**Лабораторна робота № 3**

**УПРАВЛЯЮЧІ КОНСТРУКЦІЇ В МОВІ GO. ФУНКЦІЇ. ВВЕДЕННЯ ТА ВИВЕДЕННЯ ІНФОРМАЦІЇ У КОНСОЛЬ**

**Мета:** ознайомитися з управляючими конструкціямимовипрограмування Go, навчитися користуватися функціями вводу/виводутастворювати користувацькі функції. Ознайомитись з методамигенеруваннявипадкових чисел.

**Хід роботи:**

**Завдання:   
**

Розробити функцію генерування послідовності псевдовипадкових цілочислових значень (за допомогою конгруентного методу) та виконати обробку отриманого масиву даних наступним чином (таблиця 3.2):

* розрахувати частоту інтервалів появи випадкових величин (інтервал дорівнює 1);
* розрахувати статистичну імовірність появи випадкових величин;
* розрахувати математичне сподівання випадкових величин;
* розрахувати дисперсію випадкових величин;
* розрахувати середньоквадратичне відхилення випадкових величин.

-Створюємо проект Golang

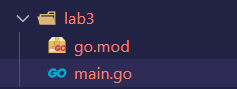


Рис.1. Реалізація завдання.

-Створюємо файл main.go

Реалізуємо функцію розрахунку статистичної імовірності появи випадкових величин.

Функція розрахунку статистичної імовірності появи випадкових величин реалізована наступним чином:

# Лістинг коду:

func CalculateProbability(frequency []int, n int) []float64 {

*// Створюємо масив статистичної імовірності появи випадкових величин.*

  probability := make([]float64, len(frequency))

  for i := 0; i < len(probability); i++ {

    probability[i] = float64(frequency[i]) / float64(n)

  }

  return probability

}

Реалізуємо функцію розрахунку математичного сподівання випадкових величин

Функція розрахунку математичного сподівання випадкових величин реалізована наступним чином:

# Лістинг коду:

func CalculateMean(sequence []int, n int) float64 {

*// Створюємо змінну для зберігання математичного сподівання.*

  mean := 0.0

  for \_, value := range sequence {

    mean += float64(value)

  }

  mean /= float64(n)

  return mean

}

Реалізуємо функцію розрахунку дисперсії випадкових величин

Функція розрахунку дисперсії випадкових величин реалізована наступним чином:

# Лістинг коду:

func CalculateVariance(sequence []int, mean float64, n int) float64 {

*// Створюємо змінну для зберігання дисперсії.*

  variance := 0.0

  for \_, value := range sequence {

    variance += float64((value - mean) \* (value - mean))

  }

  variance /= float64(n)

  return variance

}

Реалізуємо функцію розрахунку середньоквадратичного відхилення випадкових величин

Функція розрахунку середньоквадратичного відхилення випадкових величин реалізована наступним чином:

# Лістинг коду:

func CalculateStandardDeviation(variance float64) float64 {

*// Створюємо змінну для зберігання середньоквадратичного відхилення.*

  standardDeviation := math.Sqrt(variance)

  return standardDeviation

}

Реалізуємо функцію генерування псевдовипадкових цілочислових значень

Функція генерування псевдовипадкових цілочислових значень реалізована наступним чином:

# Лістинг коду:

func GeneratePseudoRandomIntegers(a, c, m, n int) []int {

*// Створюємо послідовність псевдовипадкових цілочислових значень.*

    sequence := make([]int, n)

    x := rand.Intn(m)

    for i := 0; i < n; i++ {

      sequence[i] = (a \* x + c) % m

    }

    return sequence

  }

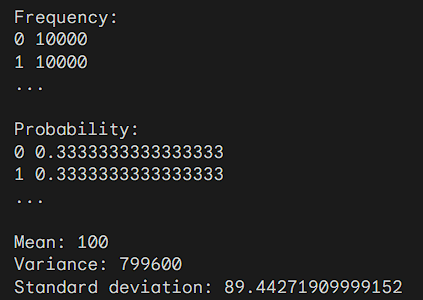


Рис.2. Реалізація завдання.

**Висновок:** У ході виконання завдань я отримав уявлення про основні управляючі конструкції, функції вводу/виводу і методи генерування випадкових чисел у Go.