Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Институт Информационных Технологий и Анализа Данных

Кафедра вычислительной техники

**Название работы** – “Кодировки текста”

Отчет по лабораторной работе “Лабораторная работа №3”

по дисциплине Операционные системы

Выполнил

Студент, номер группы ИСМб-19-1 \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Д.Д.Солопов

Дата: \_\_\_\_\_\_\_\_\_\_

Принял

Должность \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Д.В.Пестерев

Дата: \_\_\_\_\_\_\_\_\_\_

Иркутск 2021 г.

Содержание

[Постановка задач 3](#_Toc64885293)

[Решение задач 4](#_Toc64885294)

[Задание 1 4](#_Toc64885295)

[Задание 2 6](#_Toc64885296)

Постановка задач

Лабораторная работа №3. Кодировки текста.

1. Выберите любую пару кодировок текста

2. Напишите файл сценария Windows Script Host, который считывает содержимое текстового файла в одной кодировке и записывает содержимое в другой кодировке в новый текстовый файл. Написать обратный скрипт. Для перевода возможно использование вставок функций на языке javascript (Учебник по JavaScript в папке с заданием).

Вставка скрипта на другом языке в файл сценария описана в книге из лабораторной №2, страница 275.

3. Написать html-приложение (\*.hta), которое переводит текст из одной кодировки в другую и наоборот. Для ввода текста в исходной кодировке использовать поля ввода. Для перевода текста также возможно использование функций на языке javascript. Так как \*hta приложения построены на синтаксисе языка разметки HTML, то встраивание скрипта такое же, как для обычной HTML страницы, описано в учебнике по Javascript, страница 40.

4. Написать приложение на любом выбранном языке программирования, которое также переводит текст из одной кодировки в другую. Для ввода текста использовать поля ввода.

5. Напишите приложение для выполнения под ОС Android, которое переводит текст в одной кодировке (например win-1251) и записывает содержимое в другой кодировке (например koi-8) и обратную задачу.

Приложение должно содержать текстовые поля для ввода, а также иметь возможность загружать текстовый файл с текстом в исходной кодировке и сохранять в текстовый файл переведенный текст на ваше мобильное устройство.

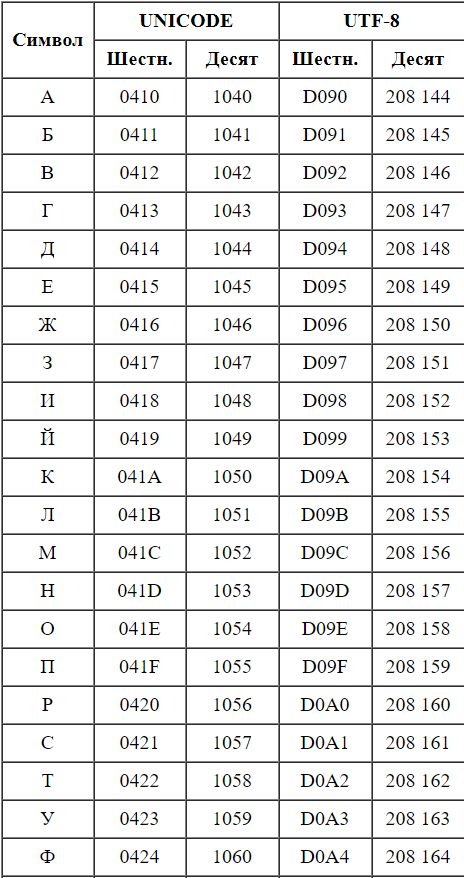
6. Выбор среды разработки, при умении работать в средах разработки - свободный. Если Вы ранее не работали в средах разработки для Android, то лучше выбрать Android Studio (инструкция находится ниже).

7. Оформить отчет по стандартам оформления ИрНИТУ, в отчете приложить краткое описание выбранных кодировок, таблицы перевода, скриншоты исходного кода, скриншоты результата работы приложений и скрипта.

Выбранные кодировки: UTF-8 и UTF-16 (LE)

Решение задач

Таблица 1 – Таблица кодировок UNICODE (UTF-16) и UTF-8



Реализация скрипта перевода на VBscript

Скрипт перевода файла с кодировкой UTF-8 в файл с кодировкой UTF-16:

Set stream = CreateObject("ADODB.Stream")

stream.Type = 2

stream.Charset = "UTF-8"

stream.Open()

stream.LoadFromFile("UTF-8-FROM.txt")

text = stream.ReadText()

stream.Close()

stream.Charset = "UTF-16LE"

stream.Open()

stream.WriteText(text)

stream.SaveToFile "UTF-16-TO.txt", 2

stream.Close()

