Учреждение образования

«Минский государственный колледж цифровых технологий»

Учебный предмет

«Защита компьютерной информации»

Отчет

по лабораторной работе №4

Тема «Реализация генератора псевдослучайной последовательности »

|  |  |
| --- | --- |
| Выполнил:  учащийся 82ТП | Шатуха А.К. |
| Проверил:  преподаватель | Оскерко В.С. |

Минск, 2024

**Цель работы:** **изучить и закрепить умение генерирования алгоритмов работы генераторов псевдослучайных чисел**

Порядок выполнения работы:

1. Изучить теоретический материал
2. Выбрать язык программирования
3. Выполнить практическую часть.
4. Оформить отчет.

Практическая часть

Задание№1

Выдайте на экран 10 случайных равномерно распределенных чисел в диапазоне:

* От 3 до 12, целые.
* Из множества {–3, 0, 6, 9, 12, 15}.
* От 3 до 12, вещественные.
* От –2,3 до 10,7 с шагом 0,1.
* Из множества {–30; 10; 63; 59; 120; 175}.
* Из множества {1; 0,1; 0,01; …; 10–15}.

#include <iostream>

#include <vector>

#include <random>

#include <iomanip>

#include <set>

using namespace std;

int main() {

setlocale(LC\_ALL, "ru");

random\_device rd;

mt19937 gen(rd());

// 1.

uniform\_int\_distribution<> dist1(3, 12);

cout << "Случайные целые числа от 3 до 12:\n";

for (int i = 0; i < 10; ++i) {

cout << dist1(gen) << " ";

}

cout << "\n\n";

// 2.

set<int> set2 = { -3, 0, 6, 9, 12, 15 };

vector<int> vec2(set2.begin(), set2.end());

cout << "Случайные числа из множества {-3, 0, 6, 9, 12, 15}:\n";

for (int i = 0; i < 10; ++i) {

cout << vec2[dist1(gen) % vec2.size()] << " ";

}

cout << "\n\n";

// 3.

uniform\_real\_distribution<> dist3(3.0, 12.0);

cout << "Случайные вещественные числа от 3 до 12:\n";

for (int i = 0; i < 10; ++i) {

cout << fixed << setprecision(2) << dist3(gen) << " ";

}

cout << "\n\n";

// 4.

cout << "Случайные числа от -2.3 до 10.7 с шагом 0.1:\n";

uniform\_int\_distribution<> dist4(-23, 107);

for (int i = 0; i < 10; ++i) {

std::cout << std::fixed << std::setprecision(1) << dist4(gen) / 10.0 << " ";

}

cout << "\n\n";

// 5.

set<int> set5 = { -30, 10, 63, 59, 120, 175 };

vector<int> vec5(set5.begin(), set5.end());

cout << "Случайные числа из множества {-30, 10, 63, 59, 120, 175}:\n";

for (int i = 0; i < 10; ++i) {

cout << vec5[dist1(gen) % vec5.size()] << " ";

}

cout << "\n\n";

// 6.

vector<float> vec6;

for (int i = 0; i <= 15; ++i) {

vec6.push\_back(pow(10, -i));

}

cout << "Случайные числа из множества {1, 0.1, 0.01, ..., 10^-15}:\n";

for (int i = 0; i < 10; ++i) {

cout << vec6[dist1(gen) % vec6.size()] << " ";

}

cout << "\n";

return 0;

}

Задание№2

Напишите программу, моделирующую игру «Быки и коровы». Программа выбирает с помощью датчика случайных чисел четырехзначное число с разными цифрами. Цель игры – угадать это число. На каждом шаге играющий называет четырехзначное число, а программа сообщает, сколько цифр числа угадано (быки) и сколько угаданных цифр стоит на нужном месте (коровы).

