**Федеральное агентство по образованию РФ**

**Государственное образовательное учреждение высшего профессионального образования**

**Белгородский государственный технологический университет им В. Г. Шухова**

Кафедра программного обеспечения вычислительной техники и

автоматизированных систем.

**Расчетно-графическое задание**

Обфускация строк кода на C++

Выполнил (а):

Студент (ка) группы:

КБ-212

Жданов Дмитрий Александрович

Принял:

\_\_\_\_\_\_\_\_\_\_\_\_ Осипов О.В.

***Цель работы:*** написать обфускатор строк кода на C++.

***Выполнение:***

***Код программы:***

#include <iostream>

#include <fstream>

#include <sstream>

#include <map>

#include <vector>

#include <random>

#include <regex>

using namespace std;

// Функция генерации случайного имени переменной заданной длины

string generateRandomName(int length) {

static const char alphanum[] =

"ABCDEFGHIJKLMNOPQRSTUVWXYZ"

"abcdefghijklmnopqrstuvwxyz";

random\_device rd;

mt19937 gen(rd());

uniform\_int\_distribution<> dis(0, sizeof(alphanum) - 2);

stringstream ss;

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

ss << alphanum[dis(gen)];

}

return ss.str();

}

// Функция удаления комментариев из текста

string removeComments(const string& code) {

regex commentRegex("/\\\*([^\*]|(\\\*+[^\*/]))\*\\\*+/|//[^\\n]\*");

return regex\_replace(code, commentRegex, "");

}

// Функция обфускации кода

string obfuscateCode(const string& code) {

map<string, string> variableMap;

string result = removeComments(code);

// Убираем все табуляции, переносы строк и заменяем множественные пробелы на одиночные

result.erase(remove(result.begin(), result.end(), '\t'), result.end());

result.erase(remove(result.begin(), result.end(), '\n'), result.end());

regex spaceRegex("\\s+");

result = regex\_replace(result, spaceRegex, " ");

// Регулярное выражение для поиска текста в кавычках

regex stringRegex("\"(?:\\\\.|[^\"])\*\"");

// Находим все строки в кавычках и временно заменяем их временно

string placeholder = "##STRING##";

vector<string> stringsToPreserve;

sregex\_iterator stringIter(result.begin(), result.end(), stringRegex);

sregex\_iterator stringEnd;

while (stringIter != stringEnd) {

smatch match = \*stringIter;

stringsToPreserve.push\_back(match.str());

result.replace(match.position(), match.length(), placeholder);

++stringIter;

}

// Проводим обфускацию переменных

regex variableRegex("(\\b(?:int|float|double|bool|char|string|auto)\\s+)(\\b(?!main\\b)\\w+)");

sregex\_iterator iter(result.begin(), result.end(), variableRegex);

sregex\_iterator end;

while (iter != end) {

smatch match = \*iter;

if (match.size() == 3) {

string dataType = match[1];

string varName = match[2];

// Проверяем, что переменная не встречается внутри строк

bool isInString = false;

for (const auto& str : stringsToPreserve) {

if (str.find(varName) != string::npos) {

isInString = true;

break;

}

}

if (!isInString) {

string newName = generateRandomName(50);

variableMap[varName] = newName;

regex replaceRegex("\\b" + dataType + varName + "\\b");

result = regex\_replace(result, replaceRegex, dataType + newName);

}

}

++iter;

}

// Восстанавливаем строки обратно в текст

for (size\_t i = 0; i < stringsToPreserve.size(); ++i) {

size\_t pos = result.find(placeholder);

result.replace(pos, placeholder.length(), stringsToPreserve[i]);

}

for (const auto& pair : variableMap) {

regex variableUsageRegex("\\b" + pair.first + "\\b");

result = regex\_replace(result, variableUsageRegex, pair.second);

}

return result;

}

int main() {

// Открываем файл .txt для чтения

ifstream inputFile("input.txt");

if (!inputFile.is\_open()) {

cout << "Невозможно открыть input файл!" << endl;

return 1;

}

// Считываем содержимое файла в строку

stringstream buffer;

buffer << inputFile.rdbuf();

string originalContent = buffer.str();

inputFile.close();

size\_t includePos = originalContent.find("using");

string includes = originalContent.substr(0, includePos);

// Обфусцируем содержимое файла

string obfuscated = obfuscateCode(originalContent.substr(includePos));

string obfuscatedContent = includes + obfuscated;

// Открываем файл .txt для записи обфусцированного содержимого

ofstream outputFile("output.txt");

if (!outputFile.is\_open()) {

cout << "Невозможно открыть output файл!" << endl;

return 1;