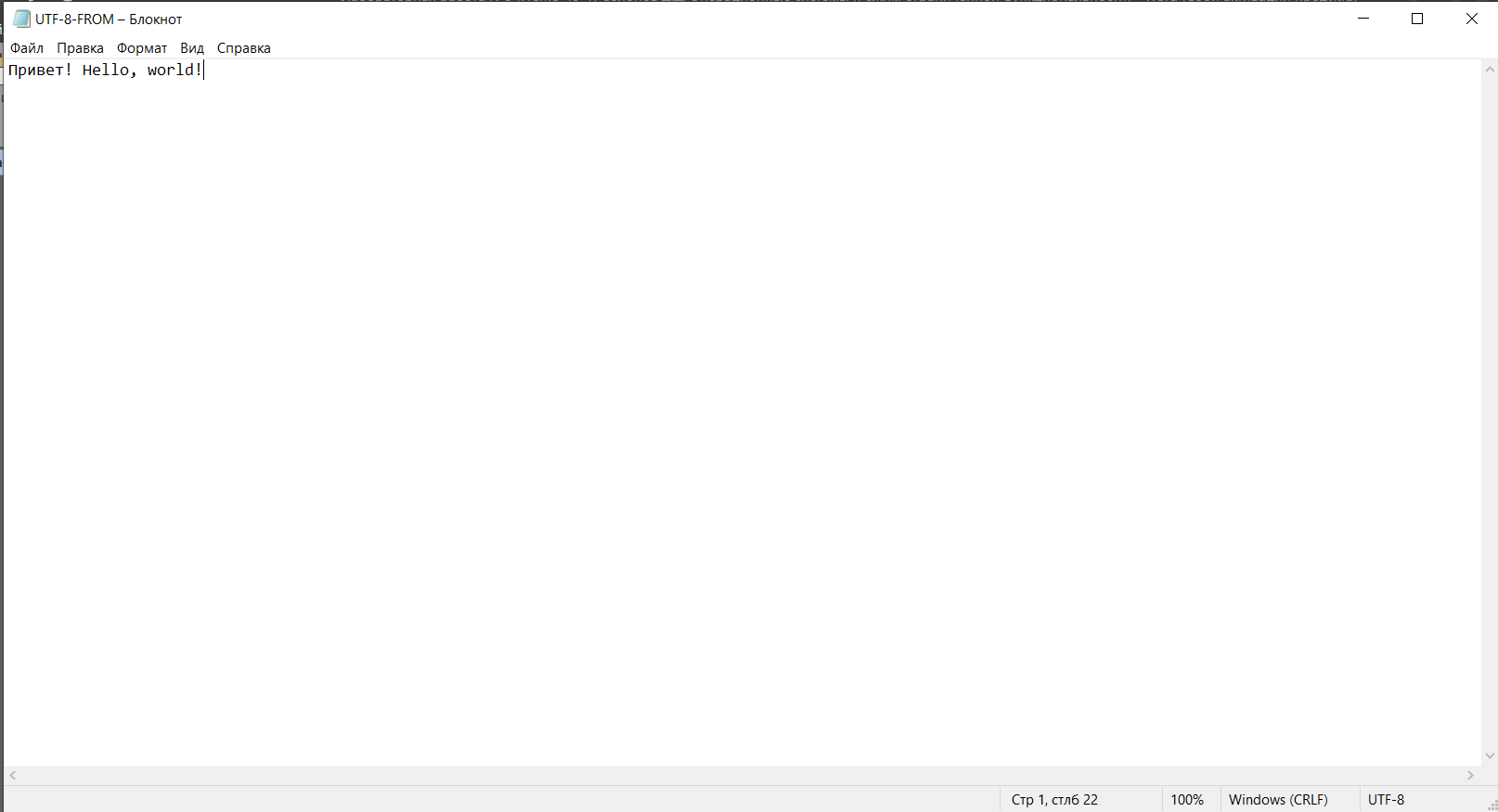


Рисунок 1 – Содержимое файла UTF-8-FROM.txt до запуска скрипта

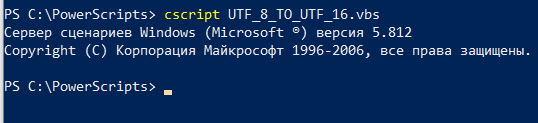


Рисунок 2 – Выполнение скрипта перевода UTF-8 в UTF-16

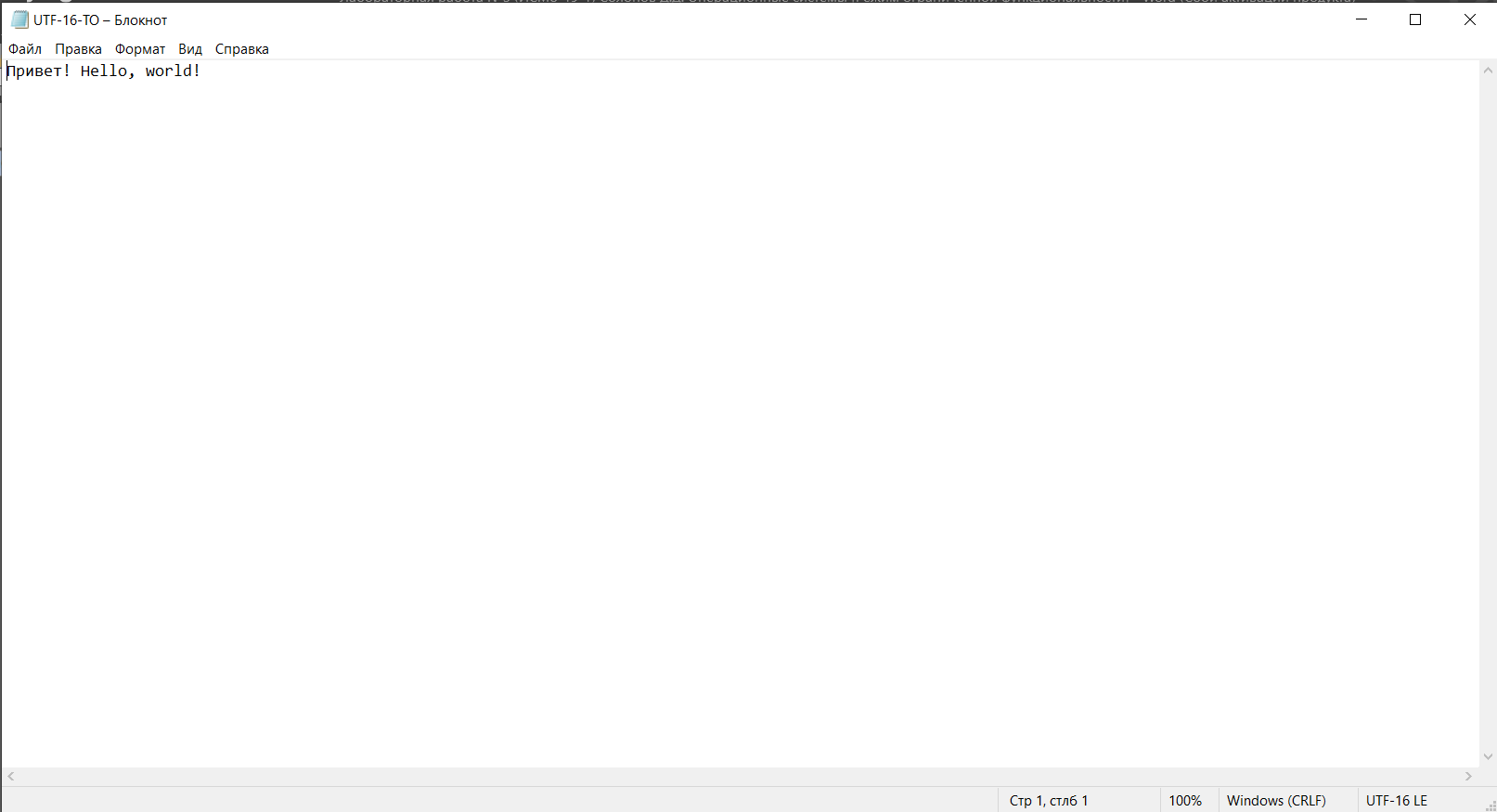


Рисунок 3 – Содержимое файла UTF-16-FROM.txt после запуска скрипта

Скрипт перевода файла с кодировкой UTF-16 в файл с кодировкой UTF-8:

Set stream = CreateObject("ADODB.Stream")

stream.Type = 2

stream.Charset = "UTF-16LE"

stream.Open()

stream.LoadFromFile("UTF-16-FROM.txt")

text = stream.ReadText()

stream.Close()

stream.Charset = "UTF-8"

stream.Open()

stream.WriteText(text)

stream.SaveToFile "UTF-8-TO.txt", 2

stream.Close()