#include <iostream>

#include <vector>

#include <random>

#include <set>

#include <algorithm>

bool hasUniqueDigits(int number) {

std::set<int> digits;

while (number > 0) {

int digit = number % 10;

if (digits.find(digit) != digits.end()) {

return false;

}

digits.insert(digit);

number /= 10;

}

return true;

}

int generateRandomNumber() {

std::random\_device rd;

std::mt19937 gen(rd());

int number;

do {

number = gen() % 9000 + 1000;

} while (!hasUniqueDigits(number));

return number;

}

std::pair<int, int> getBullsAndCows(int secret, int guess) {

int bulls = 0;

int cows = 0;

std::vector<int> secretDigits(4), guessDigits(4);

for (int i = 3; i >= 0; --i) {

secretDigits[i] = secret % 10;

guessDigits[i] = guess % 10;

secret /= 10;

guess /= 10;

}

for (int i = 0; i < 4; ++i) {

if (guessDigits[i] == secretDigits[i]) {

bulls++;

}

else if (std::find(secretDigits.begin(), secretDigits.end(), guessDigits[i]) != secretDigits.end()) {

cows++;

}

}

return { bulls, cows };

}

int main() {

setlocale(LC\_ALL, "ru");

int secretNumber = generateRandomNumber();

int guess;

int attempts = 0;

std::cout << "Добро пожаловать в игру «Быки и коровы»!" << std::endl;

std::cout << "Угадайте четырехзначное число с разными цифрами." << std::endl;

while (true) {

std::cout << "Введите вашу догадку (четырехзначное число): ";

std::cin >> guess;

if (guess < 1000 || guess > 9999 || !hasUniqueDigits(guess)) {

std::cout << "Ошибка: введите корректное четырехзначное число с уникальными цифрами." << std::endl;

continue;

}

attempts++;

std::pair<int, int> result = getBullsAndCows(secretNumber, guess);

int bulls = result.first;

int cows = result.second;

std::cout << "Быки: " << bulls << ", Коровы: " << cows << std::endl;

if (bulls == 4) {

std::cout << "Поздравляем! Вы угадали число " << secretNumber << " за " << attempts << " попыток!" << std::endl;

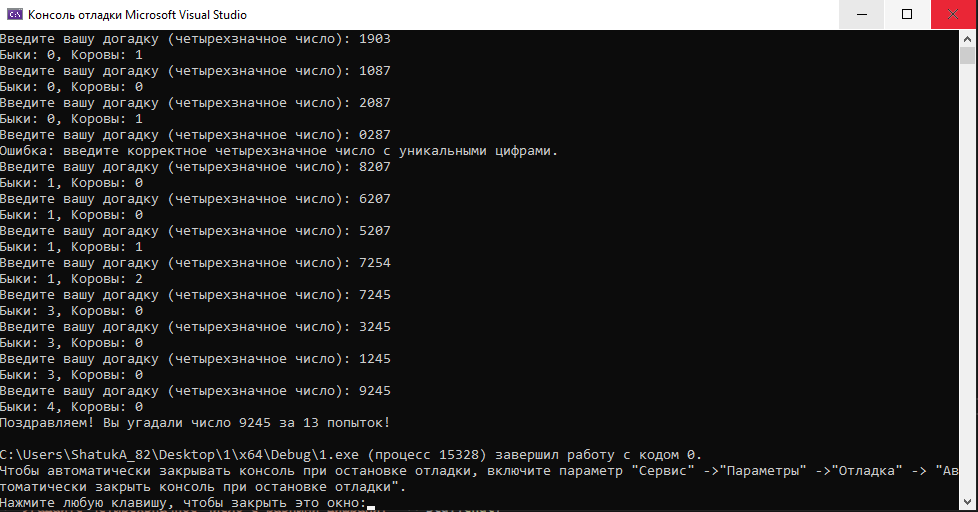
break;

}

}

return 0;

}



Контрольные вопросы:

### 1. Случайное число

Число, выбираемое из диапазона с равной вероятностью.

### 2. Детерминированный алгоритм

Алгоритм, который всегда дает один и тот же результат при одинаковых входных данных.

### 3. Псевдослучайное число

Число, генерируемое детерминированным алгоритмом, но выглядящее случайным.

### 4. Физическое случайное число

Число, полученное на основе непрыыыедсказуемых физических процессов.

### 5. Генератор псевдослучайных чисел

Алгоритм, создающий последовательность чисел с свойствами случайности, но детерминированный.

### 6. Линейный конгруэнтный генератор

Использует формулу Xn+1=(aXn+c)mod  m*Xn*+1​=(*aXn*​+*c*)mod*m* для генерации псевдослучайных чисел.

### 7. Метод Фибоначчи с запаздыванием

Генерирует числа как сумму двух предыдущих с запаздыванием: Xn=(Xn−k+Xn−(k+1))mod  m*Xn*​=(*Xn*−*k*​+*Xn*−(*k*+1)​)mod*m*.