**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

**імені ТАРАСА ШЕВЧЕНКА**

**Факультет інформаційних технологій**

**Кафедра мережевих та інтернет технологій**

**Звіт до лабораторної роботи №1**

З основ інформаційної безпеки на тему:

“ Генерація послідовності випадкових чисел”

Лабораторна робота студентки 2 курсу

першого (бакалаврського) рівня

денної форми навчання

Кубєдінової Поліни Анатоліївни

Викладач:

Махович О. І.

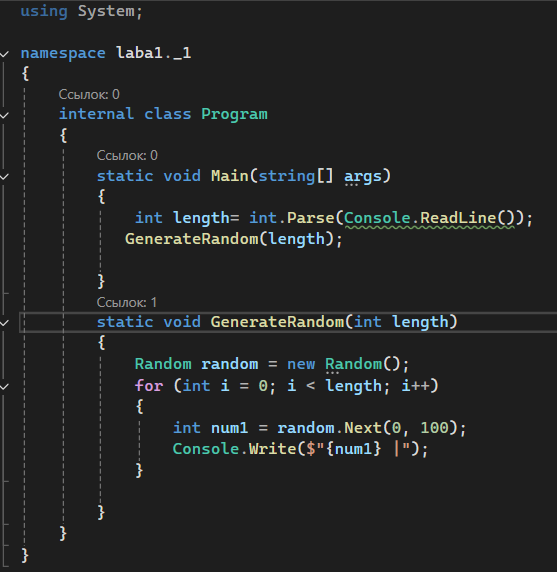
**Київ — 2024**

Мета: Ознайомитися з методами генерації псевдовипадкових та криптографічно стійких випадкових чисел на платформі .NET, а також навчитися використовувати класи System.Random і RandomNumberGenerator для створення послідовностей чисел. Порівняти результати генерації для різних сценаріїв та оцінити їхню застосовність.

Хід роботи

1. Завдання : Написати програму, яка забезпечує генерацію та виведення на екран послідовності псевдовипадкових чисел. Порівняти елементи послідовності для однакових та різних початкових значень. Зробити висновки.

Вигляд коду:



* static void GenerateRandom(int length) — метод для генерації випадкових чисел. Параметр length визначає кількість чисел, які потрібно згенерувати.
* Random random = new Random(); — створює новий об'єкт класу Random, який використовується для генерації псевдовипадкових чисел.
* for (int i = 0; i < length; i++) — цикл for виконується стільки разів, скільки вказав користувач (значення length).
* random.Next(0, 100) — метод Next об'єкта Random генерує псевдовипадкове ціле число в діапазоні від 0 до 99 (верхня межа не включається).

Результат виконання при вхідних даних 1234:



Обидва рази за однакових вхідних даних отримуємо однакові псевдовипадкові числа.

Результат виконання коду при вхідних даних 4321:



Результат виконання коду при вхідних даних 1111:



Результат виконання коду при вхідних даних 4444:

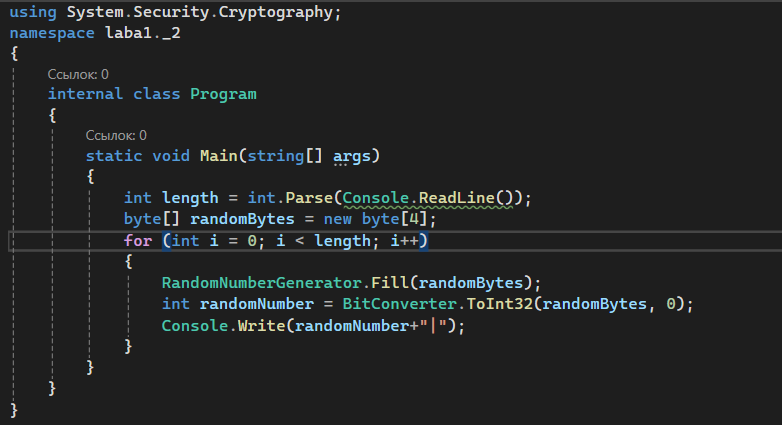


Отже, різні початкові значення генерують різні псевдовипадкові послідовності. Така детермінованість робить генератор непридатним для задач, що вимагають справжньої випадковості та непередбачуваності.

1. Завдання:

Написати програму, яка забезпечує генерацію та виведення на екран криптографічно стійкої послідовності випадкових чисел. Порівняти елементи послідовності для декількох послідовних спроб. Зробити висновки.

Вигляд коду:



* Оголошується масив randomBytes, який містить 4 байти. 4 байти (32 біти) достатньо для зберігання одного випадкового цілого числа (int), оскільки тип int у C# займає 4 байти.
* RandomNumberGenerator.Fill(randomBytes); заповнює масив randomBytes криптографічно стійкими випадковими байтами. Кожного разу цей масив заповнюється новими значеннями.
* BitConverter.ToInt32(randomBytes, 0); перетворює 4 байти з масиву в ціле число (int). Метод BitConverter.ToInt32 приймає масив байтів і початкову позицію (в даному випадку 0) для перетворення.

Спроба виконання коду 1:



Спроба виконання коду 2:



Спроба виконання коду 3:



Клас RandomNumberGenerator генерує випадкові значення, які неможливо передбачити або повторити, що є важливим для криптографічних цілей, наприклад, для генерації ключів шифрування.

Висновок: В ході виконання лабораторної роботи ми вивчили два типи генерації випадкових чисел у C#:

Псевдовипадкові числа з класом Random, які є детермінованими і повторюють ту саму послідовність для одного й того ж початкового значення.

Криптографічно стійкі випадкові числа з класом RandomNumberGenerator, які забезпечують непередбачуваність і є необхідними для безпечного шифрування.