

Лабораторна робота №5

Тема: Успадкування в C++ та C#.

Мета роботи: Набуття навичок розробки програм мовами C++ та C# з використанням базового принципу об'єктно-орієнтованого програмування – (одиначного) успадкування.

Теоретичні відомості: матеріали лекції.

Завдання на лабораторну роботу

<i>Варіант 1.</i>	
1. Описати клас, який містять вказані поля і методи.	
Клас “Вектор у R^2” – TVector2D	
поля	▪ для зберігання координат вектора;
методи	▪ конструктор без параметрів, конструктор з параметрами, конструктор копіювання; ▪ введення/виведення елементів вектора; ▪ визначення довжини вектора; ▪ нормування вектора; ▪ порівняння з іншим вектором; ▪ перевантаження операторів + (додавання векторів), – (віднімання векторів), * (знаходження скалярного добутку).
2. Створити клас-нащадок TVector3D (вектор у R^3) на основі класу TVector2D. Додати третю координату вектора та перевизначити відповідні методи.	
3. Створити програму-клієнт для тестування.	

<i>Варіант 2.</i>	
1. Описати клас, який містять вказані поля і методи.	
Клас “Прямокутник ” – TRectangle	
поля	▪ для зберігання довжин сторін;
методи	▪ конструктор без параметрів, конструктор з параметрами, конструктор копіювання; ▪ введення/виведення даних; ▪ визначення площі; ▪ визначення периметру; ▪ порівняння з іншим прямокутником; ▪ перевантаження операторів + (додавання відповідних сторін), – (віднімання довжин відповідних сторін), * (множення сторін на деяке число).
2. Створити клас-нащадок TParallelepiped (прямокутний паралелепіпед) на основі класу TRectangle. Додати поле для збереження висоти паралелепіпеда, метод знаходження об'єму паралелепіпеда та перевизначити відповідні методи.	
3. Створити програму-клієнт для тестування.	

<i>Варіант 3.</i>	
1. Описати клас, який містять вказані поля і методи.	
Клас “Квадрат ” – TSquare	
поля	<ul style="list-style-type: none"> ▪ для зберігання довжини сторін;
методи	<ul style="list-style-type: none"> ▪ конструктор без параметрів, конструктор з параметрами, конструктор копіювання; ▪ введення/виведення даних; ▪ визначення площі; ▪ визначення периметру; ▪ порівняння з іншим квадратом; ▪ перевантаження операторів + (додавання довжин сторін), – (віднімання довжин сторін), * (множення сторін на деяке число).
2. Створити клас-нащадок TCube (куб) на основі класу TSquare. Додати метод знаходження об’єму куба та перевизначити відповідні методи.	
3. Створити програму-клієнт для тестування.	

<i>Варіант 4.</i>	
1. Описати клас, який містять вказані поля і методи.	
Клас “Трикутник ” – TTriangle	
поля	<ul style="list-style-type: none"> ▪ для зберігання довжин сторін;
методи	<ul style="list-style-type: none"> ▪ конструктор без параметрів, конструктор з параметрами, конструктор копіювання; ▪ введення/виведення даних; ▪ визначення площі; ▪ визначення периметру; ▪ порівняння з іншим трикутником; ▪ перевантаження операторів + (додавання довжин сторін), – (віднімання довжин відповідних сторін), * (множення сторін на деяке число).
2. Створити клас-нащадок TTrianglePrizm (пряма призма, в основі якої трикутник) на основі класу TTriangle. Додати метод знаходження об’єму призми та перевизначити відповідні методи.	
3. Створити програму-клієнт для тестування.	

