Міністерство освіти і науки України
Український державний університет науки і технологій

Факультет «Комп'ютерні технології і системи»
(назва факультету)

Кафедра «Електронні обчислювальні машини»
(повна назва кафедри)

Пояснювальна записка

до кваліфікаційної роботи
бакалавра
(ступінь вищої освіти)

на тему: Розробка комплексу генерації випадкових та псевдовипадкових чисел

за освітньою програмою Кібербезпека

зі спеціальності: 125 Кібербезпека
(шифр і назва спеціальності)

Виконав: студент групи: КБ1811

(підпис студента) / Максим ПЛЯШКО /
(Ім'я ПРІЗВИЩЕ)

Керівник: (підпис) / доцент, Денис ОСТАПЕЦЬ /
(посада, Ім'я ПРІЗВИЩЕ)

Нормоконтролер: (підпис) / ст. викладач, Володимир ДЗЮБА/
(посада, Ім'я ПРІЗВИЩЕ)

Консультанти:

(назва розділу) (підпис) / /
(посада, Ім'я ПРІЗВИЩЕ)

(назва розділу) (підпис) / /
(посада, Ім'я ПРІЗВИЩЕ)

(назва розділу) (підпис) / /
(посада, Ім'я ПРІЗВИЩЕ)

Засвідчую, що у цій роботі немає запозичень з праць інших авторів без відповідних посилань.

Студент (підпис)

Дніпро –2022 рік

Ministry of Education and Science of Ukraine
Ukrainian State University of Science and Technologies

Faculty «Computer technologies and systems»
(faculty)

Department «Electronic computers»
(department)

Explanatory Note
to Bachelor's Thesis

(higher education degree)

on the topic: Development of a complex for generating random and pseudo-random numbers

according to educational curriculum Cybersecurity

in the Speciality: «125 Cybersecurity»
(speciality and its code)

Done by the student of the group: KB1811 / Maksim Pliashko /
(name, surname)

Scientific Supervisor: / Associate Professor Denis Ostapets /
(position, name, surname)

Normative controller : / Senior lecturer Volodymyr Dziuba /
(position, name, surname)

Supervisors

(Chapter title heading)	/ / (position, name, surname)
(Chapter title heading)	/ / (position, name, surname)
(Chapter title heading)	/ / (position, name, surname)
(Chapter title heading)	/ / (position, name, surname)

Dnipro – 2022

Міністерство освіти і науки України
Український державний університет науки і технологій

Факультет: Комп'ютерні технології і системи
Кафедра: Електронні обчислювальні машини
Рівень вищої освіти: Перший (бакалаврський)
Освітня програма: Кібербезпека
Спеціальність: 125 Кібербезпека
(шифр та назва)

ЗАТВЕРДЖУЮ
Завідувач кафедри _____
_____ _____
(підпис) (Ім'я ПРІЗВИЩЕ)
Дата _____

ЗАВДАННЯ

на кваліфікаційну роботу бакалавра
(ступінь вищої освіти)

студенту Пляшку Максиму Сергійовичу
(Прізвище, Ім'я По батькові)

1. Тема роботи: Розробка комплексу генерації випадкових та псевдовипадкових чисел.

Керівник роботи: Остапець Денис Олександрович, к.т.н., доцент
(Прізвище, Ім'я, По батькові, науковий ступінь, вчене звання)

затверджені наказом від "07" 12 2021 р. № 67 ст

2. Строк подання студентом роботи: 13.06.2022 р.
3. Вихідні дані до роботи: Методи генерації псевдовипадкових чисел, схеми апаратних генераторів випадкових чисел
4. Зміст пояснювальної записки (перелік питань, які потрібно опрацювати):
4.1 Аналітична частина:
Аналіз методів генерації псевдовипадкових чисел та джерел шуму генераторів випадкових чисел
4.2 Основна частина:
- Огляд генераторів числових послідовностей
- Структура програмного комплексу
- Розробка програмного забезпечення
- Перевірка працездатності
- Інструкція з використання
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень):
- Характеристика генераторів випадкових, псевдовипадкових чисел та джерел

шуму
- Склад та функції комплексу
- Основні алгоритми програм
- Приклади роботи комплексу, основні екранні форми

6. Консультанти розділів роботи:

Розділ	Прізвище, ініціали та посада консультанта	Завдання видав: (підпис консультанта, дата)	Завдання прийняв (підпис, дата)

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Прим
1	Огляд генераторів числових послідовностей	25.04.22	20%
2	Структура програмного комплексу	12.05.22	20%
3	Розробка програмного забезпечення	24.05.22	40%
4	Перевірка працездатності	01.06.22	10%
5	Інструкція з використання	08.06.22	5%
6	Реферат, вступ, висновки	13.06.22	5%
7	Подання кваліфікаційної роботи до кафедри	13.06.22	
8	Захист кваліфікаційної роботи на засіданні Екзаменаційної комісії		

Студент _____ (підпис) Максим ПЛЯШК (Ім'я ПРІЗВИЩ)

Керівник роботи _____ (підпис) Денис ОСТАПЕ (Ім'я ПРІЗВИЩ)

- Склад та функції комплексу
- Основні алгоритми програм
- Приклади роботи комплексу, основні екранні форми
- Характеристика генераторів випадкових, псевдовипадкових чисел та джерел

6. Консультанти розділів роботи:

Розділ	Прізвище, ініціали та посада консультанта	Завдання видав: (підпис консультанта, дата)	Завдання прийняв: (підпис студента, дата)

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Огляд генераторів числових послідовностей	25.04.22	20%
2	Структура програмного комплексу	12.05.22	20%
3	Розробка програмного забезпечення	24.05.22	40%
4	Перевірка працездатності	01.06.22	10%
5	Інструкція з використання	08.06.22	5%
6	Реферат, вступ, висновки	13.06.22	5%
7	Подання кваліфікаційної роботи до кафедри	13.06.22	
8	Захист кваліфікаційної роботи на засіданні Екзаменаційної комісії		

Студент _____ Максим ПЛЯШКО
(підпис) (Ім'я ПРІЗВИЩЕ)

Керівник роботи _____ Денис ОСТАПЕЦЬ
(підпис) (Ім'я ПРІЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи бакалавра:

50 с., 23 рис., 4 табл., 5 додатків, 15 джерел, 6 формул.

Об’єкт розробки – комплекс генерації випадкових та псевдовипадкових чисел, якій використовує мікрофон у якості джерела шуму.

Мета роботи – розробка програмного клієнт-серверного комплексу генерації випадкових та псевдовипадкових чисел.

Приведено опис та обґрунтовано вибір методів генерації випадкових та псевдовипадкових чисел. Описано архітектуру комплексу, його режими роботи та організацію взаємодії між клієнтами та сервером. Розроблені блок-схеми узагальнених алгоритмів роботи комплексу. Написано та налагоджено програмне забезпечення, перевірено його працездатність. Написано інструкцію з використання комплексу.

Результати роботи можуть бути використані на практиці для отримання випадкових та псевдовипадкових чисел та у навчальному процесі студентів відповідних спеціальностей при проведенні лабораторних і практичних робіт.

Ключові слова: ГЕНЕРАЦІЯ ЧИСЕЛ, ВИПАДКОВІ ЧИСЛА, ПСЕВДОВИПАДКОВІ ЧИСЛА, C#, TCP, СОКЕТ, МІКРОФОН, ШУМ, ЛІНІЙНИЙ КОНГРУЕНТНИЙ МЕТОД.

ЗМІСТ

ВСТУП

Генерація випадкових чисел - важливе питання в сучасному інформаційному світі. Наприклад, випадкові числа використовуються при отриманні ключів для шифрування інформації, моделюванні, іграх і т.п. Перші генератори мали ряд проблем, через які їх перестали використовувати, а саме - швидка вироджуваність, ненадійність та передбачуваність.

Розвиток технологій зумовив появу нових генераторів випадкових чисел для рішення вище описаних проблем. Розробка генераторів чисел ґрунтується на можливості створення генератора, в якому всі числа мають однакову ймовірність появи.

Дана робота присвячена розробці засобів генерації реальних випадкових та псевдовипадкових чисел, що робить тему актуальною на сьогоднішній день.

Тема роботи затверджена наказом № 67 ст від 07.12.2021 .

Мета роботи – розробка програмного клієнт-серверного комплексу генерації випадкових та псевдовипадкових чисел.

Дана робота складається зі вступу, 5 розділів та висновків.

1 ОГЛЯД ГЕНЕРАТОРІВ ЧИСЛОВИХ ПОСЛІДОВНОСТЕЙ

1.1 Загальні відомості

Генератор числових послідовностей (ГПЧ) - процес, який за допомогою пристроїв генерує послідовність чисел або символів [1].

Оскільки, сучасні генератори випадкових чисел використовуються в різних сферах криптографії, вони повинні відповідати наступним вимогам:

- статистична стійкість;
- довгий період;
- непередбачуваність;
- ефективність;
- переносимось;
- пропуск кусків генерації;
- правильна ініціалізація;
- відтворюваність [2].

Процес генерації чисел поділяють на дві групи - випадкову та псевдовипадкову.

Генератори які відтворюють випадкову послідовність чисел називають недетерміновані генератори, а ті, які генерують псевдовипадкову послідовність називають детерміновані генератори (ДГВП), або алгоритмічні. Недоліки та переваги процесів генерації чисел наведено в таблиці 1.1

Таблиця 1.1 - Недоліки та переваги процесів генерації чисел

Характеристика	Випадкова генерація	Псевдовипадкова генерація
Відсутність періодичності	Так	Ні
Непередбачуваність	Так	Умовна
Незалежність значень	Так	Умовна
Рівень крипостійкості	Високий	Умовний
Швидкість генерації	Низька	Висока
Відтворюваність	Ні	Так
Простота генерації	Ні	Так
Вартість генерації	Висока	Низька

1.2 Генератори чисел

1.2.1 Види генераторів чисел

За способом отримання чисел генератори чисел поділяють на:

- алгоритмічні;
- табличні;
- фізичні.