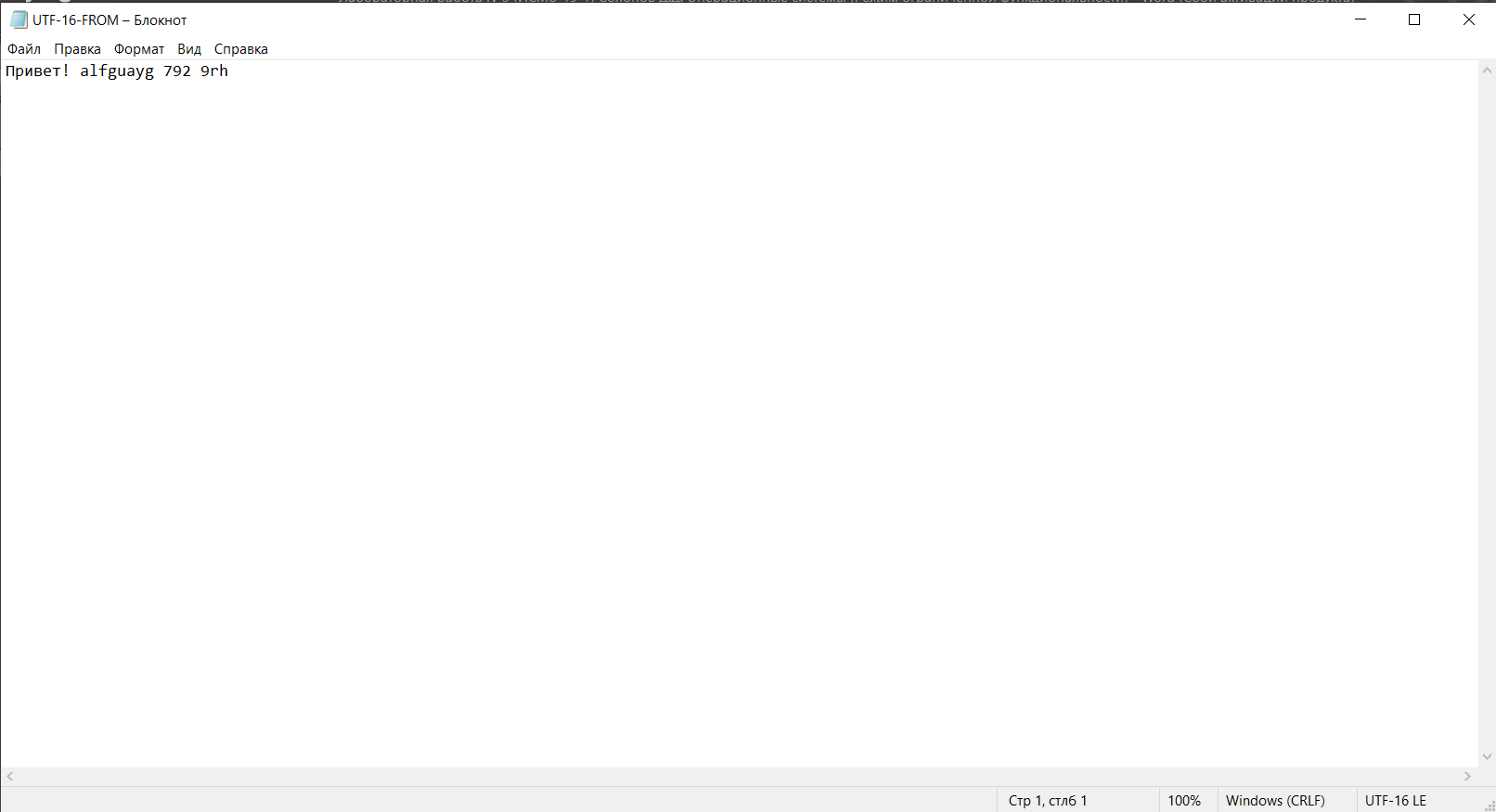


Рисунок 4 – Содержание файла UTF-16-FROM.txt до выполнения скрипта

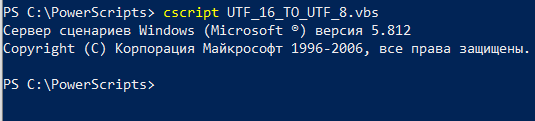


Рисунок 5 – Выполнение скрипта перевода кодировки UTF-16 в UTF-8

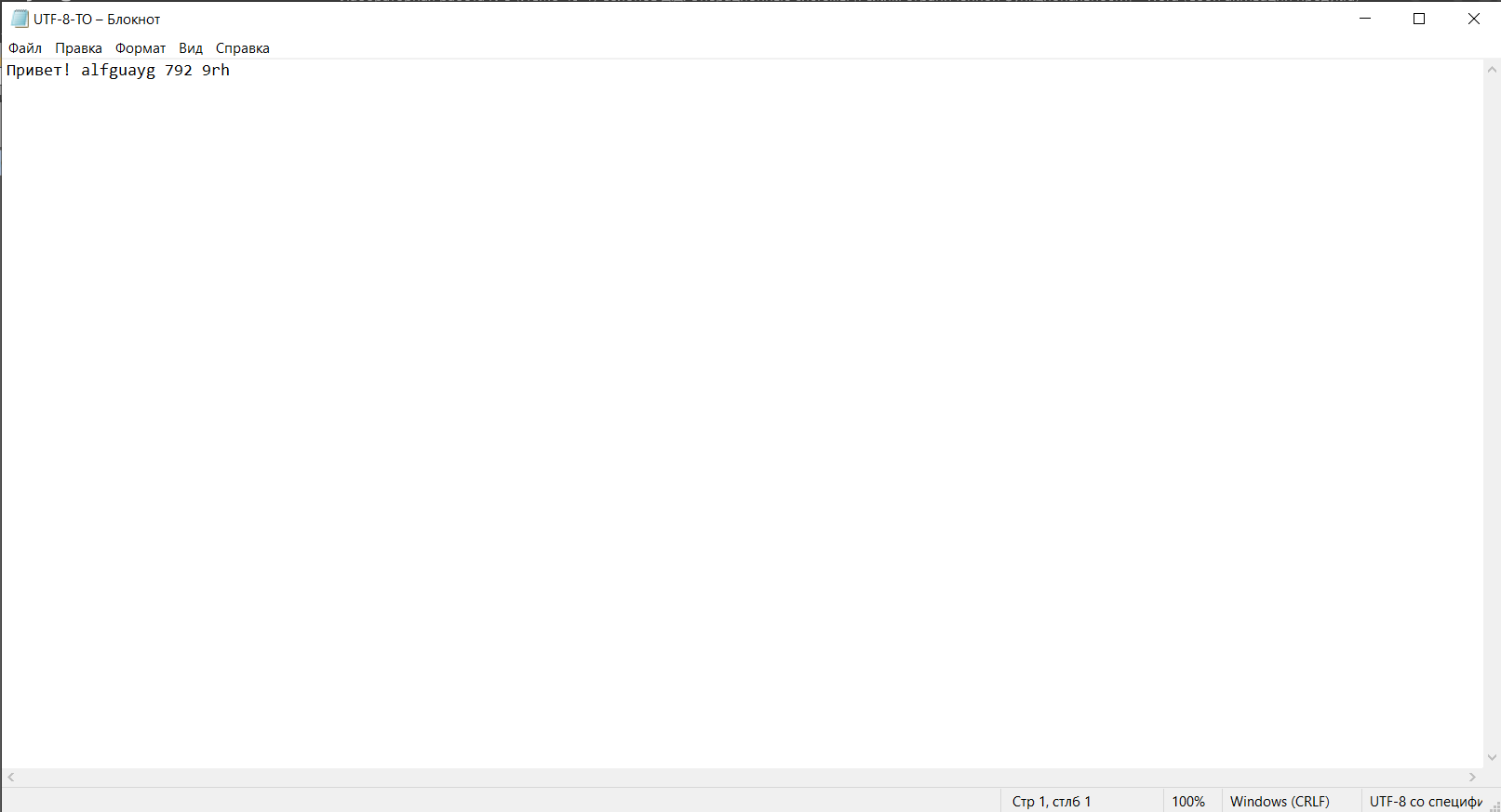


Рисунок 5 – Содержание файла UTF-8-TO.txt после выполнения скрипта

Реализация HTA-приложения

Код HTA-приложения:

<HTML>

<HEAD>

<TITLE>Application Converter</TITLE>

<meta http-equiv="Content-Type" content="text/html; charset=windows-1251"

<HTA:APPLICATION

ID="oHTA"

APPLICATIONNAME="Application"

BORDER="thin"

BORDERSTYLE="normal"

CAPTION="yes"

ICON=""

MAXIMIZEBUTTON="yes"

MINIMIZEBUTTON="yes"

SHOWINTASKBAR="no"

SINGLEINSTANCE="no"

SYSMENU="yes"

VERSION="1.0"

WINDOWSTATE="maximize" />

<style type="text/css">

#utf8text\_input{

width: 400px;

}

#convertutf8toutf16{

}

#utf16text\_result{}

#convertutf16toutf8{}

#utf16text\_input{

width: 400px;

}

#utf8text\_result{}

</style>

</HEAD>

<BODY SCROLL="no" style="background-color: orangered;">

<div style="width: inherit; height: 100px; background-color: orange;"></div>

<div style="background-color: green;">

<div style="width: 100%%; color: black; display: inline-block">

<input style="margin-left: 30px; margin-right: 30px;" id="utf8text\_input" type="text" placeholder="Введите текст в кодировке UTF-8" size="18" />

<button id="convertutf8toutf16" onclick="Button1Click()">Перевести в кодировку UTF-16</button>

</div>

<div style="margin-top: 20px; width: 100%%; height: 50px; color: black; display: inline-block">

<label style="margin-left: 30px; margin-right: 30px;" style="color: black;">Перекодированный текст (UTF-8 to UTF-16): </label>

