КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ІМЕНІ ТАРАСА ШЕВЧЕНКА

Факультет інформаційних технологій

Кафедра програмних систем і технологій

Захист програм та даних

Практична робота № 2.4

Виконали студенти

групи ІПЗ-33

Щука Богдан

Ємець Євгенія

Олійник Анастасія

Перевірила

доцент Супрун О.М.

Київ-2018

**Хід роботи**

**Завдання 1**

В практической работе необходимо определить контрольные данные с использованием следующих способов:

* битов четности. В качестве исходных данных принять битовое представление букв фамилии в соответствии с кодировкой Windows 1251
* контрольных цифр. В качестве исходных данных принять необходимое количество цифр (за исключением контрольной) из строки, состоящей из кодов букв фамилии, имени и отчества согласно их положению в алфавите
* по алгоритму Луна (15 цифр);
* для штрихкода по стандарту ЕАН-13 (12 цифр);
* для ИНН физического лица (10 цифр);
* для кодов станций на железнодорожном транспорте (5 цифр);
* контрольных сумм (CRC). В качестве исходных данных принять коды 1-ой, 2-ой и 3-ей букв своей фамилии согласно их положению в алфавите; порождающего полинома -
* кода коррекции ошибок (ЕСС). В качестве исходных данных принять первые 11 битов первых двух букв своей фамилии в соответствии с кодировкой Windows 1251 (табл. 3.2). Рассчитать вектор контрольных битов и вектора синдромов при отсутствии ошибки, одиночной и двойной ошибке.

Лістинг програми

**public class** Main {  
 **public static void** main(String[] args) {  
 String alphabet = **"ABCDEFGHIJKLMNOPQRSTUWXYZ"**;  
 String lastName = **"YEMETS"**;  
 String full = **"YEMETSSCHUKA"**;  
 **int**[] positions = **new int**[full.length()];  
 **for** (**int** i = 0; i < positions.**length**; i++){  
 positions[i] = alphabet.indexOf(full.charAt(i)) + 1;  
 }  
 *Bits*(lastName);  
 *Lun*(*MakeArray*(positions, 14));  
 *EAN13*(*MakeArray*(positions, 13));  
 *INN*(*MakeArray*(positions, 10));  
 *RailwayTransport*(*MakeArray*(positions, 5));  
 *CRC*(**new int**[] { alphabet.indexOf(lastName.charAt(0)) + 1, alphabet.indexOf(lastName.charAt(1))+1,alphabet.indexOf(lastName.charAt(2))+1});  
  
 String message = lastName.substring(0, 2);  
 **byte**[] messageInASCII = message.getBytes();String newMessage = **""**;  
  
 **for** (**int** i = 0; i < messageInASCII.**length**; i++){  
 newMessage+=Integer.*toBinaryString*(Integer.*valueOf*(messageInASCII[i])); }  
 newMessage = newMessage.substring(0, 11);  
  
 *ECC*(newMessage);  
  
 }  
  
  
 **private static void** ECC(String newMessage)  
 {  
 System.***out***.println(**"Алгоритм ECC. Входные данные:"**);  
 System.***out***.println(newMessage);  
  
 String result = **""**;  
  
 result = **"11"** + newMessage.charAt(0) + **"0"** + newMessage.substring(1, 3) + **"0"** + newMessage.substring(4, newMessage.length() - 4);  
  
 **int** pb = 0;  
  
 **for** (**int** i = 0; i < result.length(); i++)  
 {  
 pb += Integer.*valueOf*(String.*valueOf*(result.charAt(i)));  
 }  
  
 pb %= 2;  
 System.***out***.println(result + **", pb = "** + pb);  
 }  
  
 **private static void** CRC(**int**[] v)  
 {  
 System.***out***.println(**"G(x) = x^4 + x^1 + x^0"**);  
 System.***out***.println(**"Алгоритм CRC. Входные данные:"**);  
  
 **for** (**int** item : v)  
 {  
 System.***out***.print(item + **"\t"**);  
 }  
 System.***out***.println();  
  
 String dect = **"1011"**;  
 System.***out***.println(**"dect = "** + dect + **"("** + Integer.*toBinaryString*(1011) + **")"**);*//dect.toString(2); Integer. Convert.ToInt32(dect, 2) + ")");* System.***out***.println();  
  