1.2.2 Алгоритмічні генератори випадкових чисел

Алгоритмічними генераторами чисел також називають детерміновані генератори випадкових чисел, оскільки, згенеровані числа являються псевдовипадковими, тобто підпорядковуються формулі (1.1).

$$r_{x+1} = f(r_x) \tag{1.1}$$

Основними вимогами, що висуваються до ДГВП, є необоротність, нерозрізненість, непередбачуваність, просторова та тимчасова складність, відновлюваність в просторі і часі, гарантований період повторення, основа алфавіту, тощо [3].

1.2.3 Лінійно конгруентний метод

В лінійно конгруентному методі числа генеруються по формулі (1.2)

$$r_{i+1} = mod(k * r_i + b, M) \tag{1.2}$$

де M — модуль ($0 < M$);

k — множник ($0 \leq k < M$);

b — приріст ($0 \leq b < M$);

r_0 — початкове значення ($0 \leq r_0 < M$).

Послідовність чисел, які отримуються за допомогою формули (1.2), називають лінійно-конгруентною послідовністю. Для роботи генератора треба підібрати правильні коефіцієнти.

1.2.4 Алгоритм BBS

Одним з найвідоміших алгоритмів генерації псевдовипадкових чисел є алгоритм BBS. Алгоритм генерації псевдовипадкових чисел BBS був запропонований в 1986 році Ленором Блумом, Мануелем Блумом та Майклом Шубом. Принцип роботи генератора BBS базується на використанні односторонньої функції факторизації [4]. Класичний алгоритм генерації чисел використовує формулу (1.3).

$$x_{n+1} = x_n^2 \, mod \, M \tag{1.3}$$

де x_n - поточне число, M - число яке є результатом множення двох непарних простих чисел q та p (див фор. 1.4).

$$M = q * p \tag{1.4}$$

де q та p прості числа для яких виконуються такі умови (див фор. 1.5):

$$p \equiv 3 (mod\ 4), q \equiv 3 (mod\ 4) \tag{1.5}$$

1.2.5 Алгоритм LFSR

Також треба згадати алгоритм генерації псевдовипадкових чисел LFSR.

Алгоритм LFSR полягає в використанні n-розрядного регістру здвигу, який проектують використовуючи D – тригери [5]. Генерація ґрунтується на використання полінома (1.6).

$$\varphi(x) = 1 \oplus \alpha_1 x^1 \oplus \alpha_2 x^2 \oplus \cdots \oplus \alpha_{n-1} x^{n-1} \oplus \alpha_n x^n \tag{1.6}$$

де n – число розрядів, а α_i коефіцієнти, $\alpha_i \in \{0,1\}$ (i = 1…n)

Приклад роботи алгоритму LFSR наведений на рис 1.3

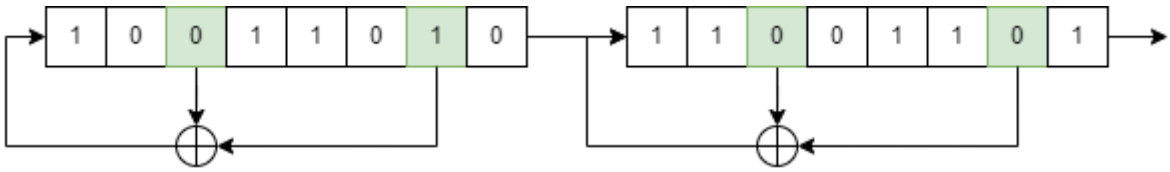

Рисунок 1.2 - Робота алгоритму LFSR

1.2.6 Фізичні генератори випадкових чисел

Генератор випадкових послідовностей – це механізм генерації випадкових послідовностей, у якому для генерації випадкового потоку використовується джерело ентропії, що ґрунтується на фізично випадкових явищах чи випадковому явищі. Прикладом простого ГВЧ може виступати монета, гральний кубик, колода карт [6].

Апаратні генератори числових послідовностей можна віднести до таких підкласів [7]:

- паразитні;
- генератори на основі ПЗС матриці;
- функціональні;
- квантові.

Паразитні генератори використовують шуми електронних елементів на платі, такі як шуми резисторів. В аналогових фізичних генераторах використовують генератори шуму для задання певної ентропії під час генерації випадкових чисел. Старі ГПЧ, як правило, використовували методи дискретизації шуму, але вони вимагають аналогових схем, які часто вимогливі до ресурсів, оскільки вимагають спеціальної компоновки кремнію і їх важко інтегрувати в цифрові пристрої, особливо при геометрії кремнію, що постійно зменшується. Наприклад, як джерело теплового шуму можна використовувати МОП-транзистор. Потім його необхідно посилити за допомогою операційного підсилювача, перетворити на цифру за допомогою АЦП, а потім квантувати, і все це без усунення сигналу або перешкод джерела шуму. Це вимагає значного налаштування або автоматичного

зміщення нуля, щоб гарантувати те, що квантування призводить до дуже близьких ймовірностей отримання значення 0 або 1 із ГВЧ [8].

Приклад роботи паразитного генератора чисел наведений на рисунках 1.3.

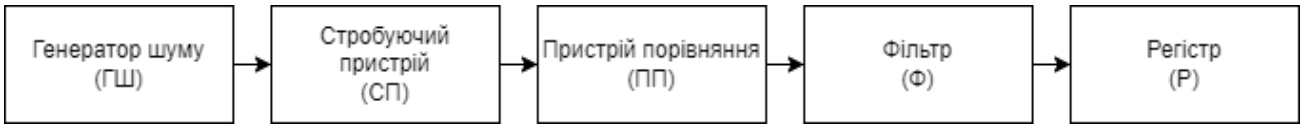

Рисунок 1.3 – Апаратна генерація випадкових чисел

Основними джерелами шуму в старих апаратних генераторах чисел можуть виступати такі шуми:

- тепловий шум від резистора;
- лавинний шум, генерований лавинним діодом;
- атмосферний шум, шум з ефіру.

Одним із сучасних генераторів випадкових чисел виступає Intel 82802 Firmware Hub. Intel 82802 Firmware Hub – хаб який містить систему BIOS та апаратний датчик випадкових чисел [9].

Квантові генератори випадкових чисел використовують квантові явища. Оскільки, результат квантово-механічних подій неможливо передбачити, вони являються золотим стандартом для генерації випадкових чисел. Квантові генератори випадкових чисел - це оптичні пристрої, які зараз розробляються для невеликих кремнієвих пристроїв, таких як спеціалізовані апаратні модулі безпеки.

Генератори на основі ПЗС матриці використовують ПЗС матрицю. ПЗС матриця – аналогова мікросхема із кремнію, складається з світлочутливих фотодіодів. В основі лежить технологія приборів з розрядним зв’язком, яка дозволяє зчитувати електричний потенціал [10].

Функціональні генератори використовують природний радіоактивний фон. Оскільки радіоактивний розпад являється природним випадковим процесом він добре підходить для джерела випадковості [11].

1.2.7 Табличні генератори випадкових чисел

Табличні генератори для генерації чисел використовують спеціальні таблиці, в яких числа являються некорельованими, тобто не залежать один від одного. Приклад таблиці наведено в табл. 1.3.

Таблиця 1.3 – Таблиця випадкових чисел для табличного генератора

Випадкова цифра				
5	6	9	7	8
9	7	7	0	3
4	2	9	0	0
1	8	8	3	1
4	3	9	2	6

Залежно від алгоритму вибору чисел одна й та ж сама таблиця може давати різні числові послідовності:

- якщо брати кожну другу цифру з таблиці то отримаємо послідовність 0.598, 0.973, 0.490, …

- якщо брати кожну третю цифру з таблиці то отримаємо послідовність 0.577, 0.391, 0.336, …

Перевагою такого метода є те, що числа є дійсно випадковими, оскільки вони не залежать один від одного. Недоліком виступає великий об'єм пам'яті який потрібний для зберігання цієї таблиці.

1.3 Висновки за розділом

Розглянуто:

- типи генераторів числових послідовностей (випадкові та псевдо випадкові);
- основні методи отримання випадкових та псевдовипадкових чисел.