<i>Варіант 5.</i>	
1. Описати клас, який містять вказані поля і методи.	
Клас “Рівносторонній трикутник ” – TRTriangle	
поля	<ul style="list-style-type: none"> ▪ для зберігання довжини сторін;
методи	<ul style="list-style-type: none"> ▪ конструктор без параметрів, конструктор з параметрами, конструктор копіювання; ▪ введення/виведення даних; ▪ визначення площі; ▪ визначення периметру; ▪ порівняння з іншим трикутником; ▪ перевантаження операторів + (додавання довжин сторін), – (віднімання довжин відповідних сторін), * (множення сторін на деяке число).
2. Створити клас-нащадок TPiramid (правильна трикутна піраміда) на основі класу TRTriangle. Додати метод знаходження об’єму піраміди та перевизначити відповідні методи.	
3. Створити програму-клієнт для тестування.	

Варіант 6.	
1. Описати клас, який містять вказані поля і методи.	
Клас “прямокутний трикутник” – TPTriangle	
поля	▪ для зберігання довжин катетів;
методи	<ul style="list-style-type: none"> ▪ конструктор без параметрів, конструктор з параметрами, конструктор копіювання; ▪ введення/виведення даних; ▪ визначення площі; ▪ визначення периметру; ▪ порівняння з іншим трикутником; ▪ перевантаження операторів + (додавання довжин катетів), – (віднімання довжин відповідних катетів), * (множення сторін на деяке число).
2. Створити клас-нащадок TPPiramid (прямокутна трикутна піраміда, у якій бічне ребро перпендикулярне до катетів) на основі класу TPTriangle. Додати метод знаходження об’єму піраміди та перевизначити відповідні методи.	
3. Створити програму-клієнт для тестування.	

Варіант 7.	
1. Описати клас, який містять вказані поля і методи.	
Клас “коло” – TCircle	
поля	▪ для зберігання радіуса;
методи	<ul style="list-style-type: none"> ▪ конструктор без параметрів, конструктор з параметрами, конструктор копіювання; ▪ введення/виведення даних; ▪ визначення площі круга (обмеженого колом); ▪ визначення довжини кола; ▪ порівняння з іншим колом; ▪ перевантаження операторів + (додавання радіусів), – (віднімання радіусів), * (множення радіуса на число).
2. Створити клас-нащадок TCylinder (циліндр) на основі класу TCircle. Додати метод знаходження об’єму циліндра та перевизначити відповідні методи.	
3. Створити програму-клієнт для тестування.	

Варіант 8.	
1. Описати клас, який містять вказані поля і методи.	
Клас “коло” – TCircle	
поля	▪ для зберігання радіуса;
методи	<ul style="list-style-type: none"> ▪ конструктор без параметрів, конструктор з параметрами, конструктор копіювання; ▪ введення/виведення даних; ▪ визначення площі круга (обмеженого колом), площі сектора; ▪ визначення довжини кола; ▪ порівняння з іншим колом; ▪ перевантаження операторів + (додавання радіусів), – (віднімання радіусів), * (множення радіуса на число).
2. Створити клас-нащадок TCone (конус) на основі класу TCircle. Додати поле для збереження висоти конуса, метод знаходження об’єму конуса та перевизначити відповідні методи.	
3. Створити програму-клієнт для тестування.	

<i>Варіант 9.</i>	
1. Описати клас, який містять вказані поля і методи.	
Клас “коло” – TCircle	
поля	<ul style="list-style-type: none"> ▪ для зберігання радіуса;
методи	<ul style="list-style-type: none"> ▪ конструктор без параметрів, конструктор з параметрами, конструктор копіювання; ▪ введення/виведення даних; ▪ визначення площі круга (обмеженого колом), площі сектора; ▪ визначення довжини кола; ▪ порівняння з іншим колом; ▪ перевантаження операторів + (додавання радіусів), – (віднімання радіусів), * (множення радіуса на число).
2. Створити клас-нащадок TSphere (сфера) на основі класу TCircle. Додати метод знаходження площі поверхні сфери та перевизначити відповідні методи.	
3. Створити програму-клієнт для тестування.	