 System.***out***.println(**"Делимое \t Формула \t Частное \t Остаток \t Результат"**);  
  
  
 **for** (**int** i = 0; i < v.**length**; i++)  
 {  
 String input = Integer.*toBinaryString*(v[i]);  
  
 String input2 = input.concat(**"0000"**);  
  
 String chastnoe = *GetChastnoeAndOstatokForCRC*(input2, dect)[0];  
 String ostatok = *GetChastnoeAndOstatokForCRC*(input2, dect)[1];  
  
 **int** input2Integer = Integer.*valueOf*(input2);  
  
 **int** ostatokInteger = Integer.*valueOf*(ostatok, 2);  
  
 **int** chastnoeInteger = Integer.*valueOf*(chastnoe, 2);  
  
 String result = input.concat(ostatok);  
  
 **int** finalRes = Integer.*valueOf*(result, 2);  
  
 System.***out***.println((v[i] + **"("** + input + **")"** + **"\t"** +  
 input2 + **"("** + input2Integer + **")"** + **"\t"** +  
 chastnoe + **"("** + chastnoeInteger + **")"** + **"\t"** + ostatok + **"("** + ostatokInteger + **")"** + **"\t"** + result + **"("** + finalRes + **")"**));  
 }  
 }  
  
 **private static** String[] GetChastnoeAndOstatokForCRC(String imya2, String dect)  
 {  
 String delit, resultat = **""**;  
 **boolean** f = **true**;  
 StringBuilder sb;  
 **do** {  
 delit = **""**;  
 **int** l1 = imya2.length() - 1;  
 **int** l2 = dect.length() - 1;  
 **int** raz = l1 - l2;  
  
 String dect2 = dect;  
 **for** (**int** i = 0; i < raz; i++)  
 {  
 dect2 = dect2 + **"0"**;  
 }  
  
 **for** (**int** i = 0; i <= l1; i++)  
 {  
 delit = delit + String.*valueOf*(Integer.*valueOf*(Integer.*toBinaryString*(imya2.charAt(i))) ^ Integer.*valueOf*(Integer.*toBinaryString*(dect2.charAt(i))));  
 }  
  
 **int** counter = 0;  
  
 **boolean** h = **true**;  
 **if** (delit.indexOf(**'1'**) >= 0)  
 {  
 **do** {  
 **if** (delit.charAt(1) != **'0'**)  
 h = **false**;  
 **if** (delit.charAt(0) == **'0'** && delit.length() >= dect.length())  
 {  
 delit = delit.substring(1, delit.length() - 1);  
 counter++;  
 }  
 **else** {  
 h = **false**;  
 }  
 }  
 **while** (h);  
 }  
 **else** {  
 imya2 = **"0000"**;  
 sb = **new** StringBuilder(**"1"**);  
 **for** (**int** i = 0; i < dect.length() - 1; i++) {  
 sb.append(**'0'**);  
 }  
 resultat = sb.toString();  
 f = **false**;  
 **break**;  
 }  
 sb = **new** StringBuilder(**"1"**);  
 **for** (**int** i = 0; i < counter - 1; i++) {  
 sb.append(**'0'**);  
 }  
 resultat += sb.toString();  
  
  
 **if** (delit.length() < dect.length())  
 f = **false**;  
  
 imya2 = delit;  
 }  
 **while** (f);  
  
 **return new** String[] { resultat, imya2 };  
 }  
  
 **private static void** RailwayTransport(**int**[] v)  
 {  
 System.***out***.println(**"Алгоритм для кодов станций на ж/д транспорте. Входные данные:"**);  
  
 **for** (**int** item : v)  
 {  
 System.***out***.print(item + **"\t"**);  
 }  
  
 System.***out***.println();  
  
 **int** controlNumber = (1 \* v[0] + 2 \* v[1] + 3 \* v[2] + 4 \* v[3]) % 11;  
  
 **if** (controlNumber == 10)  
 {  
 controlNumber = (3 \* v[0] + 4 \* v[1] + 5 \* v[2] + 6 \* v[3]) % 11;  
  
 **if** (controlNumber == 10)  
 {  
 controlNumber = 0;  
 }  
 }  
  
 System.***out***.println(**"Контрольная цифра должна равняться "** + controlNumber);  
 }  
  