2 СТРУКТУРА ПРОГРАМНОГО КОМПЛЕКСУ

2.1 Архітектура комплексу

Комплекс що розроблюється матиме клієнт-серверну архітектуру. Завдяки клієнт-серверній архітектурі сторонні процеси можуть отримувати від генератора числові послідовності, не задумуючись про процес генерації чисел. Для клієнт - серверної комунікації будуть використовуватися сокети з протоколом TCP, оскільки протокол TCP гарантує встановлення з'єднання між клієнтом та сервером, якщо це можливо [12]. Для передачі інформації між клієнтом та сервером використовуватиметься відправка JSON - запитів та відповідей. Архітектура програмного комплексу наведена на рисунку 2.1

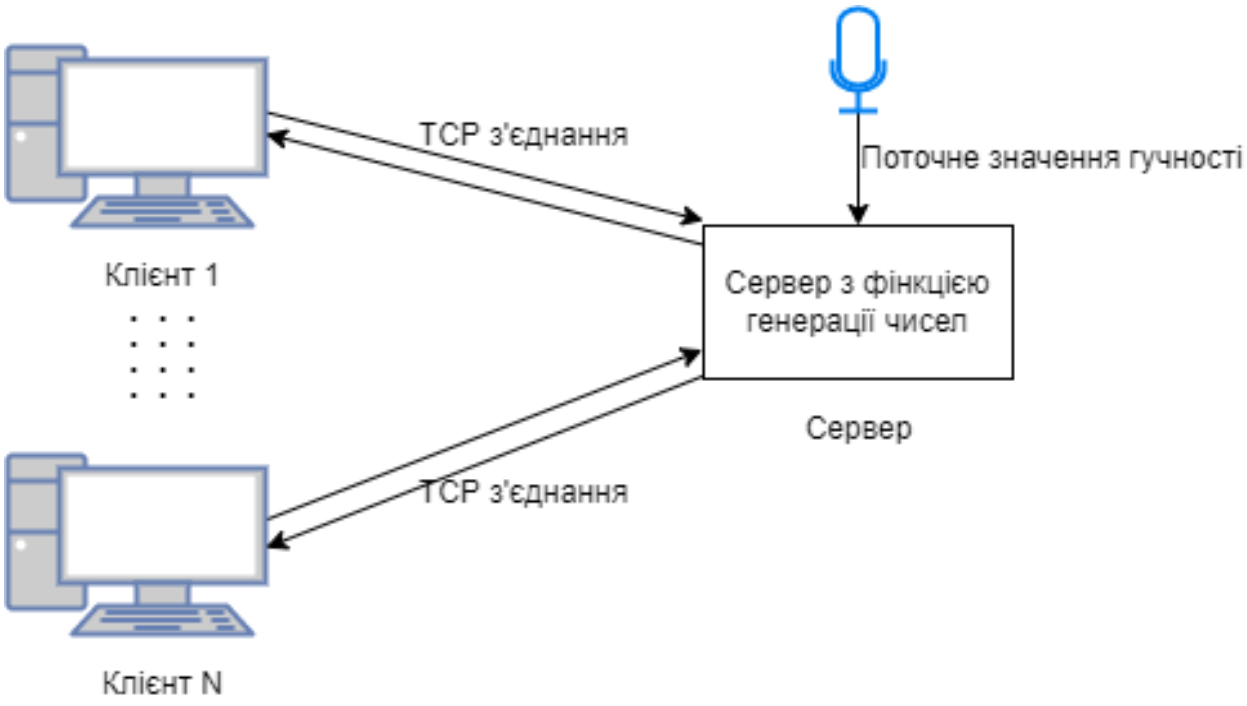

Рисунок 2.1 – Архітектура комплексу

Комплекс який розроблюється складається з клієнтської програми, яка буде імітувати запит стороннього процесу та серверу, який працює в двох режимах генерації числової послідовності:

- випадковий;
- псевдовипадковий.

Також під час роботи режимів розроблена можливість збереження числових послідовностей в файл. Постановка задачі описана в додатку Д.

2.2 Режими роботи комплексу

Для роботи випадкового режиму джерелом ентропії вибрано рівень гучності (шум) мікрофону, оскільки зняття значення гучності з мікрофону легше реалізувати ніж з іншими видами шумів.

У режимі генерації випадкових чисел генеруються числова послідовність, числа якої генеруються в діапазоні [mix; max) завдяки зіставленням з граничним значенням поточного значення гучності мікрофону. Процес генерації чисел складається з таких етапів:

- зчитування шуму;
- оцифрування шуму;
- зіставлення з граничним значенням;
- генерація бітів інформації.

Процес генерації чисел завдяки шуму мікрофону наведений на рисунку 2.2.

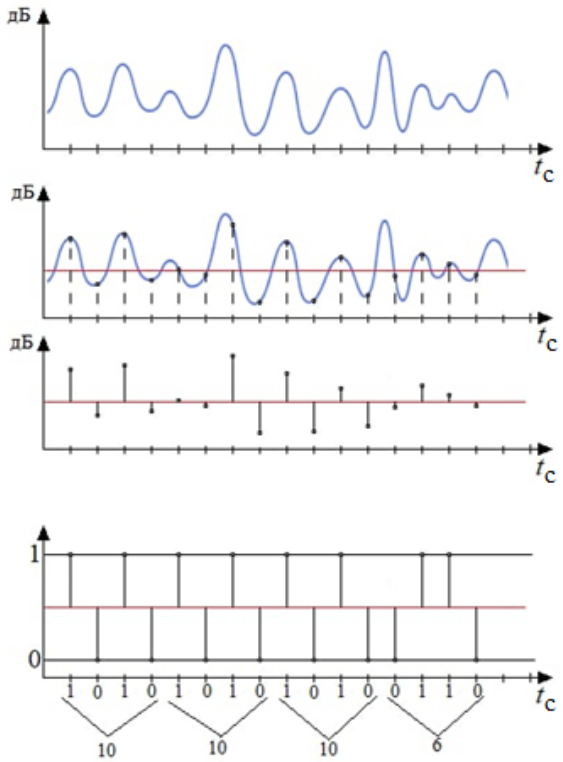

Рисунок 2.2 – Принцип генерації чисел з шуму мікрофону

Для роботи псевдовипадкового режиму обрано лінійно конгруентний метод, оскільки спираючись на критерій складності реалізації, лінійно конгруентний метод має перевагу над BBS та LFSR, це є доцільним у вирішенні поставленої задачі.

У режимі генерації псевдовипадкових чисел генеруються числова послідовність, числа якої генеруються в діапазоні [mix; max) використовуючи лінійно конгруентний метод, де в якості зерна може виступати або стандартне число або генероване випадкове число або задане користувачем зерно. Для роботи лінійно конгруентного методу можуть використовуватися стандарті аргументи або аргументи задані користувачем.

2.3 Організація взаємодії

Серверна частина використовує функціонал двох режимів. Сервер відкривається на порту 8090. Для обміну інформацією між сервером та клієнтом використовуються сокети з протоколом TCP. Обмін інформації відбувається за допомогою запитів та відповідей в форматі JSON. Процес комунікації починається з відправки клієнтом JSON запиту на генерацію числа. Структура JSON запиту наведена в таблиці 2.1.

Таблиця 2.1 – структура JSON запиту

mode	string
max	uint
min	uint
argA	uint
argC	uint
countNumbers	uint
newseed	bool
bordervalue	uint

Поле «Mode» виступає режимом роботи та приймає два значення: «Random» або «PseudoRandom», поля max та min – виступають діапазонами генерації чисел

[min;max) (використовуються в обох режимах), поля argA та argC - виступають значеннями аргументів при роботі в режимі «PseudoRandom», поле countNumbers – виступає довжиною числової послідовності (використовується в обох режимах), поле newseed – приймає два значення true або false, при роботі в режимі «PseudoRandom» вказує чи потрібно згенерувати нове зерно для алгоритму чи ні, поле bordervalue - виступає граничним значенням для генерації бітів числа(використовується в обох режимах). Загальній вигляд JSON запитів наведено на рисунку 2.3

Рисунок 2.3 - Приклади JSON запитів

Після отримання JSON запиту сервер обробляє запит та відправляє клієнту відповідь де передається генероване число. Структура JSON відповіді наведена в таблиці 2.2.

Таблиця 2.2 - Структура JSON відповідей

number	uint

Поле number містить генероване сервером число. Загальній вигляд JSON відповідей наведено на рисунку 2.4

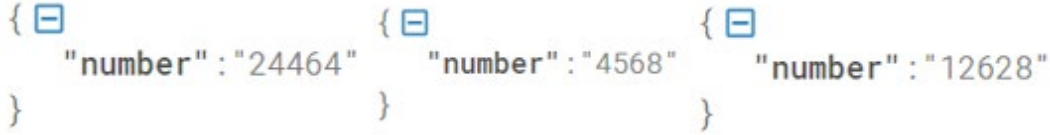

Рисунок 2.4 - Приклади JSON відповідей

2.4 Висновки за розділом

Представлено архітектуру програмного комплексу, який складається з клієнтських програм, що зв’язуються з серверною програмою. Програмний комплекс будується на реалізації двох методів генерації чисел, використання шумів мікрофону - для генерації випадкових чисел, лінійно конгруентний метод - для генерації псевдовипадкових чисел. Для комунікації вибрано комунікацію через TCP - сокет. Наведена структура запитів та відповідей між клієнтом та сервером. Прийнято рішення передавати інформацію між процесами за допомогою JSON - запитів та відповідей.

3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Вибір середовища та засобів розробки

В якості середовища розробки вибрано Visual Studio 2022, оскільки це середовище дозволяє зручно працювати з сторонніми бібліотеками. Середовище також дає багато інструментів для дебагу та аналізу працездатності програми.

В якості мови програмування вибрано C#. Оскільки на мові C# розроблена бібліотека яка дає змогу зручно та легко працювати з мікрофоном.

Бібліотека NAudio [13] дозволяє створювати аудіо файли в різних форматах, відтворювати звук а також зчитувати звук з мікрофону, що потрібно в умовах проекту.

Для встановлення комунікації між клієнтом та сервером буде використовуватися сокетний зв’язок з протоколом TCP, для цього використовується простір найменувань System.Net.Sockets [14].

Також для комунікації між клієнтом та сервером потрібно встановити бібліотеку Newtonsoft.Json [15] для роботи з даними в форматі JSON.

3.2 Основні алгоритми

В програмному комплексі реалізовано наступні функції:

- генерація випадкових чисел;
- генерація псевдовипадкових чисел;
- встановлення зв’язку з клієнтами;
- отримання та опрацювання запитів від клієнтів;
- відправлення відповіді клієнтам;
- встановлення зв’язку з сервером;
- відправка запитів серверу;
- отримання відповіді від серверу.

Головними функціями програмного комплексу – є функції генерації випадкових (рис. 3.1) та псевдовипадкових чисел (рис. 3.2). Вихідний код випадкового та псевдовипадкового режимів наведено в додатках А та Б відповідно.

