

Лабораторна робота №6

Об'єктно-орієнтоване програмування у С#

Мета лабораторної роботи:

1. Вивчити поняття класу та структури в С#.
2. Вивчити успадкування та поліморфізм у С#.
3. Вивчити прийоми програмування з використанням класів, наслідування та поліморфізму мовою С#.

Зміст роботи.

1. Теоретичні відомості

1.1 Основи ООП в С

Класи - основні конструкції мови (свого роду описи), задають одночасно тип (посилальний) і схему-зразок (може бути неповну) для побудови об'єктів, що населяють цей тип. Саме об'єкти, а чи не класи, виконують всю роботу (обчислення). Класи служать для проектування об'єктів.

Склад класів.

Члени класу (members): поля, властивості, методи, ... Той самий склад мають об'єкти цього класу.

Статичні члени (static) пов'язані з класом повністю, інші (non-static) прив'язані до об'єктів. Цим визначається форма виклику: cl.memб або obj.memб.

Крім того, є модифікатори доступу до класу та/або члена класу: public (без

обмежень),

private (тільки з цього класу),

protected (тільки з даного класу та його спадкоємців) та

internal (для класу, лише з даного складання).

Структури C# аналогічні класам, але задають типи-значення (а чи не посилання).

Типові змінні (generics).

Вони допускаються у визначенні класу, що призводить до завдання типів, які залежать від типових параметрів.

Наприклад, тип масиву є параметром в описі типу масиву. Значення цього

параметра задається під час створення конкретних масивів (int[], char[], і т.д.).

1.2 Делегати та функціональні типи у C#

Делегати служать для того, щоб передавати методи як параметри іншим методам.

Об'єкт-делегат - це свого роду упаковка, до якої можна додати одну або кілька функцій (однакової сигнатури, зазначеної в описі типу делегата).

Метод, який отримує упаковку, може викликати, тобто. запустити всі ці функції одну за одну, передавши їм усім одні й самі аргументи.

Тип делегата - деякий заміник функціонального типу $A \rightarrow B$.

Прогрес від введення такого типу в тому, що він може бути використаний як тип формального параметра у методу;

в коді методу немає необхідності писати справжні імена функцій, що викликаються.

Без такої можливості доводиться передавати як параметр весь об'єкт цілком, але для того, щоб скористатися його методами, треба знати імена цих методів.

Стандартний функціональний тип $A \rightarrow B$ виходить, якщо домовитися не додавати делегат більше однієї функції.

Додаткова можливість додавати до делегата кілька функцій добре працює у механізмі обробки подій.

2. Завдання на лабораторну роботу

Розробити клас, набір методів (конструктор та мінімум два методи) для програмної моделі заданого об'єкта. Опис об'єкта та його основних властивостей наводиться нижче. Навести тестову програму, яка використовує об'єкти розробленого класу.

Номер варіанта відповідає Вашому номеру у списку журналу академічної групи.

1. Об'єкт "комплексні числа". Операції визначаються за загальноприйнятими формулами. Передбачити можливість операції присвоєння, додавання, множення та переведення в текстовий рядок поточних значень. Конструктор повинен дозволити створювати об'єкти без і з початковою ініціалізацією.
2. Об'єкт "комплексні числа". Операції визначаються за загальноприйнятими формулами. Передбачити можливість операції присвоєння, віднімання, множення та переведення в текстовий рядок поточних значень. Конструктор повинен дозволити створювати об'єкти без і з початковою ініціалізацією.
3. Об'єкт "комплексні числа". Операції визначаються за загальноприйнятими формулами. Передбачити можливість операції присвоєння, додавання, поділу та переведення в текстовий рядок поточних значень. Конструктор повинен дозволити створювати об'єкти без і з початковою ініціалізацією.
4. Об'єкт "вектор на площині" заданий у системі декартових координат. Початок вектора розташований на початку координат. Операції визначаються відповідно до загальноприйнятих формул лінійної (векторної) алгебри. Передбачити можливість операції присвоєння, додавання, скалярного множення та роздруківки координат поточних значень.

Конструктор повинен дозволити створювати об'єкти без і з початковою ініціалізацією.

5. Об'єкт "вектор на площині" заданий у системі декартових координат. Початок вектора розташований на початку координат. Операції визначаються відповідно до загальноприйнятих формул лінійної (векторної) алгебри. Передбачити можливість операції присвоєння, віднімання, скалярного множення та роздруківки координат поточних значень. Конструктор повинен дозволити створювати об'єкти без і з початковою ініціалізацією.
6. Об'єкт "вектор на площині" заданий у системі декартових координат. Початок вектора розташований на початку координат. Операції визначаються відповідно до загальноприйнятих формул лінійної (векторної) алгебри. Передбачити можливість операції присвоєння, порівняння модулів, скалярного множення та роздруківки координат поточних значень. Конструктор повинен дозволити створювати об'єкти без і з початковою ініціалізацією.
7. Об'єкт "вектор на площині" заданий у системі декартових координат. Початок вектора розташований на початку координат. Операції визначаються відповідно до загальноприйнятих формул лінійної (векторної) алгебри. Передбачити можливість операції присвоєння, знаходження кута між векторами, скалярного множення та роздруківки координат поточних значень. Конструктор повинен дозволити створювати об'єкти без і з початковою ініціалізацією.
8. Об'єкт «рівностегновий трикутник заданий довжинами сторін». Передбачити можливість операції привласнення, визначення площі та периметра, а також логічний метод, що визначає чи існує такий трикутник. Конструктор повинен

дозволити створювати об'єкти без і з початковою ініціалізацією.

