МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ

УНИВЕРСИТЕТ им. Р.Е.АЛЕКСЕЕВА

Институт радиоэлектроники и информационных технологий

Кафедра информатики и систем управления

ОТЧЁТ

по лабораторной работе 2

по дисциплине

Защита информации

РУКОВОДИТЕЛЬ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись) (фамилия, и.,о.)

СТУДЕНТ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Симанов В.С.

(подпись) (фамилия, и.,о.)

21-ИС\_\_\_\_\_\_\_\_

(шифр группы)

Работа защищена «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_

С оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Нижний Новгород

2024

Цель: шифрование методом гаммирования

Задания:

Задание № 2

Реализовать алгоритм шифрования данных «Шифрование методом гаммирования» по модулю N.

«Шифрование методом гаммирования»:

Гаммирование – метод последовательного симметричного шифрования, суть которого состоит в том, что символы шифруемого текста последовательно складываются с символами некоторой специальной последовательности, которая называется гаммой.

Перед шифрованием символы сообщения и гаммы заменяются их номерами в алфавите и само кодирование выполняется по формуле

***Ci = (Ti+Gi) mod N***

Примечания:

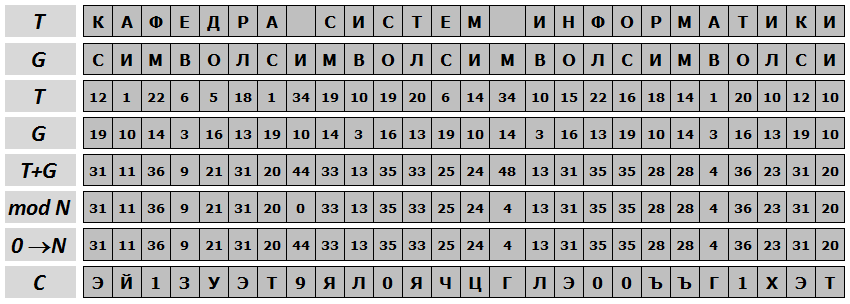
а) *mod* - операция целочисленного деления, вычисляющая остаток от деления. Например, *18 mod 5 = 3*

или *48 mod 44 = 4*. Данная операция доступна в Windows-калькуляторе в режиме "Инженерный".

б) *N* равен количеству символов применяемого алфавита.

в) *Ci*, *Ti* и *Gi* - номера *i*-х символов, соответственно, шифрограммы, шифруемого текста и гаммы

г) если *Ci* будет равно нулю, то его следует приравнять *N*.



В данной теме используется алфавит, состоящий из 44 символов (N=44).