<i>Варіант 10.</i>	
1. Описати клас, який містять вказані поля і методи.	
Клас “круг” – TCircle	
поля	<ul style="list-style-type: none"> ▪ для зберігання радіуса; ▪ для зберігання центра кола
методи	<ul style="list-style-type: none"> ▪ конструктор без параметрів, конструктор з параметрами, конструктор копіювання; ▪ введення/виведення даних; ▪ визначення площі круга; ▪ перевірка належності точки кругу; ▪ перевантаження операторів + (додавання радіусів), – (віднімання радіусів), * (множення радіуса на число).
2. Створити клас-нащадок TBall (куля) на основі класу TCircle. Додати метод знаходження об’єму кулі та перевизначити відповідні методи.	
3. Створити програму-клієнт для тестування.	

<i>Варіант 11.</i>	
1. Описати клас, який містять вказані поля і методи.	
Клас “одновимірний масив” – TArray	
поля	<ul style="list-style-type: none"> ▪ для зберігання елементів масиву; ▪ для зберігання кількості елементів.
методи	<ul style="list-style-type: none"> ▪ конструктор без параметрів, конструктор з параметрами, конструктор копіювання; ▪ введення/виведення даних; ▪ знаходження найбільшого/найменшого елемента; ▪ сортування масиву; ▪ знаходження суми елементів; ▪ перевантаження операторів + (додавання елементів), – (віднімання елементів), * (множення масиву на число).
2. Створити клас-нащадок TOderedArray (упорядкований масив) на основі класу TArray. Додати методи додавання та вилучення елементів (перевизначивши оператори додавання та віднімання числа).	
3. Створити програму-клієнт для тестування.	

<i>Варіант 12.</i>	
1. Описати клас, який містять вказані поля і методи.	
Клас “матриця ” – TMatrix	
поля	<ul style="list-style-type: none"> ▪ для зберігання елементів матриці; ▪ для зберігання розмірності матриці.
методи	<ul style="list-style-type: none"> ▪ конструктор без параметрів, конструктор з параметрами, конструктор копіювання; ▪ введення/виведення даних; ▪ знаходження найбільшого/найменшого елемента; ▪ знаходження суми елементів.
2. Створити клас-нащадок TOpMatrix (матриця, для якої перевантажено оператори +, -, *) на основі класу TMatrix..	
3. Створити програму-клієнт для тестування.	

<i>Варіант 13.</i>	
1. Описати клас, який містять вказані поля і методи.	
Клас “квадратна матриця ” – TSMatrix	
поля	<ul style="list-style-type: none"> ▪ для зберігання елементів матриці; ▪ для зберігання розмірності матриці.
методи	<ul style="list-style-type: none"> ▪ конструктор без параметрів, конструктор з параметрами, конструктор копіювання; ▪ введення/виведення даних; ▪ знаходження найбільшого/найменшого елемента; ▪ знаходження суми елементів. ▪ перевантаження операторів + (додавання елементів), – (віднімання елементів).
2. Створити клас-нащадок TDeterminant2 (визначник квадратної матриці порядку 2) на основі класу TSMatrix. Розробити метод для знаходження визначника.	
3. Створити програму-клієнт для тестування.	

<i>Варіант 14.</i>	
1. Описати клас, який містять вказані поля і методи.	
Клас “квадратна матриця ” – TSMatrix	
поля	<ul style="list-style-type: none"> ▪ для зберігання елементів матриці; ▪ для зберігання розмірності матриці.
методи	<ul style="list-style-type: none"> ▪ конструктор без параметрів, конструктор з параметрами, конструктор копіювання; ▪ введення/виведення даних; ▪ знаходження найбільшого/найменшого елемента; ▪ знаходження суми елементів. ▪ перевантаження операторів + (додавання матриць), – (віднімання матриць).
2. Створити клас-нащадок TMSMatrix (клас доповнюється новими методами: транспонування, перевантаження оператора * (множення матриці на матрицю) та перевантаження оператора * (множення матриці на число)) на основі класу TSMatrix. Розробити метод для знаходження визначника.	
3. Створити програму-клієнт для тестування.	

