

УДК 533.9.082+53.087.45

**РЕАЛИЗАЦИЯ ЛИНЕЙНОГО КОНГРУЭНТНОГО ГЕНЕРАТОРА  
ПСЕВДОСЛУЧАЙНЫХ ЧИСЕЛ С ГРАФИЧЕСКИМ ИНТЕРФЕЙСОМ НА PYTHON**

**Мидов Дамир Амирович, студент,  
Арванова Саният Мухамедовна, старший преподаватель,  
Казанов Тембулат Арсенович, студент,  
ФГБОУ ВО «КБГУ», г.Нальчик, Россия**

**REALIZATION OF LINEAR CONGRUENT PSEUDORANDOM NUMBER GENERATOR  
WITH GRAPHICAL INTERFACE IN PYTHON**

**Midov Damir Amirovich, student, damirmidov123@gmail.com,  
Arvanova Saniyat Mukhamedovna, Senior Lecturer, sani\_07@mail.ru,  
Kazanov Tembulat Arsenovich, student, Temba077@mail.ru,  
Kabardino-Balkarian State University, Nalchik, Russia**

*Представлена программная система на языке Python, реализующая линейный конгруэнтный генератор (ЛКГ) псевдослучайных чисел. Разработанное приложение включает графический интерфейс, созданный с использованием библиотеки tkinter, который позволяет задавать параметры генерации и отображать результаты. Описаны алгоритм ЛКГ, его программная реализация и возможности использования системы в образовательных целях. Продемонстрирована корректность работы программы на основе анализа сгенерированных чисел. Система может применяться для изучения методов генерации случайных чисел и моделирования процессов.*

**Ключевые слова:** линейный конгруэнтный генератор, псевдослучайные числа, Python, tkinter, графический интерфейс, генерация случайных чисел, образовательные технологии.

*A Python software system implementing a linear congruent generator (LCG) of pseudorandom numbers is presented. The developed application includes a graphical interface created using the tkinter library, which allows setting the generation parameters and displaying the results. The LKG algorithm, its program implementation and possibilities of using the system for educational purposes are described. The correctness of the program operation based on the analysis of the generated numbers is demonstrated. The system can be used to study the methods of random number generation and process modeling.*

**Keywords:** linear congruent generator, pseudorandom numbers, Python, tkinter, GUI, random number generation, educational technology.

Генерация псевдослучайных чисел является важной задачей в таких областях, как криптография [1], моделирование и статистика [2]. Линейный конгруэнтный генератор (ЛКГ) представляет собой один из наиболее простых и эффективных методов генерации псевдослучайных чисел, что делает его популярным выбором для образовательных и исследовательских целей [6].

Целью данной работы является создание программной системы на языке Python с графическим интерфейсом, реализованным с использованием библиотеки tkinter [3, 4], для генерации псевдослучайных чисел методом ЛКГ. Новизна исследования заключается в разработке доступного ин-

струмента, который может быть использован студентами и начинающими программистами для изучения принципов генерации случайных чисел.

Для достижения цели были выполнены следующие задачи:

1. Изучение теоретических основ ЛКГ на основе литературы [1, 2, 6].
2. Разработка программы на Python с использованием tkinter [3, 4, 5].
3. Тестирование системы и анализ сгенерированных чисел для подтверждения её корректности.

ЛКГ основан на рекуррентной формуле:

$$x_{n+1} = (a * x_n + c) \bmod m$$

где:

- $x_n$  — текущее псевдослучайное число,
- $a$  — множитель,
- $c$  — приращение,
- $m$  — модуль,
- $x_0$  — начальное значение (seed) [2, 6].

Качество последовательности определяется выбором  $a$ ,  $c$  и  $m$ . Например, при  $m = 2^{32}$ ,  $a = 1664525$ ,  $c = 1013904223$  достигается полный период длиной  $2^{32}$ [1]. Неправильный выбор параметров может привести к короткому периоду или низкой случайности чисел [6].