}

// Добавляем переносы строк перед #

//regex newLineBeforeHash("#");

//obfuscatedContent = regex\_replace(obfuscatedContent, newLineBeforeHash, "\n#");

// Записываем обфусцированное содержимое в файл

outputFile << obfuscatedContent;

outputFile.close();

cout << "Обфускация выполнена!" << endl;

return 0;

}

***Пример работы программы №1:***

***Исходный код:***

#include <iostream>

#include <math.h>

#define abs(x) ((x)>=0?(x):-(x))

#define step(a, b) (pow(a, b))

#define max(x,y) (x>y?(x):(y))

#define min(x, y) (x<y?(x):(y))

using namespace std;

int main(){

cout << "text with a" << endl;

int start = 0;

cout << endl <<"Enter: "<< endl<< "va1 - 1" << endl << "va2 - 0"<<endl;

cin >> start;

while (start == 1) {

int va1 ;

int va2 ;

cout <<endl << "Enter: " << endl;

cin >> va1;

cout << "Enter: " << endl;

cin >> va2;

int l1;

int l2;

if (5 < va1 < 10) {

l1 = min(va1, va2)\*max(va1 + va2, va1\*va2);

}

if ((va1 <= 5) and (va2 >= 10)) {

l2 = step(max(abs(va1 - va2), va2), 2);

}

cout << endl << "Answer: " << l1 << endl;

cout << "Wanna leave(0) or repeat(1)?" << endl;

cin >> start;

}

return 0;

}

***Вывод программы:***

***Изображение выглядит как текст, снимок экрана, программное обеспечение, дисплей

Автоматически созданное описание***

***Код после работы программы:***

#include <iostream>

#include <math.h>

#define abs(x) ((x)>=0?(x):-(x))

#define step(a, b) (pow(a, b))

#define max(x,y) (x>y?(x):(y))

#define min(x, y) (x<y?(x):(y))

using namespace std; int main(){ cout << "text with a" << endl; int KeFRKNJREkNLpIFZnaflDMKrvHxwKBoCcTNUmuyjkhEVggdygZ = 0; cout << endl <<"Enter: "<< endl<< "var1 - 1" << endl << "var2 - 0"<<endl; cin >> KeFRKNJREkNLpIFZnaflDMKrvHxwKBoCcTNUmuyjkhEVggdygZ; while (KeFRKNJREkNLpIFZnaflDMKrvHxwKBoCcTNUmuyjkhEVggdygZ == 1) { int iIIwikMpujLTwvfzxhwziviFsZKPiwAmHGaIHkugWkWjfxWWFb ; int QOVgVTFKOHefGQsELAMXvnYfphqidmcJTZbPLklegrdnnolLzi ; cout <<endl << "Enter: " << endl; cin >> rCveFGapZooIALWuWtFhOOqMoTXZkiiNzIxzuNxmTHkMchZHZV; cout << "Enter: " << endl; cin >> QOVgVTFKOHefGQsELAMXvnYfphqidmcJTZbPLklegrdnnolLzi; int NDnjchaKIsEvjiuoGfFcbAWtbqUBpeUiMftaFtvoFulcUNVHdC; int ZzUrkCJZtZSdbMuFDHOKduwKGCeQypbkfcAbWLVzhwJLMabnzJ; if (5 < rCveFGapZooIALWuWtFhOOqMoTXZkiiNzIxzuNxmTHkMchZHZV < 10) { NDnjchaKIsEvjiuoGfFcbAWtbqUBpeUiMftaFtvoFulcUNVHdC = min(rCveFGapZooIALWuWtFhOOqMoTXZkiiNzIxzuNxmTHkMchZHZV, QOVgVTFKOHefGQsELAMXvnYfphqidmcJTZbPLklegrdnnolLzi)\*max(rCveFGapZooIALWuWtFhOOqMoTXZkiiNzIxzuNxmTHkMchZHZV + QOVgVTFKOHefGQsELAMXvnYfphqidmcJTZbPLklegrdnnolLzi, rCveFGapZooIALWuWtFhOOqMoTXZkiiNzIxzuNxmTHkMchZHZV\*QOVgVTFKOHefGQsELAMXvnYfphqidmcJTZbPLklegrdnnolLzi); } if ((rCveFGapZooIALWuWtFhOOqMoTXZkiiNzIxzuNxmTHkMchZHZV <= 5) and (QOVgVTFKOHefGQsELAMXvnYfphqidmcJTZbPLklegrdnnolLzi >= 10)) { FXCmFEXHYyzTZjpYZczabsSSIGqiPLOQrSYbBbzGPmjdcrzxxo = step(max(abs(rCveFGapZooIALWuWtFhOOqMoTXZkiiNzIxzuNxmTHkMchZHZV - QOVgVTFKOHefGQsELAMXvnYfphqidmcJTZbPLklegrdnnolLzi), QOVgVTFKOHefGQsELAMXvnYfphqidmcJTZbPLklegrdnnolLzi), 2); } cout << endl << "Answer: " << NDnjchaKIsEvjiuoGfFcbAWtbqUBpeUiMftaFtvoFulcUNVHdC << endl; cout << "Wanna leave(0) or repeat(1)?" << endl; cin >> KeFRKNJREkNLpIFZnaflDMKrvHxwKBoCcTNUmuyjkhEVggdygZ; } return 0; }