<label id="utf16text\_result" style="color: black; margin-left: 10px;"></label>

</div>

</div>

<br><br>

<div style="background-color: green;">

<div style="width: 100%%; display: inline-block">

<input style="margin-left: 30px; margin-right: 30px;" id="utf16text\_input" type="text" placeholder="Введите текст в кодировке UTF-16" size="18" />

<button id="convertutf16toutf8" onclick="Button2Click()">Перевести в кодировку UTF-8</button>

</div>

<div style="margin-top: 20px; width: 100%%; height: 50px; color: black; display: inline-block">

<label style="margin-left: 30px; margin-right: 30px;" style="color: black;">Перекодированный текст (UTF-16 to UTF-8): </label>

<label id="utf8text\_result" style="color: black; margin-left: 10px;"></label>

</div>

</div>

<div style="width: inherit; height: 500px; background-color: orange;"></div>

<script language="javascript">

//Перевод из Utf-8 в windows-1251

function fromUtf8(utftext) {

var string = "";

var i = 0;

var c = c1 = c2 = 0;

while ( i < utftext.length ) {

c = utftext.charCodeAt(i);

if (c < 128) {

string += String.fromCharCode(c);

i++;

}

else if((c > 191) && (c < 224)) {

c2 = utftext.charCodeAt(i+1);

string += String.fromCharCode(((c & 31) << 6) | (c2 & 63));

i += 2;

}

else {

c2 = utftext.charCodeAt(i+1);

c3 = utftext.charCodeAt(i+2);

string += String.fromCharCode(((c & 15) << 12) | ((c2 & 63) << 6) | (c3 & 63));

i += 3;

}

}

return string;

}

//Перевод текста из windows-1251 в Utf-8

function toUtf8(string) {

string = string.replace(/\r\n/g,"\n");

var utftext = "";

for (var n = 0; n < string.length; n++) {

var c = string.charCodeAt(n);

if (c < 128) {

utftext += String.fromCharCode(c);

}

else if((c > 127) && (c < 2048)) {

utftext += String.fromCharCode((c >> 6) | 192);

utftext += String.fromCharCode((c & 63) | 128);

}

else {

utftext += String.fromCharCode((c >> 12) | 224);

utftext += String.fromCharCode(((c >> 6) & 63) | 128);

utftext += String.fromCharCode((c & 63) | 128);

}

}

return utftext;

}

//Перевод из Utf-8 в Utf-16

function Utf8To16(input) {

var \_escape = function(s) {

function q(c) {

c = c.charCodeAt();

return '%' + (c<16 ? '0' : '') + c.toString(16).toUpperCase();

}

return s.replace(/[\x00-),:-?[-^`{-\xFF]/g, q);

};

try{

return decodeURIComponent(\_escape(input));

}catch (URIError) {

return input;

}

}

//Перевод из Utf-16 в Utf-8

function Utf16To8(input) {

var \_unescape = function(s) {

function d(x, n) {

return String.fromCharCode(parseInt(n, 16));

}

return s.replace(/%([0-9A-F]{2})/ig, d);

};

try{

return \_unescape(encodeURIComponent(input));

}catch (URIError) {

return input;

}

}

function convertUtf8ToUtf16LE(str) {

return Utf8To16(toUtf8(str));

}

function convertUtf16LEToUtf8(str) {

return Utf16To8(toUtf8(str));

}

//перекодировка Windows-1251 в Utf16

function Win1251ToUtf16(str){

if (str == null){ return null;}

var result = "";

var o\_code = "";

var i\_code = "";

for (var I=0; I < str.length; I++){

i\_code = str.charCodeAt(I);

if (i\_code == 184){

o\_code = 1105;

} else if (i\_code == 168){

o\_code = 1025;

} else if (i\_code > 191 && i\_code < 256){

o\_code = i\_code + 848;

} else {

o\_code = i\_code;

}

result = result + String.fromCharCode(o\_code);

}

return result;

}

function Button1Click(){

utf16text\_result.innerHTML = Utf8To16(toUtf8(utf8text\_input.value));

}

function Button2Click(){

utf8text\_result.innerHTML = fromUtf8(Utf16To8(Utf8To16(toUtf8(utf16text\_input.value))));

}

</script>

</BODY>

</HTML>

На странице установлена кодировка windows-1251, следовательно вводимый текст в поля ввода будет изначально находиться в кодировке windows-1251. Для преобразования кодировки windows-1251 в UTF-8 используется метод toUtf8(), а преобразование из UTF-8 в windows-1251 используется с помощью метода fromUtf8().

