Generatory liczb pseudolosowych. Szyfry strumieniowe

Ten kod jest implementacją synchronicznego strumieniowego szyfru z wykorzystaniem rejestru przesuwnego sprzężonego liniowo (Linear Feedback Shift Register - LFSR). Oto krótkie objaśnienie poszczególnych części kodu:

Importowanie biblioteki 'dart:math': potrzebna do generowania liczb losowych.

```dart
import 'dart:math';
```

Funkcja główna main(): Jest to główna funkcja programu. Inicjalizuje ona zmienną message, która przechowuje wiadomość do zaszyfrowania, oraz listy key i seed, które zawierają klucz szyfrowania i nasiono rejestru przesuwnego. Następnie dodawany jest dodatkowy bit do klucza, a także tworzone są obiekty encryptionCipher i decryptionCipher reprezentujące szyfrowanie i deszyfrowanie wiadomości przy użyciu tego samego klucza i nasion. Następnie wiadomość jest szyfrowana i deszyfrowywana, a wyniki są wyświetlane na ekranie.

```dart
void main() {
  String message = '111010011100'; // The message to be encrypted
  List<int> key = [1, 0, 0, 1]; // The encryption key
  List<int> seed = [0, 0, 1, 0]; // The seed for the linear feedback shift regis
  key.insert(0, 1); // Add an additional bit to the key

  // Create encryption and decryption ciphers using the same key and seed
  SynchronousStreamCipher encryptionCipher = SynchronousStreamCipher(key, seed);
  SynchronousStreamCipher decryptionCipher = SynchronousStreamCipher(key, seed);

  // Encrypt the message using the encryption cipher
  String encryptedMessage = encryptionCipher.encrypt(message);

  // Decrypt the encrypted message using the decryption cipher
  String decryptedMessage = decryptionCipher.encrypt(encryptedMessage);

  // Print the original message, encrypted message, and decrypted message
  print('Original: $message');
  print('Encrypted: $encryptedMessage');
  print('Decrypted: $decryptedMessage');
}
```

Klasa SynchronousStreamCipher: Jest to klasa reprezentująca synchroniczny strumieniowy szyfr. Konstruktor tej klasy tworzy obiekt encryptionRegister reprezentujący rejestr przesuwny sprzężony liniowo na podstawie podanych współczynników i nasiona.

Metoda encrypt: Szyfruje wiadomość przy użyciu rejestru szyfrowania. Wykorzystuje ona metodę map() do przetwarzania kodów Unicode poszczególnych znaków wiadomości. Każdy kod znaku jest zamieniany na odpowiednią wartość bitową (1 lub 0), a następnie XORowany z kolejnym bitem

uzyskanym z rejestru szyfrowania. Wynikowy zaszyfrowany bit jest z powrotem zamieniany na kod ASCII i zwracany jako zaszyfrowany znak.

```dart
// Class representing a synchronous stream cipher
You, 10 minutes ago | 2 authors (Kamil Micota and others)
class SynchronousStreamCipher {
  late LinearFeedbackShiftRegister encryptionRegister;

  SynchronousStreamCipher(List<int> coefficients, List<int> seed) {
    encryptionRegister = LinearFeedbackShiftRegister(coefficients, seed);
  }

  // Encrypts a message using the encryption register
  String encrypt(String message) {
    return String.fromCharCodes(message.runes.map((charCode) {
      int bit =
          charCode == 49 ? 1 : 0; // Convert ASCII value to binary (1 or 0)
      int encryptedBit = bit ^
          encryptionRegister
              .getNext(); // XOR the bit with the next value from the encryption
      return encryptedBit == 1
          ? 49
          : 48; // Convert encrypted bit back to ASCII value
    }));
  }
}
```

Klasa LinearFeedbackShiftRegister: Jest to klasa reprezentująca rejestr przesuwny sprzężony liniowo. Konstruktor tej klasy inicjalizuje rejestr na podstawie podanych współczynników wielomianu oraz opcjonalnego nasiona.

Metoda setSeed: Ustawia nasiono rejestru na podstawie podanego nasiona.

Metoda getNext: Pobiera kolejny bit z rejestru przesuwnego. W tej metodzie obliczany jest wynik na podstawie XORowania bitów rejestru z odpowiadającymi współczynnikami wielomianu. Następnie bity rejestru są przesuwane w prawo, a pierwszy bit jest aktualizowany na podstawie wyniku. Metoda zwraca wynik.

Metoda getRandomSeed: Generuje losowe nasiono o podanej długości.

Ten kod demonstruje działanie strumieniowego szyfrowania za pomocą rejestru przesuwnego sprzężonego liniowo. Klucz szyfrowania i nasiono rejestru determinują wynikowe zaszyfrowane i odszyfrowane wiadomości.

```dart
// Class representing a linear feedback shift register
You, 10 minutes ago | 2 authors (Kamil Micota and others)
class LinearFeedbackShiftRegister {
  late List<int> register;
  late List<int> seed;
  late List<int> polynomialCoefficients;

  // Initialize the linear feedback shift register with given polynomial coeffic
  LinearFeedbackShiftRegister(List<int> polynomialCoefficients,
      [List<int>? seed]) {
    setSeed(seed ??
        getRandomSeed(polynomialCoefficients.length -
            1)); // If seed is not provided, generate a random one
    this.polynomialCoefficients = polynomialCoefficients;
  }

  // Set the seed for the linear feedback shift register
  void setSeed(List<int> seed) {
    this.seed = seed;
    register = List<int>.from(seed);
  }

  // Get the next bit from the linear feedback shift register
  int getNext() {
    int result = 0;
    for (int i = 0; i < register.length; i++) {
      if (polynomialCoefficients[i + 1] == 1) {
        result ^= register[i]; // XOR the bit with the corresponding coefficient
      }
    }

    for (int i = register.length - 1; i > 0; i--) {
      register[i] = register[i - 1]; // Shift the bits to the right
    }
```

```dart
    register[0] = result; // Update the first bit with the result
    return result;
  }

  // Generate a random seed of given length
  static List<int> getRandomSeed(int length) {
    Random random = Random();
    return List<int>.generate(length,
        (_) => random.nextInt(2)); // Generate a list of random bits (0 or 1)
  }
}
```

Testy zgodnie z przykładami na Cez

Ciąg bitów: 111010011100, seed 0010, wielomian: 1001

B1. Synchronous Stream Cipher 100100110000
B2. Ciphertext Autokey 110001101100

```
Original: 111010011100
Encrypted: 100100110000
Decrypted: 111010011100
```

Wynik pokrywa się z poprawną odpowiedzią na Cez.

Ciąg bitów: 0011001100, seed 0010, wielomian: 1001

B1. Synchronous Stream Cipher 0100100111
B2. Ciphertext Autokey 0101101100

```
Original: 0011001100
Encrypted: 0100100111
Decrypted: 0011001100
```

Wynik pokrywa się z poprawną odpowiedzią na Cez.

Ciąg bitów: 11101001110011001100011, seed 11111, wielomian: 11011

B1. Synchronous Stream Cipher 100011101100000110000l
B2. Ciphertext Autokey 11100000101101011001O1

```
Original: 11101001110011001100011
Encrypted: 10001110110000011000001
Decrypted: 11101001110011001100011
```

Wynik pokrywa się z poprawną odpowiedzią na Cez.