 **private static void** INN(**int**[] v)  
 {  
 System.***out***.println(**"Алгоритм ИНН. Входные данные:"**);  
  
 **for** (**int** item : v)  
 {  
 System.***out***.print(item + **"\t"**);  
 }  
  
 System.***out***.println();  
  
 **int** result = ((2 \* v[0] + 4 \* v[1] + 10 \* v[2] + 3 \* v[3] + 5 \* v[4] + 9 \* v[5] + 5 \* v[6] + 6 \* v[7] + 8 \* v[8]) % 11) % 10;  
  
 System.***out***.println(**"Контрольная цифра должна равняться "** + result);  
 }  
  
 **private static void** EAN13(**int**[] v)  
 {  
 System.***out***.println(**"Алгоритм EAN-13. Входные данные:"**);  
  
 **for** (**int** item : v)  
 {  
 System.***out***.print(item + **"\t"**);  
 }  
  
 System.***out***.println();  
  
 **int** sumNeChet = 0;  
 **int** sumChet = 0;  
  
 **for** (**int** i = 1; i <= v.**length**; i++)  
 {  
 **if** (i % 2 == 0)  
 {  
 sumChet += v[i - 1];  
 }  
 **else if** (i != v.**length**)  
 {  
 sumNeChet += v[i - 1];  
 }  
 }  
  
 sumChet \*= 3;  
  
 **int** result = 0;  
  
 **for** (**int** i = 0; ; i++)  
 {  
 **if** ((sumChet + sumNeChet + i) % 10 == 0)  
 {  
 result = i;  
 **break**;  
 }  
 }  
  
  
 System.***out***.println(**"Контрольная цифра должна равняться "** + result);  
 }  
  
 **private static void** Lun(**int**[] v)  
 {  
 System.***out***.println(**"Алгоритм Луна. Входные данные:"**);  
  
 **for** (**int** item : v)  
 {  
 System.***out***.print(item + **"\t"**);  
 }  
  
 System.***out***.println();  
  
 **int** sumNeChet = 0;  
 **int** sumChet = 0;  
  
 **for** (**int** i = 1; i <= v.**length**; i++)  
 {  
 **if** (i % 2 == 0)  
 {  
 v[i - 1] = (v[i - 1] \* 2) % 9;  
  
 sumChet += v[i - 1];  
 }  
 **else** {  
 sumNeChet += v[i - 1];  
 }  
 }  
  
 **int** result = 0;  
  
 **for** (**int** i = 0; ; i++)  
 {  
 **if** ((sumChet + sumNeChet + i) % 10 == 0)  
 {  
 result = i;  
 **break**;  
 }  
 }  
  
  
 System.***out***.println(**"Контрольная цифра должна равняться "** + result);  
 }  
  
 **private static void** Bits(String lastName)  
 {  
 System.***out***.println(**"Биты четности"**);  
 System.***out***.println(**"Буква \t Битовая строка \t Четный бит \t Нечетный бит"**);  
 **for** (**char** item : lastName.toCharArray())  
 {  
 **byte**[] messageInASCII = String.*valueOf*(item).getBytes();*//new char[] { item }).;Encoding.ASCII.GetBytes* String newMessage = **""**;  
  
 **for** (**int** i = 0; i < messageInASCII.**length**; i++)  
 {  
 newMessage += Integer.*toBinaryString*(Integer.*valueOf*(messageInASCII[i]));  
 }  
  
 **int** EvenBit = (newMessage.length() - newMessage.replace(**"1"**, **""**).length()) % 2;  
  
 **int** OddBit = EvenBit == 1 ? 0 : 1;  
  
 System.***out***.println(String.*format*(**"%s \t %s \t %d \t %d"**, item, newMessage, EvenBit, OddBit));  
 }  
  
 }  
  
 **private static int**[] MakeArray(**int**[] array, **int** count)  
 {  
 **int**[] result = **new int**[count];  
  
 **for** (**int** i = 0; i < count; i++)  
 {  
 result[i] = array[i];  
 }  
  
 **return** result;  
 }  
}

Приклад роботи програми