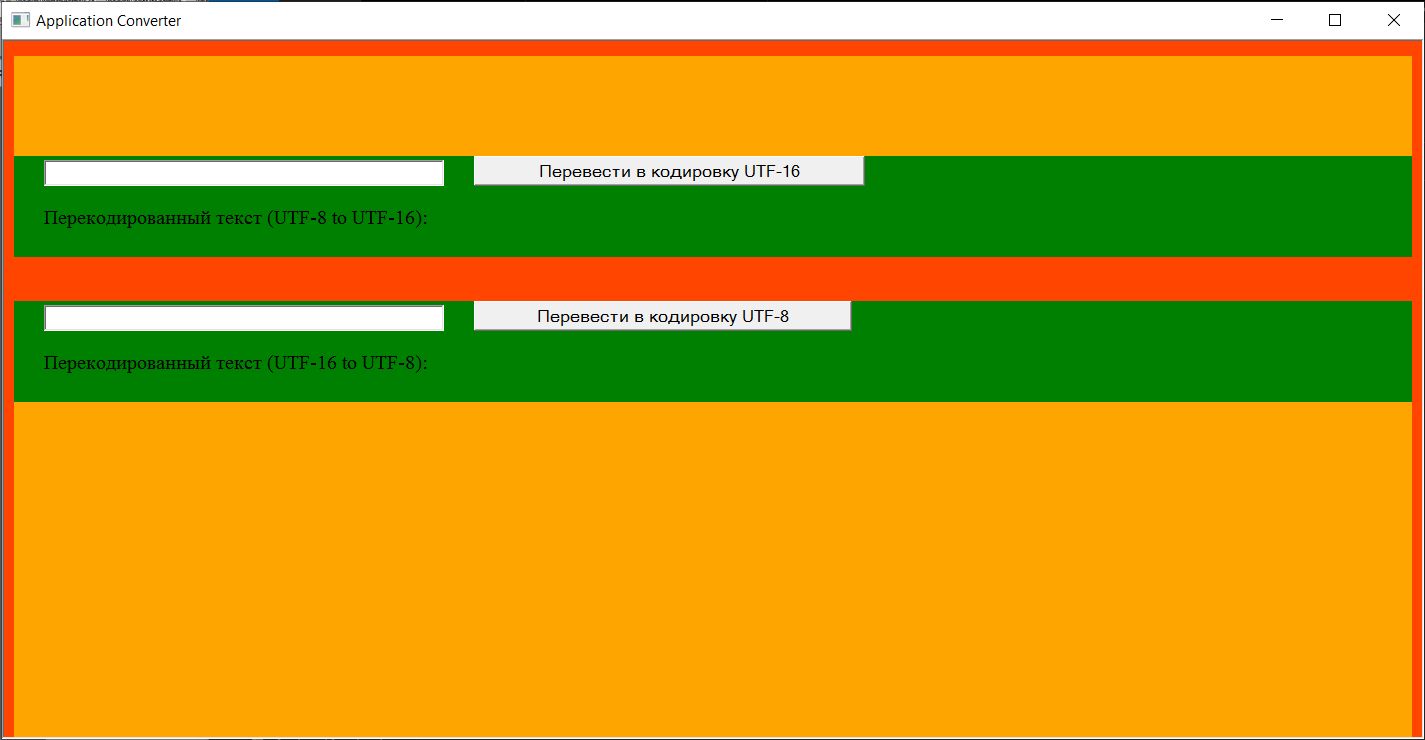


Рисунок 6 – Пользовательский интерфейс HTA-Приложения

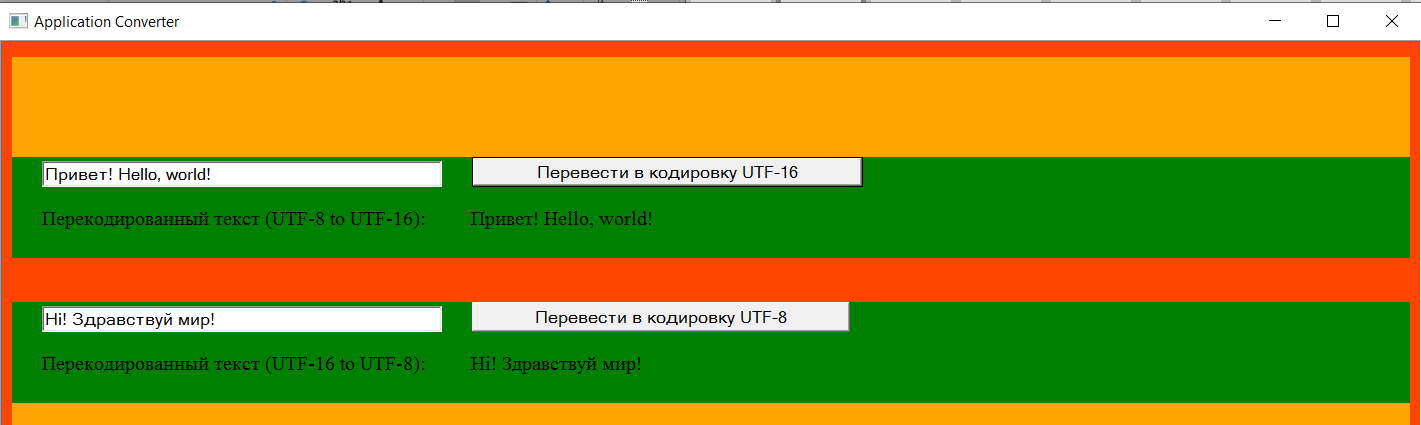


Рисунок 7 – Результат конвертации кодировки UTF-8 в UTF-16 и обратно.

Реализация приложения с графическим интерфейсом

**Код приложения:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.IO;

using System.Linq;

using System.Runtime.InteropServices;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace ChangeEncoding

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

openFileDialog1.Filter = "Text files(\*.txt)|\*.txt|All files(\*.\*)|\*.\*";

comboBox1.Items.AddRange(new object[] { "UTF-16LE", "UTF-8" });

comboBox2.Items.AddRange(new object[] { "UTF-8", "UTF-16LE" });

}

private void \_btnToUtf16\_Click(object sender, EventArgs e)

{

richTextBox2.Text = "";

richTextBox5.Text = "";

if (richTextBox1.Text.ToString().Length == 0)

{

MessageBox.Show("Ошибка: не введён текст для преобразования!");

return;

}

byte[] utf8Bytes = Encoding.UTF8.GetBytes(richTextBox1.Text.ToString());

byte[] utf16Bytes = Encoding.Convert(Encoding.Unicode, Encoding.UTF8, utf8Bytes);

richTextBox5.Text = richTextBox5.Text + "\n{ ";

for (int i = 0; i < utf8Bytes.Length; i++)

{

richTextBox5.Text = richTextBox5.Text + utf8Bytes[i].ToString() + ", ";

}

richTextBox5.Text = richTextBox5.Text + "} ";

richTextBox2.Text = richTextBox2.Text + "{ ";

for (int i = 0; i < utf16Bytes.Length; i++)

{

richTextBox2.Text = richTextBox2.Text + utf16Bytes[i].ToString() + ", ";

}

richTextBox2.Text = richTextBox2.Text + "} ";

}

private void button1\_Click(object sender, EventArgs e)

{

richTextBox3.Text = "";

richTextBox6.Text = "";

if (richTextBox4.Text.ToString().Length == 0)

{

MessageBox.Show("Ошибка: не введён текст для преобразования!");