***Вывод программы:***

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

***Пример работы программы №2:***

***Исходный код:***

#include <iostream>

#include <cstdlib>

#include <ctime>

#include <string>

#include <chrono>

using namespace std;

using namespace std::chrono;

// Функция для генерации случайного имени

string generateRandomName() {

string names[] = {"Александр", "Екатерина", "Михаил", "Анна", "Иван", "Ольга", "Дмитрий", "Наталья"};

return names[rand() % (sizeof(names) / sizeof(names[0]))];

}

// Функция для вычисления примера

double calculateExpression(double a, double b) {

return (a \* a) + (b / 2) - (a \* b);

}

int main() {

// Начало отсчета времени выполнения программы

auto start = high\_resolution\_clock::now();

// Настройка генератора случайных чисел

srand(static\_cast<unsigned int>(time(nullptr)));

cout << "Случайные имена, возрасты и результат сложного примера:\n\n";

// Генерация и вывод 10 случайных имен, возрастов и результатов сложного примера

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

string name = generateRandomName();

int age = rand() % 50 + 18; // Генерация возраста от 18 до 67 лет

double a = rand() % 10 + 1; // Генерация числа a от 1 до 10

double b = rand() % 10 + 1; // Генерация числа b от 1 до 10

cout << "Имя: " << name << ", Возраст: " << age << " лет, ";

// Вычисление и вывод результата сложного примера

double result = calculateExpression(a, b);

cout << "Результат выражения (а^2 + ь/2 - ь\*а) при а=" << a << " и ь=" << b << ": " << result << endl;

}

// Завершение отсчета времени выполнения и вывод времени работы программы

auto stop = high\_resolution\_clock::now();

auto duration = duration\_cast<milliseconds>(stop - start);

cout << "\nВремя выполнения программы: " << duration.count() << " миллисекунд\n";

// Вывод текущего времени

time\_t currentTime = system\_clock::to\_time\_t(system\_clock::now());

cout << "Текущее время: " << ctime(&currentTime);

return 0;

}

***Вывод программы:***

Изображение выглядит как текст, электроника, снимок экрана, программное обеспечение