Робота функції генерації випадкових чисел складається з 8 блоків.

Блок 1 - запуск функції генерації випадкових чисел.

Блок 2 - отримання аргументів для генерації випадкових чисел.

Блок 3 – цикл генерації 32-х бітного числа.

Блок 4 – зчитування гучності з мікрофону.

Блок 5 – зіставлення поточного значення гучності мікрофону з граничним значенням.

Блок 6 – генерація бітів випадкового числа.

Блок 7 – масштабування генерованого випадкового числа.

Блок 8 - завершення функції генерації випадкових чисел.

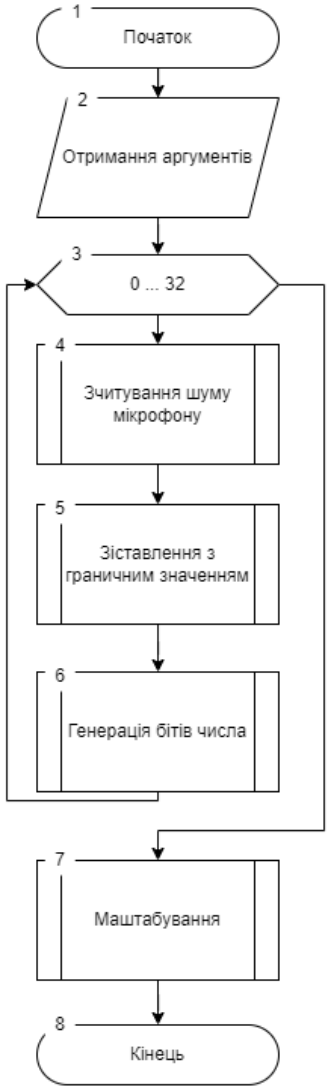

Рисунок 3.1 – Узагальнена блок схема роботи функції генерації випадкової числової послідовності

Робота функції генерації псевдовипадкових чисел складається з 7 блоків.

Блок 1 - запуск функції генерації псевдовипадкових чисел.

Блок 2 - отримання аргументів для генерації псевдовипадкових чисел.

Блок 3 – перевірка чи потрібно згенерувати нове зерно.

Блок 4 – генерація випадкового зерна для лінійно конгруентного методу.

Блок 5 – генерація псевдовипадкового числа за допомогою лінійного конгруентного методу.

Блок 6 – масштабування генерованого числа.

Блок 7 – завершення роботи функції генерації псевдовипадкових чисел.

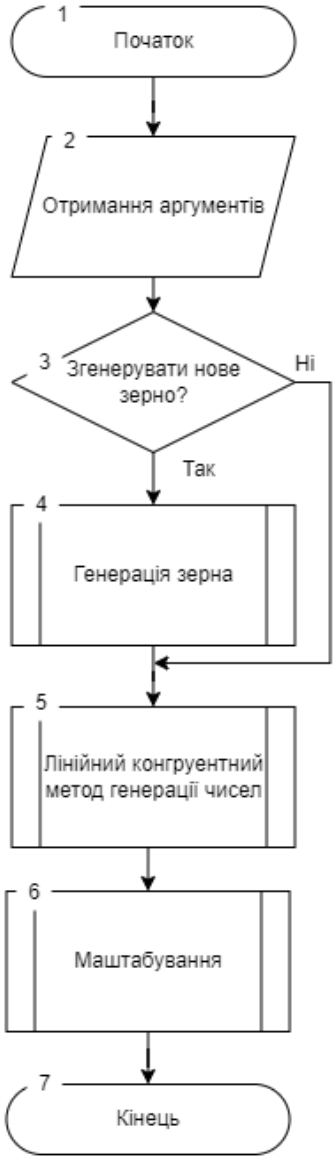

Рисунок 3.2 – Узагальнена блок схема роботи функції генерації псевдовипадкової числової послідовності

Під час запуску програми викликається функція яка відповідає за вибір режиму генерації (рис. 3.3) та функція запуску серверної частини (рис. 3.4). Вихідний код функції вибору режиму генерації та серверної частини наведено в додатку В.

Робота функції вибору режиму генерації складається з 6 блоків:

Блок 1 - запуск функції вибору генерації чисел.

Блок 2 - програма в іншому потоці викликає функцію яка відповідає за серверну частину .

Блок 3 - перевірка чи програма закінчила свою роботу.

Блок 4 - користувач вибирає режим роботи.

Блок 5 – залежно від вибраного режиму відкривається відповідний режим.

Блок 6 - завершення роботи функції вибору режим.

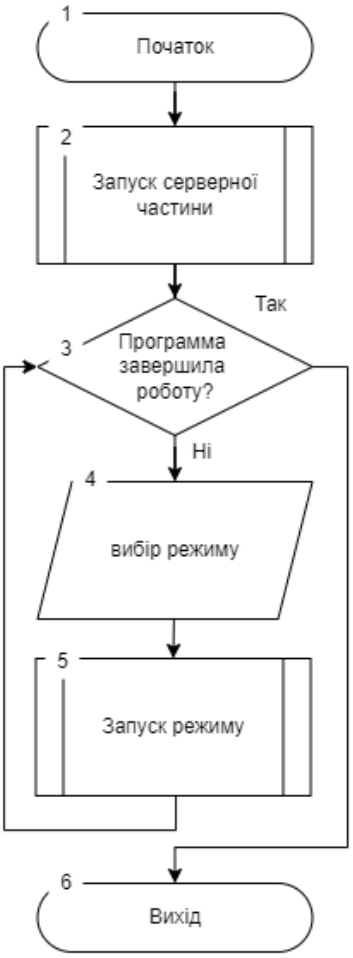

Рисунок 3.3 – Узагальнена блок схема роботи функції вибору режиму генерації

Робота функції яка відповідає за серверну частину складається 9 блоків.

Блок 1 - запуск функції яка відповідає за серверну частину.

Блок 2 - відбувається побудова TCP - сокету.

Блок 3 – відкриття TCP - сокету для комунікації.

Блок 4 – перевірка чи програма закінчила свою роботу.

Блок 5 – в сокет очікується передача запиту на генерацію чисел від клієнта.

Блок 6 – етап опрацювання запиту від клієнту де сервер взнає режим роботи генератора, аргументи генерації та кількість генерованих чисел.

Блок 7 – після того як генератор виконав свою роботу, сервер створює відповідь використовуючи генеровані блоці 6 числа.

Блок 8 – створена відповідь відправляється клієнту.

Блок 9 – завершення роботи функції яка відповідає за серверну частину.

Рисунок 3.4 – Узагальнена блок схема роботи серверної частини

Для передачі даних між клієнтом та сервером використовується функція комунікації (рис. 3.5). Вихідний код функції комунікації наведено в додатку Г.

Процес комунікації клієнта з сервером складається з 8-ми блоків.

Блок 1 - запуск функції яка відповідає за комунікацію між клієнтом та сервером.

Блок 2 – перевірка чи програма закінчила свою роботу.

Блок 3 - отримання аргументів для генерації чисел.

Блок 4 – підключення до сокету через який відбувається комунікація.

Блок 5 – формування запиту.

Блок 6 – відправлення запиту серверу.

Блок 7 – отримання відповіді, в якій передається генероване число, від серверу.

Блок 8 – завершення роботи функції яка відповідає за комунікацію між клієнтом та сервером.

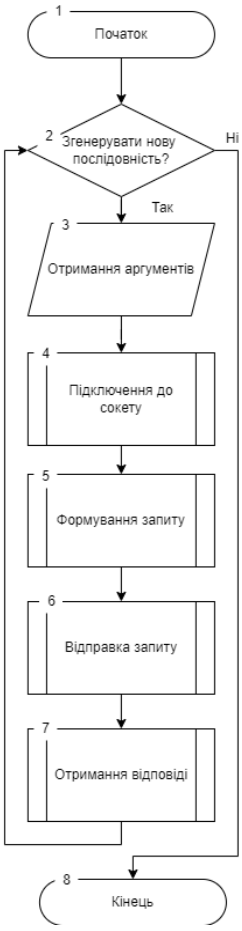

Рисунок 3.5 – Узагальнена блок схема роботи функції комунікації

3.3 Висновки за розділом

Вибрано середовище та мову розробки комплексу. Розглянуто функції програмного комплексу. Наведено блок схеми роботи для серверної частини, випадкового та псевдовипадкового режимів, комунікації на клієнтській стороні. Розроблено відповідне програмне забезпечення.