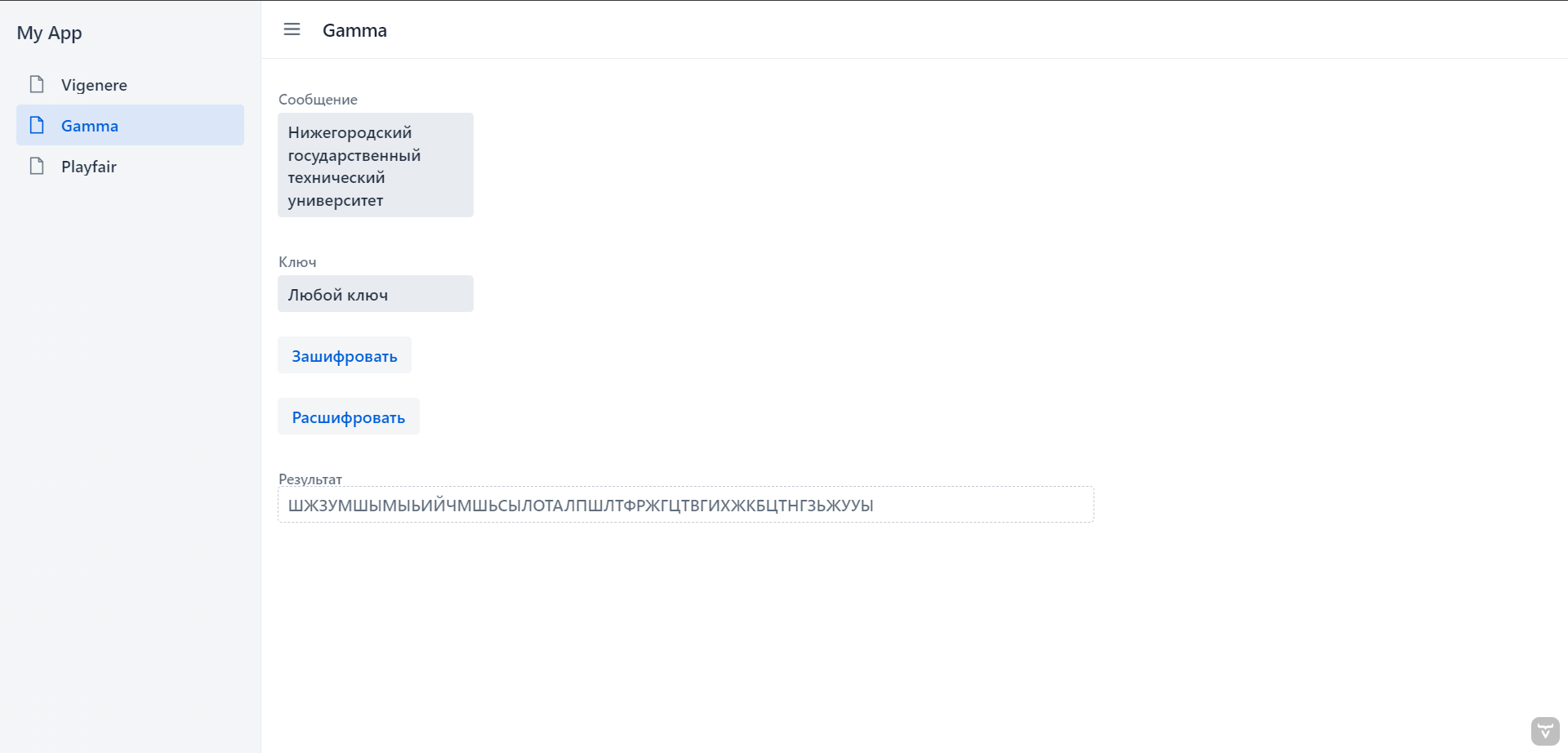
Примечание: Если в строке T+G значение меньше чем N, то значение в строке mod N должно быть таким же как в T+G, противном случае значение строки T+G убавляйте на N.

Примеры шифрования и расшифровывания:

