Лабораторна робота №4. Використання принципів та відносин об’єктно-орієнтованого програмування у мові C# .

20 варіант

**Завдання**

1.Розробити клас згідно до свого варіанта. Включити до класу методи set (...), get (...), show (...) та ін., використовуючи принцип інкапсуляції. Окрім гетерів, сетерів конструкторів та інших стандартних методів розробити не менше двох додаткових методів, які реалізують власний функціонал об’єкту відповідної предметної галузі. Написати програму, яка створює список, масив чи колекцію, об’єктів на основі відповідного класу. Під час створення використати випадковий підхід для генерації числової та текстової інформації об’єктів списку. Передбачити можливість додання, редагування та видалення об’єктів зі списку за певними критеріями. В програмі реалізувати функціонал за своїм варіантом.

**Worker:** Прізвище та ініціали, Посада, Рік надходження на роботу,

Зарплата. Створити масив об'єктів. Вивести:

• список працівників, стаж роботи яких цьому підприємстві

перевищує задане число років;

• список працівників, зарплата яких більша за задану;

• список працівників, які обіймають задану посаду.

В цьому коді я створив клас Worker, який має приватні поля для інформації про працівника, а також методи Get та Set для доступу до цих полів. Також є метод ShowInfo, який виводить інформацію про працівника. У класі Program я створюю список працівників, генеруючи деякі випадкові значення, а потім виводжу інформацію про них та виконую запити, які були описані в завданні. Цей код створює список працівників з випадковими іменами, посадами, роками початку роботи та зарплатами. Потім він виводить інформацію про всіх працівників та робить запити, які вказані в завданні. Два масиви firstNames та lastNames, які містять різні імена та прізвища. Додав також метод GenerateRandomName, який обирає випадкове ім'я та прізвище. Потім використовую цей метод при створенні об'єктів Worker.

2.Для класу з минулого завдання, ретельно занурившись у предметну

область власного варіанта, створити абстрактний клас чи інтерфейс,

який має стати предком, зв’язати ці класи відносинами спадкування.

Наприклад для класу Book (Книга) можна створити клас Reader (Засіб

читання). Окрім вашого класу додати до абстрактного класу чи інтерфейсу,

створеному у попередньому пункті, ще два класи-спадкоємця першого

рівня. Наприклад до класу Book (Книга) можна додати класи Newspaper

(Газета) Magazine (Журнал) з відповідними полями та методами. Для одного з доданих у попередньому завданні класів створити не менше двох спадкоємців другого рівня спадкування. Наприклад до класу Magazine (Журнал) можна додати класи PMagazine (Паперовий журнал), EMagazine (Електронний журнал), IMagazine (Інтернет журнал). У кожного нащадка має бути не менш ніж на два поля та метода більше ніж у предка. Створити масив чи колекцію об’єктів усіх рівнів спадкування. Виконати відповідні функції над поліморфним списком із завдання власного

варіанта.

Я використовую клас Worker для представлення працівників і класи Work, RegularWork, OvertimeWork, HighPriorityWork, LowPriorityWork для представлення різних типів робіт. У функції Main створюється список працівників та масив робіт, виводиться інформацію про них та додаткову інформацію для об'єктів RegularWork. Створив абстрактний клас Work, Додав два класи-спадкоємця для класу Work: RegularWork (Регулярна робота) та OvertimeWork (Овертайм). Створив два класи-спадкоємця для класу RegularWork: HighPriorityWork (Високопріоритетна робота) та LowPriorityWork (Низькопріоритетна робота). Створив масив об'єктів усіх рівнів спадкування та виконаємо відповідні функції. Цей код створює об'єкти різних типів робіт (регулярна робота, овертайм тощо) та викликає відповідні методи для виведення інформації про кожну роботу. Далі, використовуючи перевірку типу (is), виводить додаткову інформацію для робіт, які є типом RegularWork.

3.Розробити клас та перевантажити оператори згідно до свого варіанта. В головній функції програми протестувати роботу створених класів на прикладі використання окремих об’єктів та масивів чи колекцій цих об’єктів. Під час створення об’єктів застосувати випадковий підхід чи зчитування інформації з файлів. Цей код реалізує клас "Многочлен" (Polynomial) ступеня N. В цьому класі є такі можливості: Розробити клас "Многочлен" - Polynom ступеня N. Написати кілька конструкторів, зокрема конструктор копіювання. Перевантажити оператори додавання, віднімання, множення, інкременту, декременту, індексування, присвоєння. Створити масив об'єктів класу. Передати його в функцію, що обчислює суму поліномів масиву та повертає поліном-результат.

1. Конструктори, включаючи конструктор копіювання.

2. Перевантажені оператори: додавання, віднімання, множення, інкременту, декременту, індексування та присвоєння.

3. Здатність створювати масив об'єктів класу та передавати його у функцію для обчислення суми поліномів та повертання поліном-результату.

Цей код дозволяє працювати з поліномами та виконувати операції з ними. Наприклад, можна додавати, віднімати, множити поліноми, а також використовувати індексацію для доступу до коефіцієнтів полінома за їхніми ступенями.

4.Для класу першого згідно до свого варіанту створити головний чи

підпорядкований клас із яким організувати відносини агрегації.

Продемонструвати роботу відносин, створивши декілька об’єктів відповідних класів та позбавляючи чи додаючи властивості до цих

об’єктів. Для класу першого завдання згідно до свого варіанту створити

головний чи підпорядкований клас із яким організувати відносини

композиції. Продемонструвати роботу відносин, створивши декілька

об’єктів відповідних класів та позбавляючи чи додаючи властивості

до цих об’єктів.

Спершу створив додаткові класи, що відповідають за агрегацію та композицію. Для агрегації, додав клас Project, який буде містити список робіт. Тепер для композиції, додав клас Employee, який буде включати в себе список робіт. Тепер, в головному методі Main, створив об'єкти для демонстрації відносин. В результаті, я створив об'єкти Project та Employee, які містять в собі різні види робіт. Обидва ці класи додають та відображають роботи.