4 ПЕРЕВІРКА ПРАЦЕЗДАТНОСТІ

4.1 Перевірка працездатності у режимі генерації випадкових чисел

Під час роботи в випадковому режимі з аргументом кількість чисел 15 очікується 15 випадкових чисел в діапазоні [0;1000). Результат роботи випадкового режиму наведений на рисунку 4.1

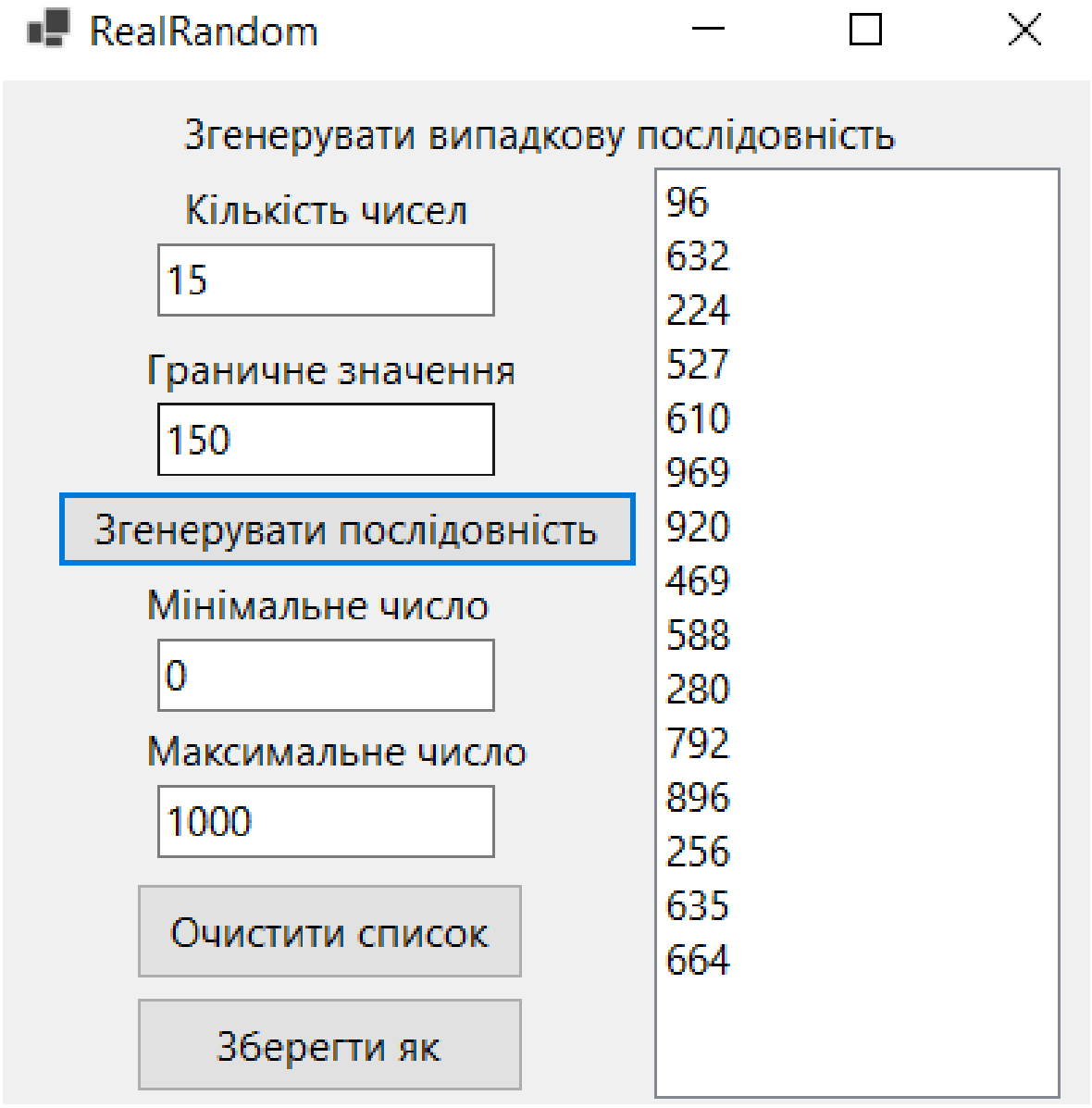

Рисунок 4.1 - Приклад роботи випадкового режиму

Під час перевірки режиму на випадковість чисел, було згенеровано 1000 випадкових чисел в діапазоні [0;100). Діапазони розподілу чисел представлені на рисунку 4.2

Рисунок 4.2 – Результат розподілу генерації 1000 випадкових чисел

4.2 Перевірка працездатності у режимі генерації псевдовипадкових чисел

Під час роботи в псевдовипадковому режимі з стандартними аргументами та аргументом кількість чисел 15 очікується генерація 15 чисел в діапазоні [0;1000). Результат роботи псевдовипадкового режиму з стандартними аргументами наведений на рисунку 4.3

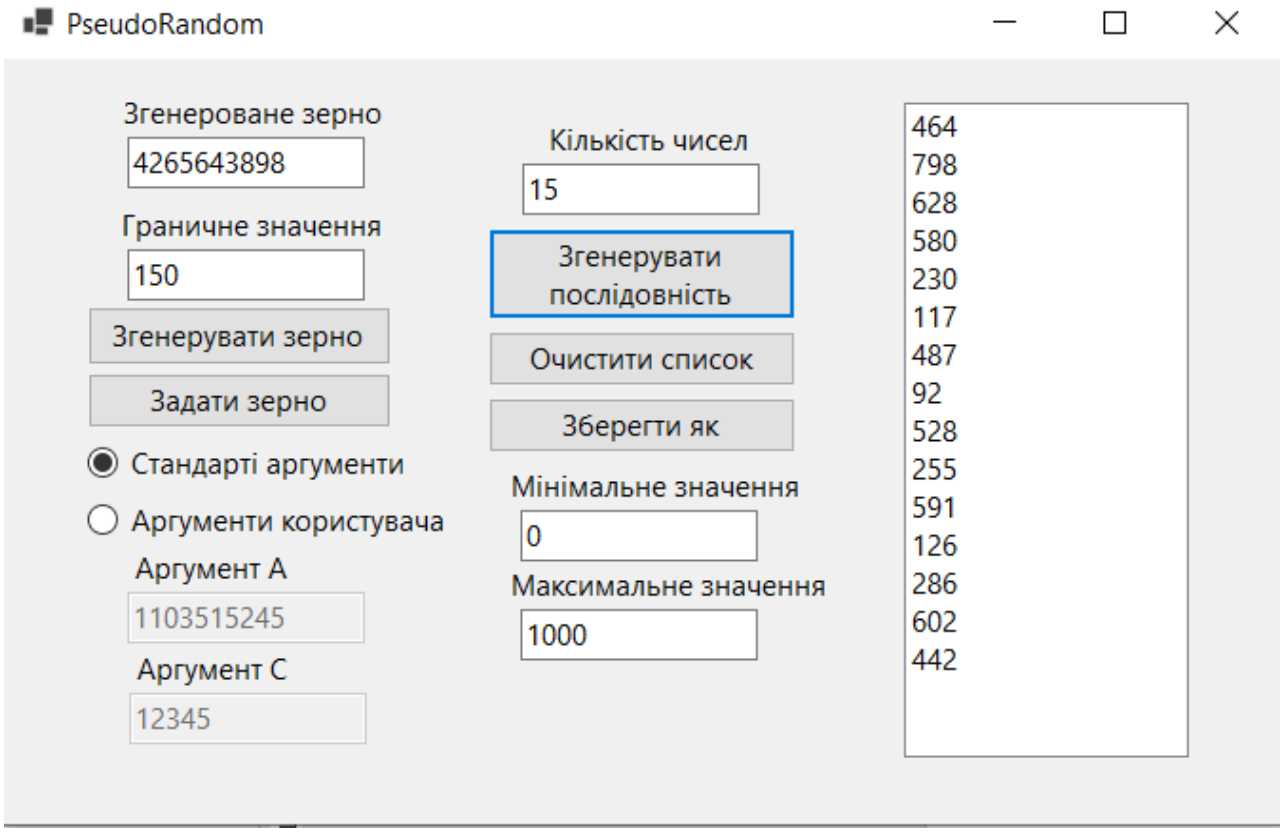

Рисунок 4.3 - Приклад роботи псевдовипадкового режиму з стандартними аргументами

Під час роботи в псевдовипадковому режимі з аргументами користувача та аргументом кількість чисел 15, оскільки аргумент a дорівнює нулю, очікується генерація 15 чисел, які будуть рівні 0. Результат роботи псевдовипадкового режиму з стандартними аргументами наведений на рисунку 4.4.

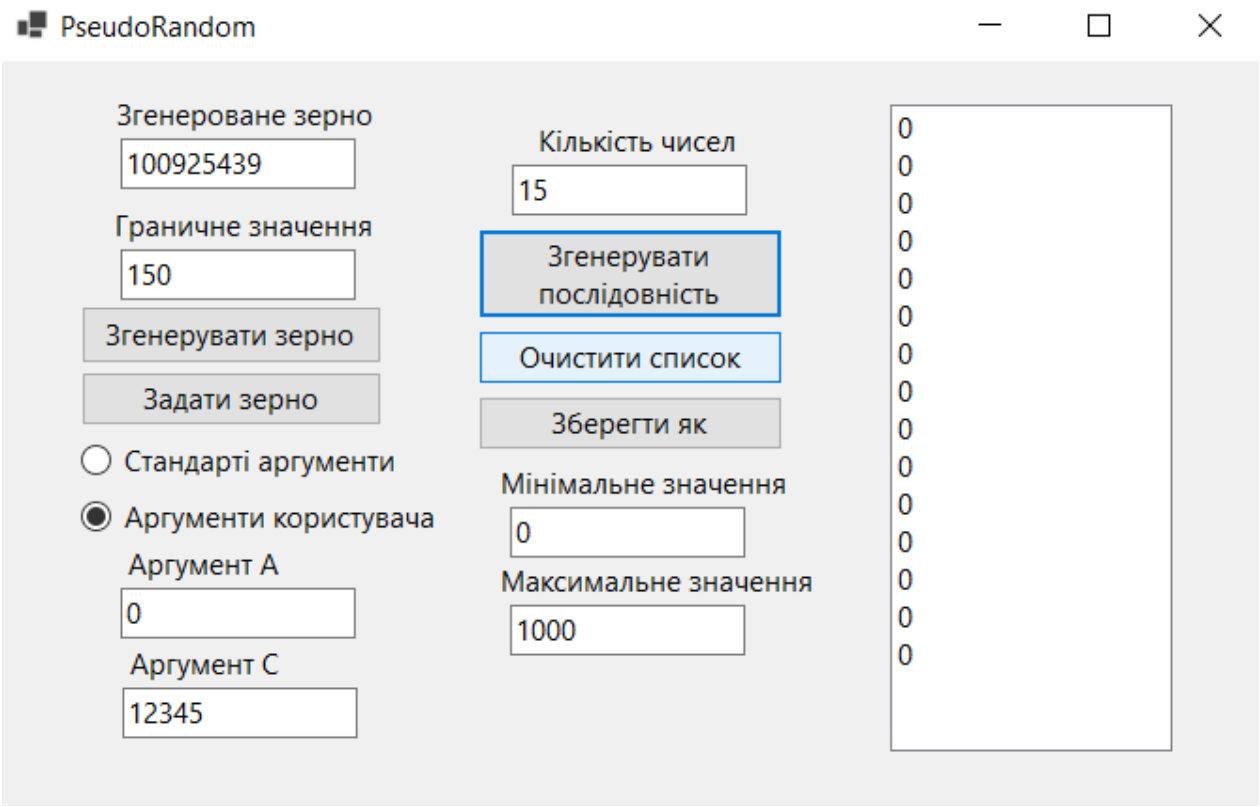

Рисунок 4.4 - Приклад роботи псевдовипадкового режиму з користувацькими аргументами