Программа представляет собой настольное приложение на Python с графическим интерфейсом, созданным с использованием библиотеки tkinter [4]. Пользователь может задавать параметры ЛКГ (начальное значение, множитель, приращение, модуль и количество чисел) и получать последовательность псевдослучайных чисел в текстовом поле.

Листинг 1 – Функция генерации псевдослучайных чисел

```
def linear_congruential_generator(seed, a, c, m, n):  
    """Генерация псевдослучайных чисел методом ЛКГ"""  
    numbers = []  
    x = seed  
    for _ in range(n):  
        x = (a * x + c) % m  
        numbers.append(x)  
    return numbers
```

Данный код реализует алгоритм генерации псевдослучайных чисел на основе линейного конгруэнтного метода. Он использует заданные параметры для создания последовательности чисел, начиная с начального значения и применяя определённые математические операции. Процесс повторяется нужное количество раз, чтобы сформировать требуемый набор псевдослучайных значений.

Листинг 2 – Инициализация главного окна

```
root = tk.Tk()  
root.title("Генератор псевдослучайных чисел (ЛКГ)")  
root.geometry("500x500")  
root.resizable(False, False)
```

Здесь создаётся основное окно приложения с установленным заголовком и фиксированными размерами. Оно спроектировано так, чтобы пользователь не мог изменять его размеры, что гарантирует стабильный и предсказуемый внешний вид интерфейса при работе с программой.

Листинг 3 – Создание фрейма и полей ввода

```
params_frame = ttk.LabelFrame(root, text="Параметры генератора")  
params_frame.pack(padx=10, pady=10, fill="x", expand=True)  
labels = ["Начальное значение (seed):", "Множитель (a):", "Приращение (c):", "Модуль (m):",  
"Количество чисел (n):"]  
entries = []  
for i, label_text in enumerate(labels):  
    ttk.Label(params_frame, text=label_text).grid(row=i, column=0, padx=5, pady=5, sticky="e")  
    entry = ttk.Entry(params_frame)  
    entry.grid(row=i, column=1, padx=5, pady=5)  
    entries.append(entry)
```

```
seed_entry, a_entry, c_entry, m_entry, n_entry = entries
```

Этот фрагмент отвечает за организацию области, где пользователь вводит параметры. В специальной рамке размещаются текстовые обозначения и поля для ввода данных, каждое из которых соответствует определённому параметру. Такое решение делает процесс ввода данных простым и понятным, улучшая взаимодействие с приложением.

Листинг 4 – Обработка ввода и вывод результатов

```
def generate_numbers():
    """Обработка нажатия кнопки и вывод результатов"""
    try:
        seed = int(seed_entry.get())
        a = int(a_entry.get())
        c = int(c_entry.get())
        m = int(m_entry.get())
        n = int(n_entry.get())
        if n <= 0 or m <= 0:
            raise ValueError("Число элементов и модуль должны быть положительными!")
    except ValueError as e:
        messagebox.showerror("Ошибка ввода", str(e) if str(e) else "Введите корректные числовые значения!")
    return
    numbers = linear_congruential_generator(seed, a, c, m, n)
    output_text.delete(1.0, tk.END)
    output_text.insert(tk.END, "Сгенерированные псевдослучайные числа:\n")
    for i, number in enumerate(numbers, 1):
        output_text.insert(tk.END, f'{i}. {number}\n')
```

Данный код управляет обработкой данных, которые ввёл пользователь. Он проверяет введённые значения на корректность, запускает процесс генерации псевдослучайных чисел при выполнении условий и отображает результаты в предназначенной для этого области интерфейса. При возникновении ошибок пользователю показывается соответствующее уведомление.

Листинг 5 – Добавление кнопки и текстового поля

```
generate_button = ttk.Button(root, text="Сгенерировать", command=generate_numbers)
generate_button.pack(pady=10)
output_text = tk.Text(root, wrap=tk.NONE, height=15)
output_text.pack(padx=10, pady=10, fill="both", expand=True)
root.mainloop()
```

Этот фрагмент завершает настройку интерфейса, добавляя элемент управления для запуска генерации чисел и область для отображения результатов. Кнопка активирует процесс обработки введённых данных, а текстовое поле служит для вывода сгенерированных значений. После этого приложение переходит в режим ожидания действий пользователя, обеспечивая полную интерактивность.