Автоматически созданное описание

***Код после работы программы:***

#include <iostream>

#include <cstdlib>

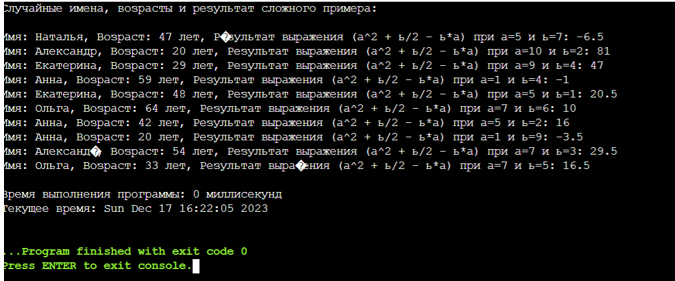
#include <ctime>

#include <string>

#include <chrono>

using namespace std; using namespace std::chrono; string dDEIZSVybKUwJKlZLWnAYchFzioNYPMuAQGEtBUiRaPmzIwcNG () { string NXEcyWlSRXMGAbboLfdVoADxsvhmwhVNAYRzyaqmwHcMelwPxY [] = {"Александр", "Екатерина", "Михаил", "Анна", "Иван", "Ольга", "Дмитрий", "Наталья"}; return NXEcyWlSRXMGAbboLfdVoADxsvhmwhVNAYRzyaqmwHcMelwPxY [rand() % (sizeof(NXEcyWlSRXMGAbboLfdVoADxsvhmwhVNAYRzyaqmwHcMelwPxY) / sizeof(NXEcyWlSRXMGAbboLfdVoADxsvhmwhVNAYRzyaqmwHcMelwPxY [0]))]; } double ZDcYqKQtUUyClOkOtdCaZBZCWIOrVUcFwPCMQAVegBUVqxKdUW(double EScUwsZIrmbAIbVztVmnJhGodwWDuqjzgccCnzbeeOKvkHmPlq, double NXEcyWlSRXMGAbboLfdVoADxsvhmwhVNAYRzyaqmwHcMelwPxY) { return (EScUwsZIrmbAIbVztVmnJhGodwWDuqjzgccCnzbeeOKvkHmPlq \* EScUwsZIrmbAIbVztVmnJhGodwWDuqjzgccCnzbeeOKvkHmPlq) + (dDEIZSVybKUwJKlZLWnAYchFzioNYPMuAQGEtBUiRaPmzIwcNG / 2) - (EScUwsZIrmbAIbVztVmnJhGodwWDuqjzgccCnzbeeOKvkHmPlq \* dDEIZSVybKUwJKlZLWnAYchFzioNYPMuAQGEtBUiRaPmzIwcNG); } int main() { auto AScUwsZIrmbAIbVztVmasdhGodwWDuqjzgccCnzbeeOKvkHmRl = high\_resolution\_clock::now(); srand(static\_cast<unsigned int>(time(nullptr))); cout << "Случайные имена, возрасты и результат сложного примера:\n\n"; for (int AScUwsZIrmbAIbVztVmasdhGodwWDuqjzgccCnzbeeOKvkHmRl = 0; AScUwsZIrmbAIbVztVmasdhGodwWDuqjzgccCnzbeeOKvkHmRl < 10; ++ AScUwsZIrmbAIbVztVmasdhGodwWDuqjzgccCnzbeeOKvkHmRl) { string dDEIZSVybKUwJKlZLWnAYchFzioNYPMuAQGEtBUiRaPmzIwcNG = generateRandomName(); int EScUwsZIrmbAIbVztVmnJhGodwWDuqjzgccCnzbeeOKvkHmPlq = rand() % 50 + 18; double EScUwsZIrmbAIbVztVmnJhGodwWDuqjzgccCnzbeeOKvkHmPlq = rand() % 10 + 1; double EScUwsZIrmbAIbVztVmnJhGodwWDuqjzgccCnzbeeOKvkHmPlq = rand() % 10 + 1; cout << "Имя: " << EScUwsZIrmbAIbVztVmnJhGodwWDuqjzgccCnzbeeOKvkHmPlq << ", Возраст: " << EScUwsZIrmbAIbVztVmnJhGodwWDuqjzgccCnzbeeOKvkHmPlq << " лет, "; double NXEcyWlSRXMGAbboLfdVoADxsvhmwhVNAYRzyaqmwHcMelwPxY = ZDcYqKQtUUyClOkOtdCaZBZCWIOrVUcFwPCMQAVegBUVqxKdUW(EScUwsZIrmbAIbVztVmnJhGodwWDuqjzgccCnzbeeOKvkHmPlq, b); cout << "Результат выражения (а^2 + b/2 - b\*а) при а=" << EScUwsZIrmbAIbVztVmnJhGodwWDuqjzgccCnzbeeOKvkHmPlq << " и b=" << NXEcyWlSRXMGAbboLfdVoADxsvhmwhVNAYRzyaqmwHcMelwPxY << ": " << EScUwsZIrmbAIbVztVmnJhGodwWDuqjzgccCnzbeeOKvkHmPlq << endl; } auto dDEIZSVybKUwJKlZLWnAYchFzioNYPMuAQGEtBUiRaPmzIwcNG = high\_resolution\_clock::now(); auto pGuIXlhXzRDDbZwjkjluYwCwZmItAHBvAddXqsleFhoQsEwBzi = duration\_cast<milliseconds>( dDEIZSVybKUwJKlZLWnAYchFzioNYPMuAQGEtBUiRaPmzIwcNG - pGuIXlhXzRDDbZwjkjluYwCwZmItAHBvAddXqsleFhoQsEwBzi); cout << "\nВремя выполнения программы: " << NXEcyWlSRXMGAbboLfdVoADxsvhmwhVNAYRzyaqmwHcMelwPxY.count() << " миллисекунд\n"; time\_t dDEIZSVybKUwJKlZLWnAYchFzioNYPMuAQGEtBUiRaPmzIwcNG = system\_clock::to\_time\_t(system\_clock::now()); cout << "Текущее время: " << ctime(&currentTime); return 0; }

***Вывод программы:***



***Вывод:*** в ходе выполнения расчетно-графического задания был написан обфускатор строк кода на C++.