Під час перевірки режиму на випадковість чисел, було згенеровано 1000 псевдовипадкових чисел в діапазоні [0;100). Діапазони розподілу чисел представлені на рисунку 4.5

Рисунок 4.5 – Результат розподілу генерації 1000 псевдовипадкових чисел

4.3 Перевірка працездатності клієнтської частини

Під час роботи випадкового режиму через TCP зв'язок з аргументом кількість чисел 15, очікується 15 випадкових чисел в діапазоні [0;1000). Результат роботи випадкового режиму наведений на рисунку 4.6.

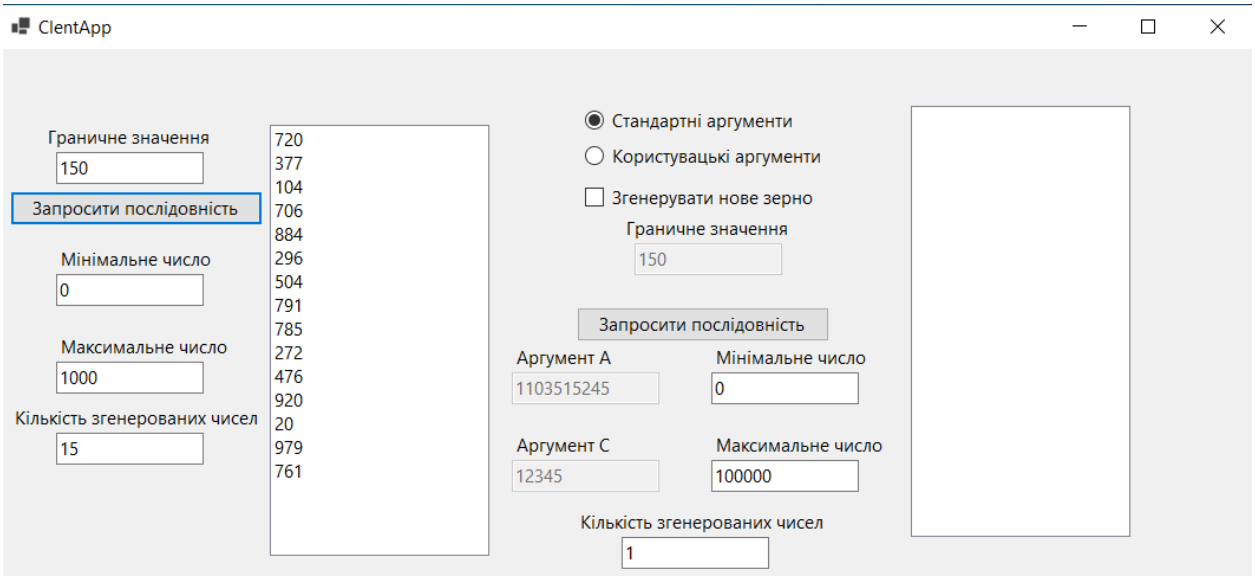

Рисунок 4.6 - Приклад роботи випадкового режиму через сторонній процес

Під час роботи псевдовипадкового режиму через TCP зв'язок з аргументом кількість чисел 15 очікується 15 псевдовипадкових чисел в діапазоні [0;1000). Результат роботи випадкового режиму наведений на рисунку 4.7.

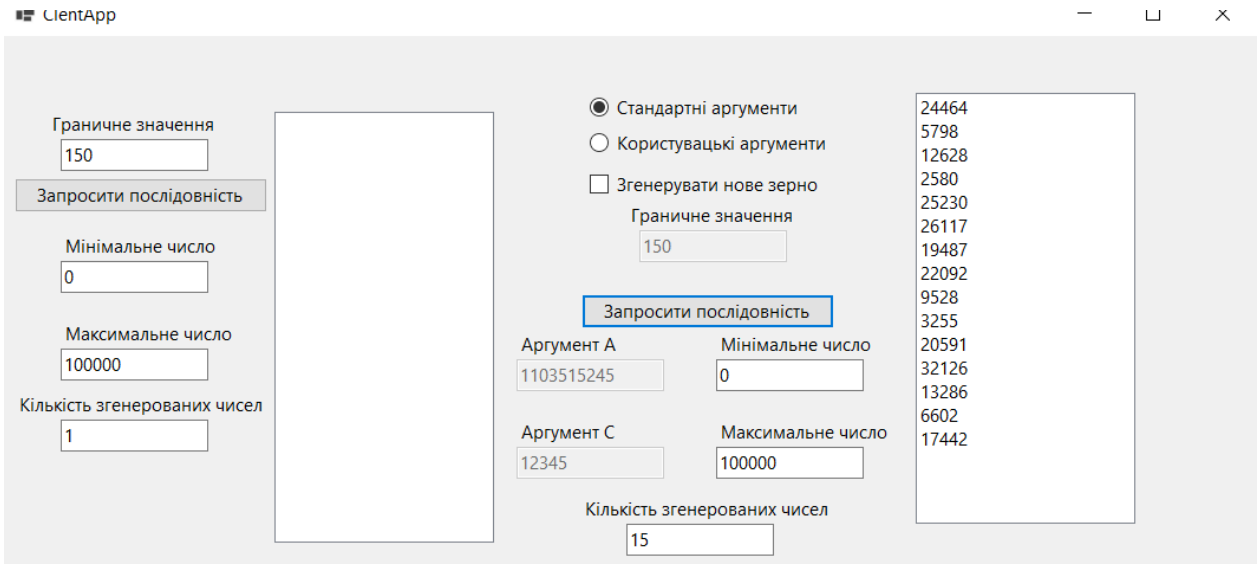

Рисунок 4.7 - Приклад роботи псевдовипадкового режиму через сторонній процес

Під час роботи псевдовипадкового режиму через TCP зв'язок одночасно два клієнти створюють запит на генерацію випадкових чисел. Результат роботи одночасного запиту наведено на рисунку 4.8.

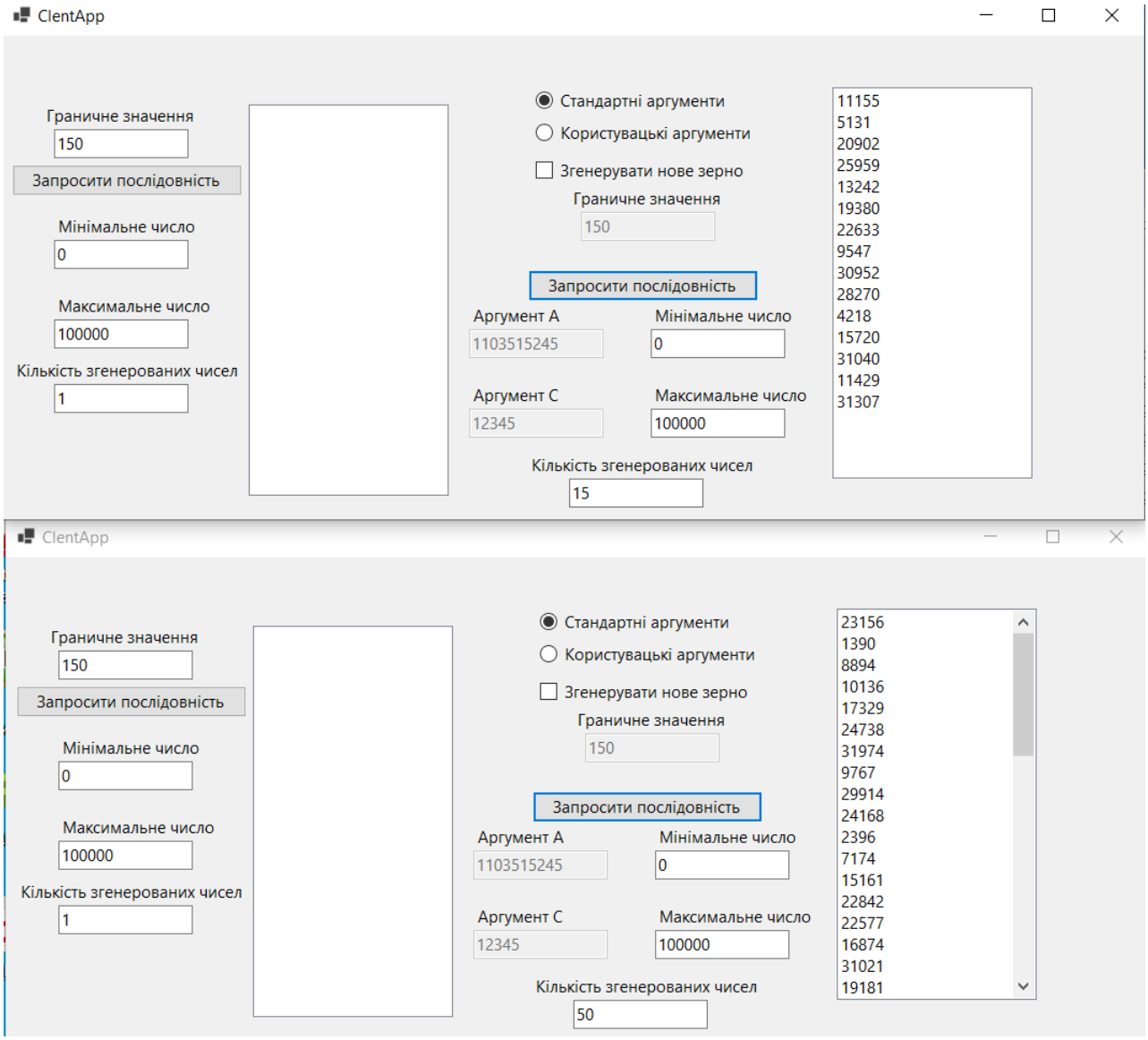

Рисунок 4.8 - Приклад роботи під одночасного запиту двох клієнтів

4.4 Висновки за розділом

Проведено перевірку працездатності основних можливостей комплексу:

- генерація випадкової послідовності в діапазоні [min, max);
- генерація псевдовипадкової послідовності з стандартними аргументами в діапазоні [min, max);
- генерація псевдовипадкової послідовності з користувацькими аргументами в діапазоні [min, max);
- запит від клієнта на генерацію випадкової послідовності в діапазоні [min, max);

- запит від клієнта на генерацію псевдовипадкової послідовності в діапазоні [min, max);
- одночасний запит від клієнтів на генерацію послідовностей.

Було проведено тести на рівномірній розподіл випадкових та псевдовипадкових чисел. Результати перевірки показали працездатність комплексу.

5 ІНСТРУКЦІЯ З ВИКОРИСТАННЯ

5.1 Інструкція з використання меню вибору режиму генерації

Меню вибору режиму генерації складається з двох кнопок: 1, 2. Кнопка 1 відкриває меню генерації випадкових чисел. Кнопка 2 відкриває меню генерації псевдовипадкових чисел. Загальний вигляд головного меню представлено на рисунку 5.1.

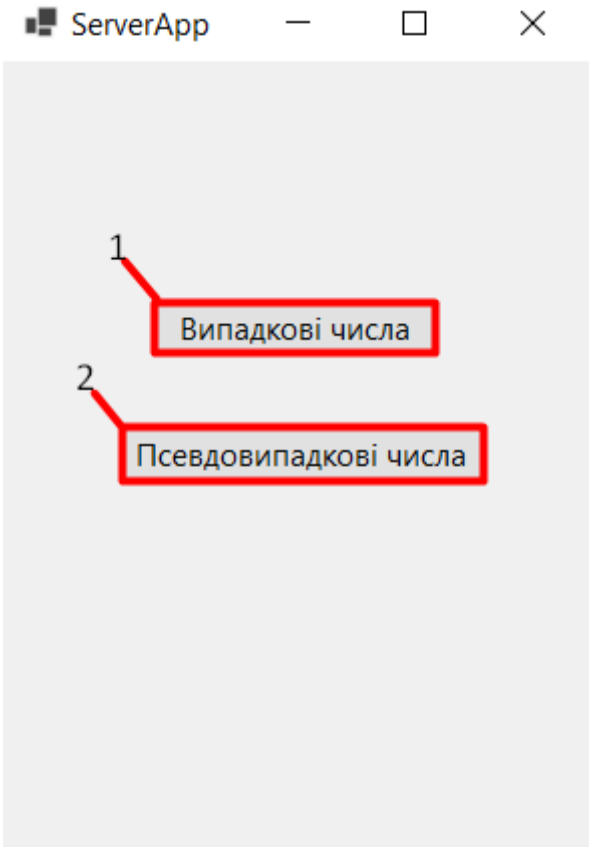

Рисунок 5.1 – Меню вибору режиму генерації

5.2 Інструкція з використання комплексу в режимі генерації випадкових чисел

Меню генерації випадкових чисел складається з кнопок 3,6,7 полів вводу інформації 1,2,4,5 та поле виводу інформації 8. В поле 1 вводиться кількість чисел які потрібно згенерувати. В поле 2 вводиться граничне значення яке використовується під час генерації бітів числа. Кнопка 3 генерує числа в кількості взяте з поля 1, результат генерації виводиться в список 8. В поле 4 вводиться мінімальна границя генерація числа. В поле 5 вводиться максимальна

границя генерація числа. Кнопка 6 очищає список 8. Кнопка 7 дозволяє зберегти дані з списку 8 в файл в форматі .txt. Загальний вигляд меню генерації випадкових чисел представлено на рисунку 5.2.

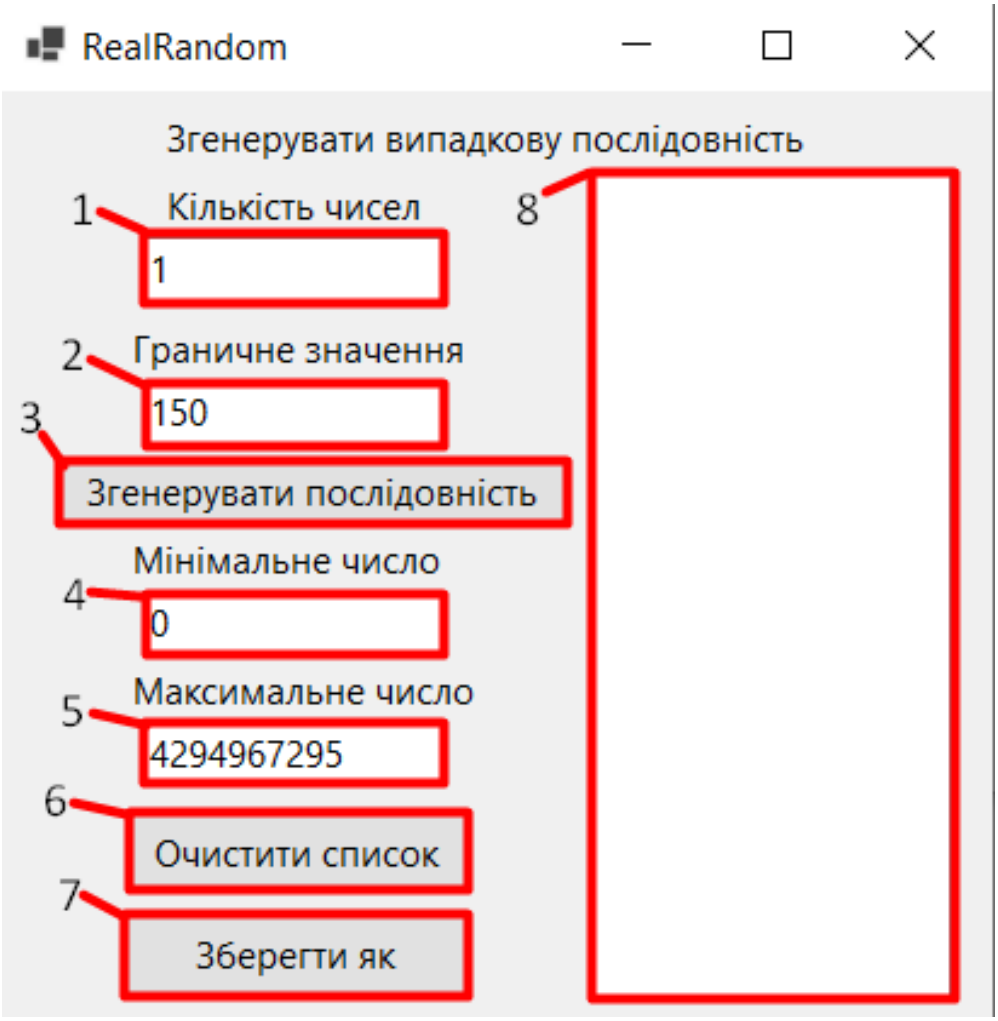

Рисунок 5.2 – Меню генерації випадкових чисел

5.3 Інструкція з використання комплексу в режимі генерації псевдовипадкових чисел

Меню генерації псевдовипадкових чисел складається з кнопок 3, 4, 10, 11, 12 полів вводу інформації 1, 2, 7, 8, 9, 13, 14 радіокнопок 5, 6 та поле виводу інформації 15. В поле 1 вводиться зерно для генерації псевдо випадкових чисел. В поле 2 вводиться граничне значення яке використовується під час генерації бітів зерна. Кнопка 3 генерує випадкове зерно, зберігаючи його для алгоритму генерації та відображає його в полі 1. Кнопка 4 зберігає значення зерна з поля 1 для алгоритму генерації. Радіокнопка 5 вказує, що генерація чисел відбувається

завдяки стандартним аргументам. Радіокнопка 6 дозволяє користувачу змінювати значення аргументів в полях 7 та 8. Поле 7 містить аргумент A для роботи лінійно конгруентного методу. Поле 7 містить аргумент C для роботи лінійно конгруентного методу. В поле 9 вводиться кількість чисел які потрібно згенерувати. Кнопка 10 генерує числа в кількості взяте з поля 9, аргументи для генерації беруться з полів 7 та 8, результат виводиться в список 15. Кнопка 11 очищає список 15. Кнопка 12 дозволяє зберегти дані з списку 15 в .txt файл. В поле 13 вводиться мінімальна границя генерація числа. В поле 14 вводиться максимальна границя генерація числа. Загальний вигляд меню генерації псевдовипадкових чисел представлено на рисунку 5.3.