<i>Варіант 15.</i>	
1. Описати клас, який містять вказані поля і методи.	
Клас “звичайний дріб” – TFraction	
поля	<ul style="list-style-type: none"> ▪ для зберігання чисельника і знаменника;
методи	<ul style="list-style-type: none"> ▪ конструктор без параметрів, конструктор з параметрами, конструктор копіювання; ▪ введення/виведення даних; ▪ скорочення дробів (якщо чисельник і знаменник містять спільні множники); ▪ перевантаження операторів +, −, *, / .
2. Створити клас-нащадок TMixFraction (мішані дробі – ціла частина, чисельник і знаменник) на основі класу TFraction. Додати поле для збереження цілої частини та перевизначити відповідні методи.	
3. Створити програму-клієнт для тестування.	

<i>Варіант 16.</i>	
1. Описати клас, який містять вказані поля і методи.	
Клас “комплексне число” – TComplex	
поля	<ul style="list-style-type: none"> ▪ для зберігання дійсної і уявної частин;
методи	<ul style="list-style-type: none"> ▪ конструктор без параметрів, конструктор з параметрами, конструктор копіювання; ▪ введення/виведення даних; ▪ перевантаження операторів +, −, *, / .
2. Створити клас-нащадок TComplex (комплексне число на площині) на основі класу TComplex. Додати методи визначення квадранта, у який попадає комплексне число, метод визначення відстані до початку координат.	
3. Створити програму-клієнт для тестування.	

<i>Варіант 17.</i>	
1. Описати клас, який містять вказані поля і методи.	
Клас “пряма на площині” – TLine2D	
поля	<ul style="list-style-type: none"> ▪ для зберігання коефіцієнтів канонічного рівняння прямої;
методи	<ul style="list-style-type: none"> ▪ конструктор без параметрів, конструктор з параметрами, конструктор копіювання; ▪ введення/виведення даних; ▪ знаходження точки перетину з іншою прямою; ▪ визначення належності точки прямій; ▪ перевантаження операторів + (додавання коефіцієнтів прямих), − (віднімання коефіцієнтів прямих).
2. Створити клас-нащадок TLine3D (пряма у просторі) на основі класу TLine2D. Додати ще одне поле для збереження коефіцієнта при невідомій Z та перевизначити відповідні методи.	
3. Створити програму-клієнт для тестування.	

Варіант 18. Описати клас, який містять вказані поля і методи.

Клас “відрізок на площині” – TInterval2D	
поля	<ul style="list-style-type: none"> ■ для зберігання координат початку і кінця відрізка;
методи	<ul style="list-style-type: none"> ■ конструктор без параметрів, конструктор з параметрами, конструктор копіювання; ■ введення/виведення даних; ■ знаходження точки перетину з іншим відрізком; ■ визначення довжини відрізка; ■ визначення середини відрізка; ■ перевантаження операторів + (утворюється відрізок початок якого є початком першого, а кінець – кінцем другого), * (збільшення довжини відрізка у вказану кількість разів зберігаючи початок відрізка незмінним).

2. Створити клас-нащадок TInterval3D (відрізок у просторі) на основі класу TInterval2D. Додати поля для збереження третьої координати точок початку і кінця та перевизначити відповідні методи.

3. Створити програму-клієнт для тестування.

Варіант 19. Описати клас, який містять вказані поля і методи.

Клас “точка на площині” – TPoint2D	
поля	<ul style="list-style-type: none"> ■ для зберігання довжин сторін;
методи	<ul style="list-style-type: none"> ■ конструктор без параметрів, конструктор з параметрами, конструктор копіювання; ■ введення/виведення даних; ■ визначення відстані до іншої точки; ■ визначення квадранту, до якого належить точка; ■ порівняння з іншою точкою; ■ перевантаження операторів + (додавання відповідних координат), – (віднімання відповідних координат).