9. Об'єкт «рівносторонній трикутник, заданий довжинами сторін». Передбачити можливість операції привласнення, визначення площі та периметра, а також логічний метод, що визначає чи існує такий трикутник. Конструктор повинен дозволити створювати об'єкти без і з початковою ініціалізацією.
10. Об'єкт «прямокутний трикутник, заданий довжинами сторін». Передбачити можливість операції привласнення, визначення площі та периметра, а також логічний метод, що визначає чи існує такий трикутник. Конструктор повинен дозволити створювати об'єкти без і з початковою ініціалізацією.
11. Об'єкт «рівностегновий трикутник заданий довжиною рівнобедреною стороною та кутом між ними». Передбачити можливість операції привласнення, визначення площі та периметра, а також логічний метод, що відповідає на питання – гостро чи тупо вугільним є заданий трикутник. Конструктор повинен дозволити створювати об'єкти без і з початковою ініціалізацією.
12. Об'єкт «трикутник заданий довжиною двома сторонами та кутом між ними». Передбачити можливість операції привласнення, визначення площі та периметра, а також логічний метод, що відповідає на питання – гостро чи тупо вугільним є заданий трикутник. Конструктор повинен дозволити створювати об'єкти без і з початковою ініціалізацією.
13. Об'єкт «прямокутник, заданий довжинами двох сторін». Передбачити можливість операції привласнення, визначення площі та периметра, а також логічний метод, що відповідає на питання – чи є прямокутник квадратом.

Конструктор повинен дозволити створювати об'єкти без і з початковою ініціалізацією.

14. Об'єкт «безліч цілих чисел заданої потужності». Передбачити можливість операції присвоєння, об'єднання двох множин, виведення на друк елементів множини, а також метод, що відповідає на запитання, чи належить зазначене значення множині. Конструктор повинен дозволити створювати об'єкти без і з початковою ініціалізацією. Потужність безлічі задається під час створення об'єкта.
15. Об'єкт «безліч дійсних чисел заданої потужності». Передбачити можливість операції присвоєння, об'єднання двох множин, виведення на друк елементів множини, а також метод, що відповідає на запитання, чи належить зазначене значення множині. Конструктор повинен дозволити створювати об'єкти без і з початковою ініціалізацією. Потужність безлічі задається під час створення об'єкта.
16. Об'єкт «безліч символів заданої потужності». Передбачити можливість операції присвоєння, об'єднання двох множин, виведення на друк елементів множини, а також метод, що відповідає на запитання, чи належить зазначене значення множині. Конструктор повинен дозволити створювати об'єкти без і з початковою ініціалізацією. Потужність безлічі задається під час створення об'єкта.
17. Об'єкт «безліч цілих чисел подвоєної довжини заданої потужності». Передбачити можливість операції присвоєння, об'єднання двох множин, виведення на друк елементів множини, а також метод, що відповідає на запитання, чи належить зазначене значення множині. Конструктор повинен дозволити створювати об'єкти без і з початковою ініціалізацією. Потужність безлічі задається під час створення об'єкта.

18. Об'єкт «безліч дійсних чисел подвоєної точності заданої потужності». Передбачити можливість операції присвоєння, об'єднання двох множин, виведення на друк елементів множини, а також метод, що відповідає на запитання, чи належить зазначене значення множині. Конструктор повинен дозволити створювати об'єкти без і з початковою ініціалізацією. Потужність безлічі задається під час створення об'єкта.
19. Об'єкт «безліч байт заданої потужності». Передбачити можливість операції присвоєння, об'єднання двох множин, виведення на друк елементів множини, а також метод, що відповідає на запитання, чи належить зазначене значення множині. Конструктор повинен дозволити створювати об'єкти без і з початковою ініціалізацією. Потужність безлічі задається під час створення об'єкта.
20. Об'єкт «безліч цілих чисел не заданої (змінної) потужності». Передбачити можливість операції додати елемент до множини, визначення кількості елементів у множині, виведення на друк всіх елементів множини, а також метод видаляє зазначений елемент з множини, якщо цей елемент належить множині. Конструктор повинен дозволити створювати об'єкти без і з початковою ініціалізацією.
21. Об'єкт «безліч дійсних чисел не заданої (змінної) потужності». Передбачити можливість операції додати елемент до множини, визначення кількості елементів у множині, виведення на друк всіх елементів множини, а також метод видаляє зазначений елемент з множини, якщо цей елемент належить множині. Конструктор повинен дозволити створювати об'єкти без і з початковою ініціалізацією.
22. Об'єкт «безліч символів не заданої (змінної) потужності». Передбачити можливість операції додати елемент до множини, визначення кількості

елементів у множині, виведення на друк всіх елементів множини, а також метод видаляє зазначений елемент з множини, якщо цей елемент належить множині. Конструктор повинен дозволити створювати об'єкти без і з початковою ініціалізацією.

23. Об'єкт «безліч цілих чисел подвоєної довжини не заданої (змінної) потужності». Передбачити можливість операції додати елемент до множини, визначення кількості елементів у множині, виведення на друк всіх елементів множини, а також метод видаляє зазначений елемент з множини, якщо цей елемент належить множині. Конструктор повинен дозволити створювати об'єкти без і з початковою ініціалізацією.
24. Об'єкт «прямокутник, заданий довжинами двох сторін». Передбачити можливість операції привласнення, визначення площі та периметра, а також логічний метод, що відповідає на запитання – чи міститься вказаний параметрами методу прямокутник усередині прямокутника. Конструктор повинен дозволити створювати об'єкти без і з початковою ініціалізацією

3. Звіт

Повинен містити: постановку завдання, програмний код розв'язання, результат роботи написаної програми