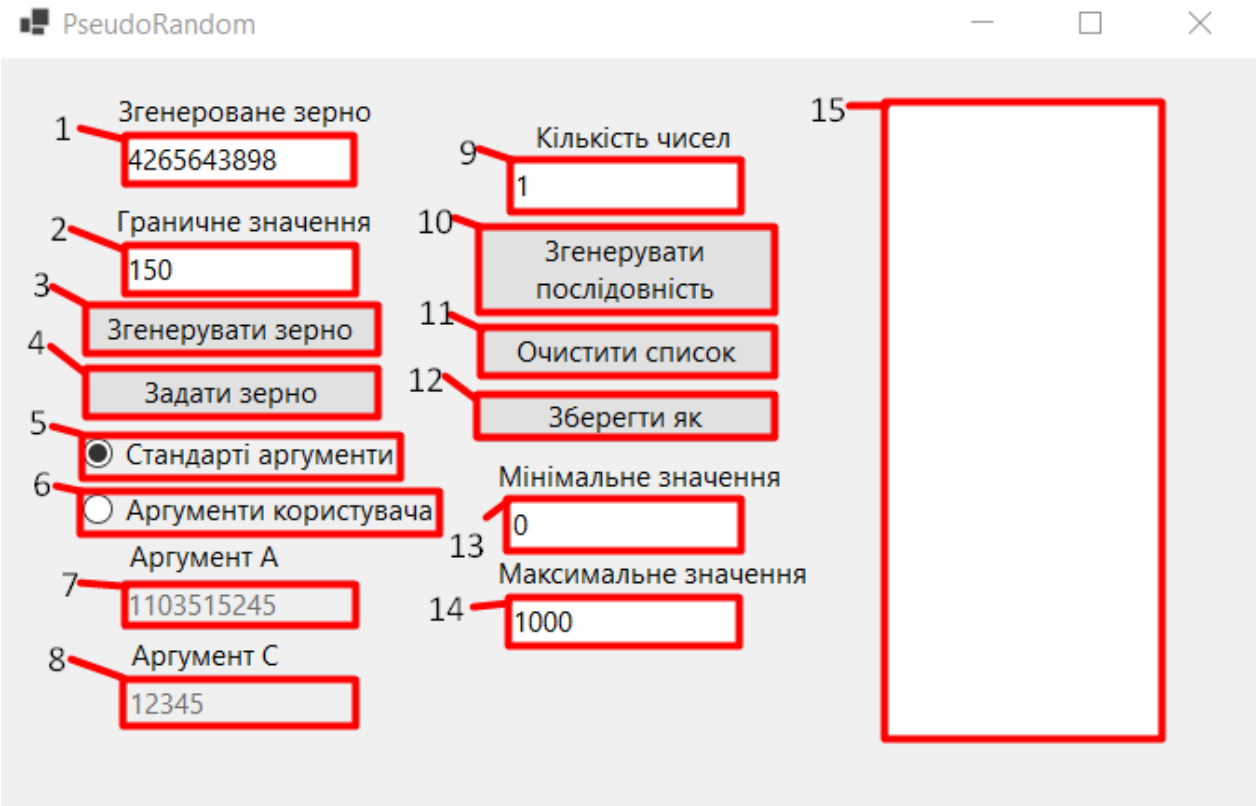

Рисунок 5.3 – Меню генерації псевдовипадкових чисел

5.4 Інструкція з використання клієнтської частини

Клієнтська частина складається з кнопок 2, 11 полів вводу інформації 1, 3, 4, 5, 9, 10, 12, 13, 14, 15, 16 чек боксу 9 радіокнопок 7, 8 та полів виводу інформації 6, 17.

В поле 1 вводиться граничне значення яке використовується в генерації бітів числа. Кнопка 2 відправляє запит на генерацію випадкового числа серверу. В поле 3 вводиться мінімальна границя генерація числа для випадкового режиму. В поле 4 вводиться максимальна границя генерація числа для випадкового режиму. В поле 5 вводиться кількість чисел які потрібно згенерувати в випадковому режимі. В списку 6 відображається результат роботи випадкового режиму. Радіокнопка 7 вказує шо генерація чисел відбувається завдяки стандартним аргументам. Радіокнопка 8 дозволяє користувачу змінювати значення в полях 12 та 13. Чекбокс 9 вказує що сервер згенерує нове зерно для генератора. В поле 10 вводиться граничне значення яке використовується в генерації бітів зерна в псевдовипадковому режимі. Кнопка 11 відправляє запит на генерацію псевдовипадкового числа серверу. Поле 12 містить аргумент A для роботи лінійно конгруентного методу. Поле 13 містить аргумент C для роботи лінійно конгруентного методу. В поле 14 вводиться мінімальна границя генерація числа. В поле 15 вводиться максимальна границя генерація числа. В поле 16 вводиться кількість чисел які потрібно згенерувати в псевдо випадковому режимі. В списку 17 відображається результат роботи псевдовипадкового режиму.

Загальний вигляд клієнтської частини представлено на рисунку 5.4.

Рисунок 5.4 – Меню клієнтської частини

5.5 Висновки за розділом

Створено інструкцію з використання клієнтської та серверної частин комплексу для обох режимів роботи.

ВИСНОВКИ

У даній роботі розроблено програмний клієнт-серверний комплекс, який генерує випадкові та псевдовипадкові числові послідовності, використовуючи шуми мікрофону.

Наведено загальні відомості про генератори числових послідовностей. Вирішено використовувати лінійно-когерентний метод при генерації псевдовипадкових чисел та шум мікрофону при генерації випадкових чисел.

Описано функціонування даного комплексу у двох режимах: випадковий та псевдо випадковий. Для більш зручного користування розроблено серверну частину. Створено структуру даних для JSON – запитів та відповідей.

Обрано середовище розробки, мову програмування та технологію для створення графічного інтерфейсу. Розроблено програмне забезпечення комплексу та створені відповідні блок-схеми узагальнених алгоритмів методів генерації чисел та клієнт-серверної взаємодії. Проведена перевірка працездатності програмного забезпечення. Створена інструкція з використання даного комплексу.

Розроблений комплекс може бути використаний на практиці для отримання випадкових та псевдовипадкових чисел та у навчальному процесі студентів відповідних спеціальностей при проведенні лабораторних і практичних робіт.

ПЕРЕЛІК ДЖЕРЕЛ

1. Генератор випадкових чисел послідовностей [Електронний ресурс]. – Режим доступу: https://inlnk.ru/VoN9jV.

2. Генератор випадкових чисел та паралельних потоків випадкових чисел для розрахунку Монте-Карло [Електронний ресурс]. – Режим доступу: https://www.researchgate.net/publication/328299090_Generation_of_Random_Numbers_and_Parallel_Random_Number_Streams_for_Monte_Carlo_Simulations/fulltext/5bc54454a6fdcc03c788c9cd/Generation-of-Random-Numbers-and-Parallel-Random-Number-Streams-for-Monte-Carlo-Simulations.pdf.

3. Горбенко Ю. І., Шапочка Н. В., Гріненко Т. О., Нейванов А. В. , Мордвінов Р. І. Методи та засоби генерування псевдовипадкових послідовностей [Текст] // Прикладная радиоэлектроника: науч.-техн. журнал. – 2011. – № 2. – С. 141–152..

4. Дослідження статичних характеристик модифікованого алгоритму BBS із залежністю від поточного та попереднього значення послідовності [Електронний ресурс].– Режим доступу:

https://www.researchgate.net/publication/342174713_Doslidzenna_statisticnih_harakteristik_modifikovanogo_algoritmu_BBS_iz_zaleznistu_vid_potocnogo_ta_poperednogo_znacenna_poslidovnosti.

5. Проектування конфігуруємого зсувного регістру з лінійним зворотнім зв`язком [Електронний ресурс].– Режим доступу:

https://www.researchgate.net/publication/321996508_PROEKTIROVANIE_KONFIGURIRUEMOGO_SDVIGOVOGO_REGISTRA_S_LINEJNOJ_OBRATNOJ_SVAZU.

6. Andrea Rock. Pseudorandom Number Generators for Cryptographic Applications / Andrea Rock // Diplomarbeit zur Erlangung des Magistergrades an der Naturwissenschaftlichen Fakultat der Paris-Lodron-Universitat Salzburg. – Salzburg. – 2005.

7. Апаратні генератори випадкових чисел [Електронний ресурс]. – Режим доступу: https://ozlib.com/1059417/informatika/apparatnye_generatory_sluchaynyh_chisel.

8. Огляд апаратних генераторів випадкових чисел [Електронний ресурс]. – Режим доступу: https://moluch.ru/archive/105/24688/.

9. Чіпсет Intel 820 [Електронний ресурс]. – Режим доступу: https://www.ixbt.com/mainboard/intel820.html.

10. Генератори на основі ПЗС матриці [Електронний ресурс]. – Режим доступу: https://www.powervideo.ru/slovar/pzs.html.

11. Генератори випадкових чисел на основі самовільного α-розпаду [Електронний ресурс]. – Режим доступу: https://patents.google.com/patent/EA003160B1/ru.

12. Принцип роботи протоколу TCP [Електронний ресурс]. – Режим доступу: http://elartu.tntu.edu.ua/bitstream/123456789/9607/2/Conf_2013v1_Radchuk_V-Printsip_roboti_protokolu_TCP_105.pdf.

13. Бібліотека NAudio [Електронний ресурс]. – Режим доступу: https://github.com/naudio/NAudio.

14. System.Net.Sockets [Електронний ресурс]. – Режим доступу: https://docs.microsoft.com/en-us/dotnet/api/system.net.sockets?view=net-6.0.

15. Бібліотека Newtonsoft.Json [Електронний ресурс]. – Режим доступу: https://github.com/JamesNK/Newtonsoft.Json.