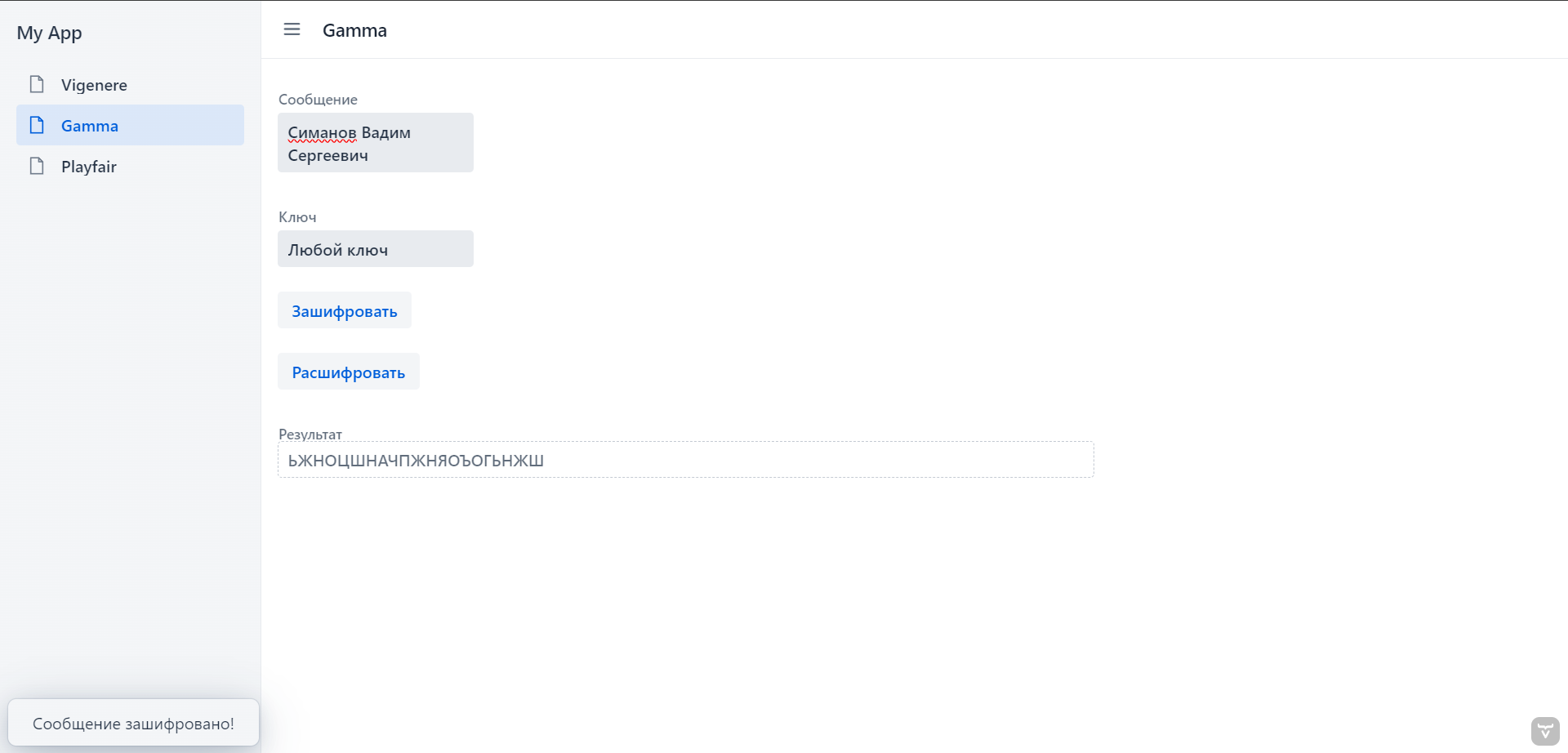
Нижегородский государственный технический университет

Шифрование:

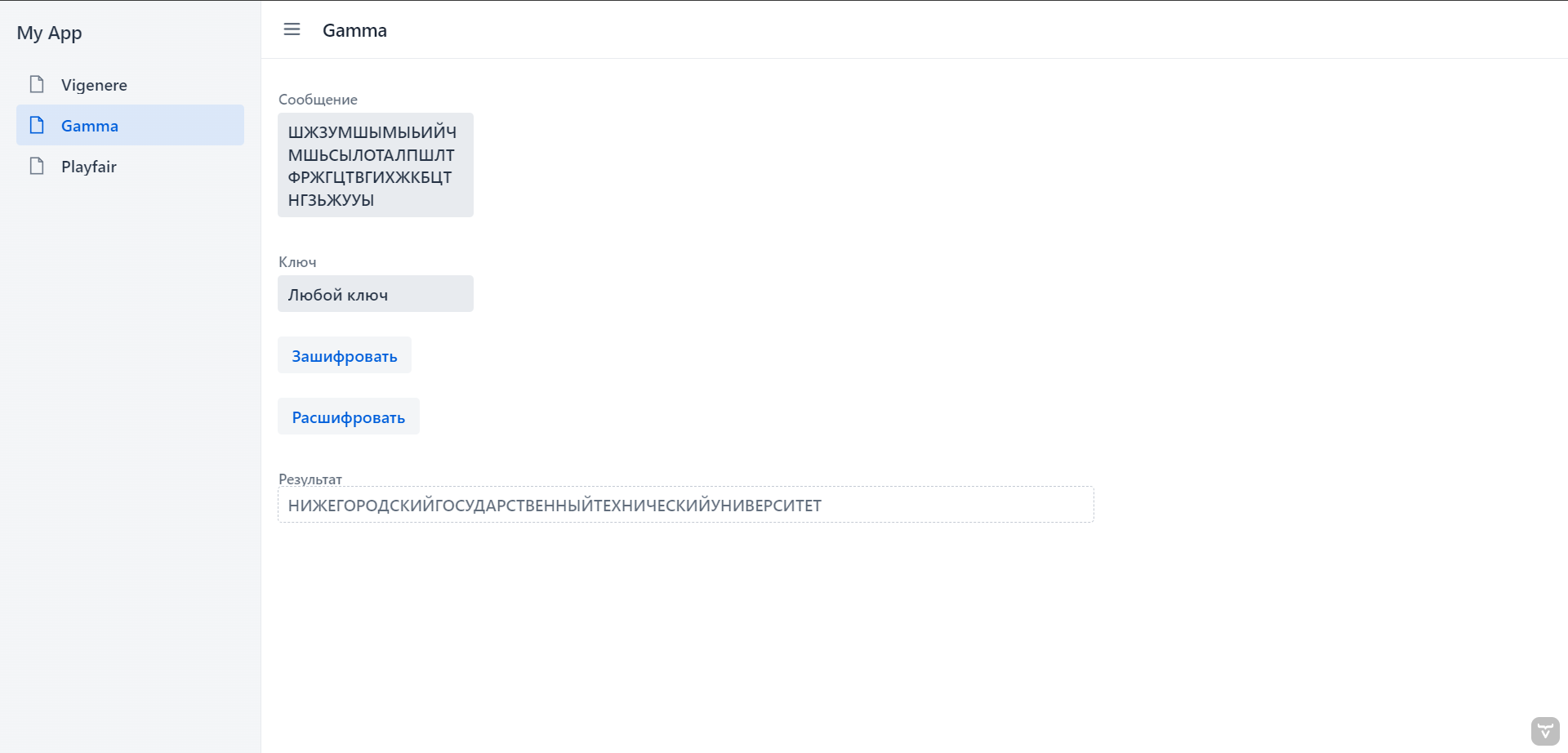
Нижегородский государственный технический университет:

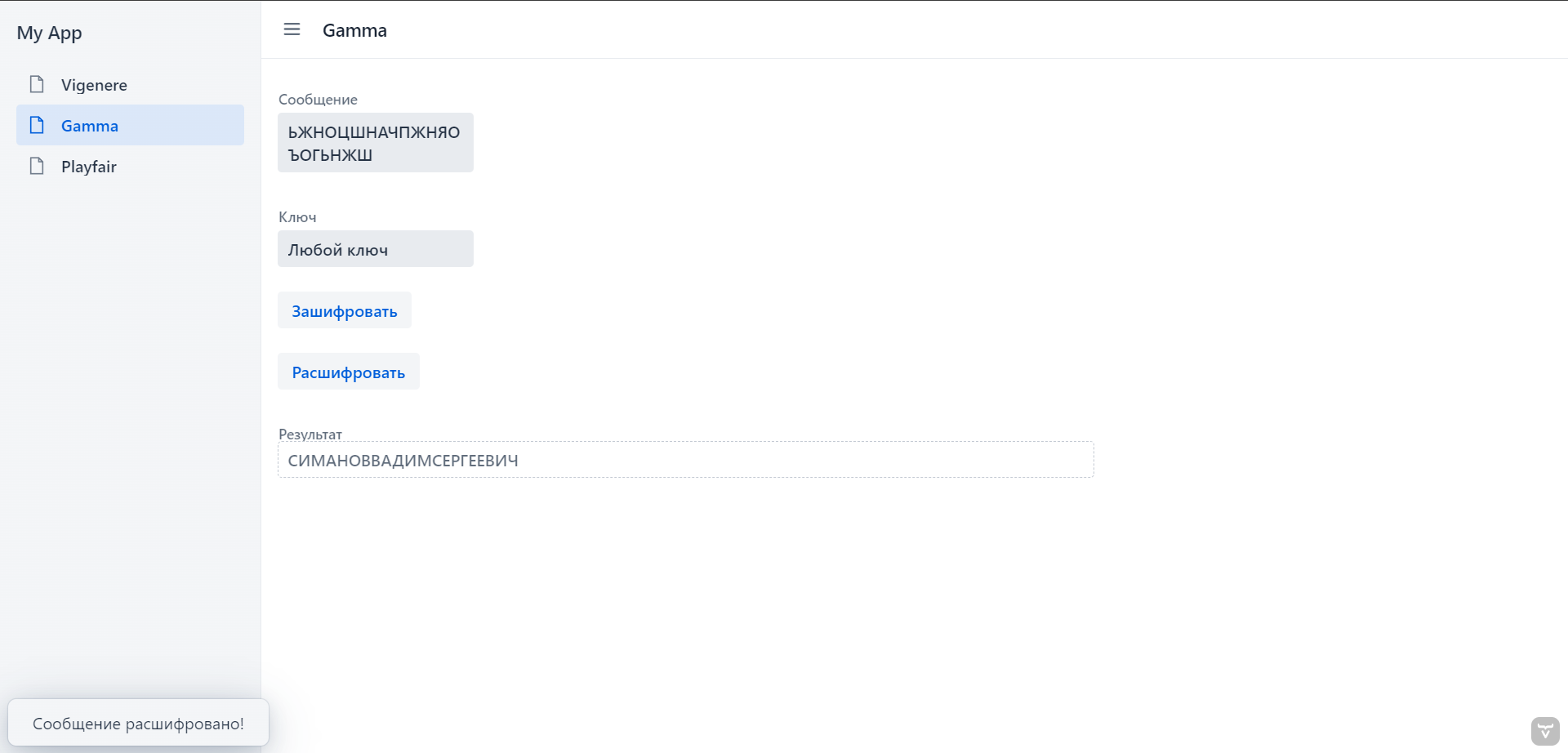


Симанов Вадим Сергеевич:



Расшифрование:





Текст программы с комментариями:

GammaView – отвечает за отображении страницы

package com.example.application.views.lb;  
  
import com.vaadin.flow.component.button.Button;  
import com.vaadin.flow.component.notification.Notification;  
import com.vaadin.flow.component.orderedlayout.VerticalLayout;  
import com.vaadin.flow.component.textfield.TextArea;  
import com.vaadin.flow.component.textfield.TextField;  
import com.vaadin.flow.router.Menu;  
import com.vaadin.flow.router.PageTitle;  
import com.vaadin.flow.router.Route;  
  
@PageTitle("Gamma")  
// Добавляет элемент в меню с иконкой и указывает порядок отображения  
@Menu(icon = "line-awesome/svg/file.svg", order = 0)  
// Определяет маршрут для отображения этой страницы в браузере  
@Route(value = "Gamma")  
public class GammaView extends VerticalLayout {  
  
 public GammaView() {  
 // Поля для ввода  
 TextArea textArea = new TextArea("Сообщение");  
 TextField Key = new TextField("Ключ");  
 TextField result = new TextField("Результат");  
  
 result.setWidth("800px");  
 result.setHeight("50px");  
 result.setReadOnly(true);  
  
 cipherGamma cipher = new cipherGamma();  
  
 // Кнопка для шифрования  
 Button button = new Button("Зашифровать", e -> {  
 String text = textArea.getValue();  
 String key = Key.getValue();  
 if (text.isEmpty() || key.isEmpty()) {  
 Notification.*show*("Пожалуйста, введите текст и ключ.");  
 } else {  
  
