовуються конструктори?
6. Чим відрізняються функції – члени класу від функцій, не пов'язаних з будь-яким класом?
7. Що означає об'єкт класу?
8. Уява об'єкта у пам'яті.
9. Чи можливо перевантажувати функції-члени класу?
10. В чому полягає особливість статичних елементів класу?

11. Які області видимості використовують для створення класів?
12. В чому різниця між *inline*-функціями та іншими функціями-членами класу?
13. Коли використовують дружні функції?
14. Чи можна присвоювати значення одного об'єкта іншому.
15. Які специфікатори видимості для членів класу ви знаєте?
16. Чи можливо створювати масиви об'єктів?
17. Для чого використовується покажчик `this`?
18. Чи можливо передавати об'єкти як аргументи функціям?
19. Чи можливо щоб функція повертала значення типу клас