2. На основі класу TPoint2D створити клас TRectangle (прямокутник задається координатами вершин). Клас повинен містити чотири поля типу TPoint2D і методи для знаходження площі, периметру прямокутника та методу визначення того, чи лежить вказана точка всередині прямокутника.

3. Створити програму-клієнт для тестування.

Варіант 20. Описати клас, який містять вказані поля і методи.

Клас “точка на площині” – TPoint2D	
поля	<ul style="list-style-type: none"> ■ для зберігання координат;
методи	<ul style="list-style-type: none"> ■ конструктор без параметрів, конструктор з параметрами, конструктор копіювання; ■ введення/виведення даних; ■ визначення відстані до іншої точки; ■ визначення квадранту, до якого належить точка; ■ порівняння з іншою точкою; ■ перевантаження операторів + (додавання відповідних координат), – (віднімання відповідних координат).

2. На основі класу TPoint2D створити клас TCTriangle (трикутник задається координатами вершин). Клас повинен містити три поля типу TPoint2D і методи для знаходження площі, периметру трикутника та методу визначення того, чи лежить вказана точка всередині трикутника.

3. Створити програму-клієнт для тестування.

<i>Варіант 21.</i> Описати клас, який містять вказані поля і методи.	
Клас “арифметична прогресія” – TArProgression	
поля	▪ для зберігання першого члена арифметичної прогресії та різниці;
методи	▪ конструктор без параметрів, конструктор з параметрами, конструктор копіювання; ▪ введення/виведення даних; ▪ знаходження n -го члена; ▪ знаходження суми n перших членів прогресії; ▪ перевантаження операторів $+$ (додавання перших членів та різниць), $-$ (віднімання перших членів та різниць).
2. На основі класу TArProgression створити клас-нащадок TArProgressionM. Цей клас повинен містити метод, який дозволяє визначити, чи утворює послідовність цілих чисел арифметичну прогресію та метод, який дозволяє встановити, чи є вказане число членом даної арифметичної прогресії.	
3. Створити програму-клієнт для тестування.	

<i>Варіант 22.</i> Описати клас, який містять вказані поля і методи.	
Клас “геометрична прогресія” – TGeomProgression	
поля	▪ для зберігання першого члена геометричної прогресії та знаменника;
методи	▪ конструктор без параметрів, конструктор з параметрами, конструктор копіювання; ▪ введення/виведення даних; ▪ знаходження n -го члена; ▪ знаходження суми n перших членів прогресії; ▪ перевантаження операторів $+$ (додавання перших членів та знаменників), $-$ (віднімання перших членів та знаменників).
2. На основі класу TGeomProgression створити клас-нащадок TGeomProgressionM. Цей клас повинен містити метод, який дозволяє визначити, чи утворює послідовність додатних цілих чисел геометричну прогресію та метод, який дозволяє встановити, чи є вказане число членом даної геометричної прогресії.	
3. Створити програму-клієнт для тестування.	

Контрольні запитання

1. У чому полягає суть успадкування?
2. Що таке клас-предок?
3. Що таке клас-нащадок?
4. Для чого використовують класи-предки?
5. Скільки класів-предків може мати клас?
6. Як описати клас-нащадок?
7. Що таке перевизначення методів?
8. Як перевизначити метод класу-предка?
9. Як здійснити доступ до перевизначених методів класу-предка?
10. Чи має об'єкт-нащадок доступ до перевизначених методів класу-предка?
11. Які члени класу-предка є доступними для класу-нащадка?
12. Чи має можливість клас-нащадок змінювати область видимості членів класу-предка?
13. Як описати члени класу-предка, щоб вони були доступними для класу-нащадка і недоступними у програмі-клієнті?
14. Як описати члени класу-предка, щоб вони не були доступними як для класу-нащадка, так і для програми-клієнта?
15. Чи може змінна типу класу-предка приймати значення об'єкта-нащадка?
16. До яких полів і методів об'єкта-нащадка може здійснити доступ об'єкт-предок?