 String encryptedText = cipher.encryptKey(text, key);  
 result.setValue(encryptedText);  
 Notification.*show*("Сообщение зашифровано!");  
 }  
 });  
  
 // Кнопка для расшифрования  
 Button button1 = new Button("Расшифровать", e -> {  
 String text = textArea.getValue();  
 String key = Key.getValue();  
 if (text.isEmpty() || key.isEmpty()) {  
 Notification.*show*("Пожалуйста, введите текст и ключ.");  
 } else {  
 String decryptedText = cipher.decryptKey(text, key);  
 result.setValue(decryptedText);  
 Notification.*show*("Сообщение расшифровано!");  
 }  
 });  
  
 // Добавляем компоненты на страницу  
 add(textArea, Key, button, button1, result);  
 }  
  
}

cipherGamma – метод реализующий алгоритм шифрования данных «Шифрование методом гаммирования».

package com.example.application.views.lb;  
  
public class cipherGamma {  
  
 private static final String *alphabet* = "АБВГДЕЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ"; // Русский алфавит без Ё  
  
 Integer modN=*alphabet*.length();  
  
 // Метод шифрования  
 public String encryptKey(String text, String key) {  
 StringBuilder result = new StringBuilder();  
  
 text = *prepareText*(text);  
 key = *prepareKey*(key, text.length());  
  
 for (int i = 0; i < text.length(); i++) {  
 int textCharIndex = *alphabet*.indexOf(text.charAt(i)); // Индекс символа текста  
 int кeyCharIndex = *alphabet*.indexOf(key.charAt(i)); // Индекс символа ключа  
  
 int encryptedIndex = (textCharIndex + кeyCharIndex) % modN; // Шифруем по модулю N  
 result.append(*alphabet*.charAt(encryptedIndex)); // Получаем символ из алфавита  
 }  
  
 return result.toString();  
 }  
  
 // Метод расшифрования  
 public String decryptKey(String text, String key) {  
 StringBuilder result = new StringBuilder();  
  
 text = *prepareText*(text);  
 key = *prepareKey*(key, text.length());  
  
 for (int i = 0; i < text.length(); i++) {  
 int textCharIndex = *alphabet*.indexOf(text.charAt(i)); // Индекс символа текста  
 int gammaCharIndex = *alphabet*.indexOf(key.charAt(i)); // Индекс символа гаммы  
  
 int decryptedIndex = (textCharIndex - gammaCharIndex + modN) % modN; // Расшифровка по модулю N  
 result.append(*alphabet*.charAt(decryptedIndex)); // Получаем символ из алфавита  
 }  
  
 return result.toString();  
 }  
  
 // Подготовка текста: переводим в верхний регистр и убираем все символы, кроме букв алфавита  
 private static String prepareText(String text) {  
 text = text.toUpperCase().replaceAll("[^А-Я]", "").replace("Ё", "Е");  
 return text;  
 }  
  
 // Подготовка гаммы: дублируем ключ, чтобы длина гаммы совпадала с длиной текста  
 private static String prepareKey(String key, int textLength) {  
 key = key.toUpperCase().replaceAll("[^А-Я]", "").replace("Ё", "Е");  
 StringBuilder extendedKey = new StringBuilder(key);  
 while (extendedKey.length() < textLength) {  
 extendedKey.append(key); // Повторяем ключ до необходимой длины  
 }  
 return extendedKey.substring(0, textLength); // Обрезаем до нужной длины  
 }  
  
}

Application – запуск программы

package com.example.application;  
  
import com.vaadin.flow.component.page.AppShellConfigurator;  
import com.vaadin.flow.theme.Theme;  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
  
@SpringBootApplication  
@Theme(value = "my-app")  
public class Application implements AppShellConfigurator {  
  
 public static void main(String[] args) {  
 SpringApplication.*run*(Application.class, args);  
 }  
  
}

Вывод: мною был реализован алгоритм шифрования данных - «Шифрование методом гаммирования» по модулю N