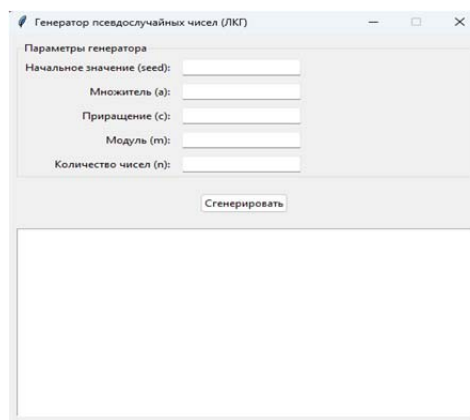


Рисунок 1. – Главное окно программы

Главное окно содержит заголовок «Генератор псевдослучайных чисел (ЛКГ)», фрейм «Параметры генератора» с пятью полями ввода, кнопку «Сгенерировать» и пустое текстовое поле для вывода результатов.

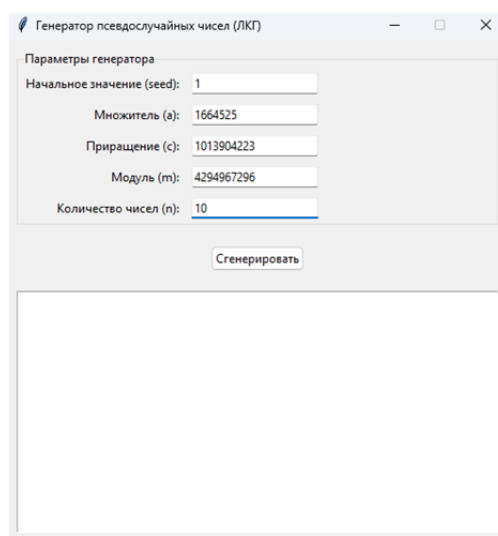


Рисунок 2. – Ввод параметров

Для проверки программы были выбраны параметры:

- $seed = 1$
- $a = 1664525$
- $c = 1013904223$
- $m = 2^{32} = 4294967296$
- $n = 10$

```
Сгенерированные псевдослучайные числа:  
1. 1015568748  
2. 1586005467  
3. 2165703038  
4. 3027450565  
5. 217083232  
6. 1587069247  
7. 3327581586  
8. 2388811721  
9. 70837908  
10. 2745540835
```

Рисунок 3. – Сгенерированная последовательность чисел

Сгенерированная последовательность подтвердила корректность реализации алгоритма ЛКГ. Числа обладают псевдослучайным характером, что соответствует теоретическим ожиданиям [1, 6].

В итоге, разработанная программная система успешно реализует линейный конгруэнтный генератор с графическим интерфейсом на Python. Она проста в использовании и может служить образовательным инструментом для изучения генерации случайных чисел. В будущем возможно добавление анализа качества последовательностей, например, тестов на равномерность распределения [2].

#### Список использованных источников

1. Шнайер, Б. Прикладная криптография. Протоколы, алгоритмы и исходные тексты на языке С. – М.: Триумф, 2002. — 816 с.
2. Кнут, Д. Искусство программирования, том 2: Получисленные алгоритмы. – М.: Вильямс, 2007. – 832 с.
3. Документация Python. URL: <https://docs.python.org/3/> (дата обращения: 01.10.2024).
4. Документация tkinter. URL: <https://docs.python.org/3/library/tkinter.html> (дата обращения: 01.10.2024).
5. Лутц, М. Изучаем Python. – СПб.: Питер, 2019. – 832 с.

6. Кормен, Т. Х., Лейзерсон, Ч. И., Ривест, Р. Л., Штайн, К. Алгоритмы: построение и анализ. — М.: Вильямс, 2013. — 1328 с.