return;

}

byte[] utf16Bytes =

Encoding.Convert(Encoding.Unicode, Encoding.UTF8,

Encoding.UTF8.GetBytes(richTextBox4.Text));

byte[] utf8Bytes = Encoding.Convert(Encoding.UTF8, Encoding.Unicode, utf16Bytes);

richTextBox6.Text = richTextBox6.Text + "\n{ ";

for (int i = 0; i < utf16Bytes.Length; i++)

{

richTextBox6.Text = richTextBox6.Text + utf16Bytes[i].ToString() + ", ";

}

richTextBox6.Text = richTextBox6.Text + "} ";

richTextBox3.Text = richTextBox3.Text + "{ ";

for (int i = 0; i < utf8Bytes.Length; i++)

{

richTextBox3.Text = richTextBox3.Text + utf8Bytes[i].ToString() + ", ";

}

richTextBox3.Text = richTextBox3.Text + "} ";

}

private string filename = "";

private void button2\_Click(object sender, EventArgs e)

{

if (openFileDialog1.ShowDialog() == DialogResult.Cancel)

return;

filename = openFileDialog1.FileName;

}

bool ValidComboBox(ComboBox cmb)

{

if ((!cmb.Text.Equals("UTF-16LE"))

&& (!cmb.Text.Equals("UTF-8")))

return false;

return true;

}

private void button3\_Click(object sender, EventArgs e)

{

if((!ValidComboBox(comboBox1)) || (!ValidComboBox(comboBox2)))

{

MessageBox.Show("Ошибка: не выбрана кодировка файла или перекодирования!",

"Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}else if(filename.Length == 0)

{

MessageBox.Show("Ошибка: не выбран файл для перекодирования!",

"Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

string newFileName = filename + comboBox2.Text.ToString() + ".txt";

using (StreamReader reader = new StreamReader(filename, Encoding.GetEncoding(comboBox1.Text.ToString())))

using (StreamWriter writer = new StreamWriter(newFileName, false, Encoding.GetEncoding(comboBox2.Text.ToString())))

{

writer.Write(reader.ReadToEnd());

}

}

}

}

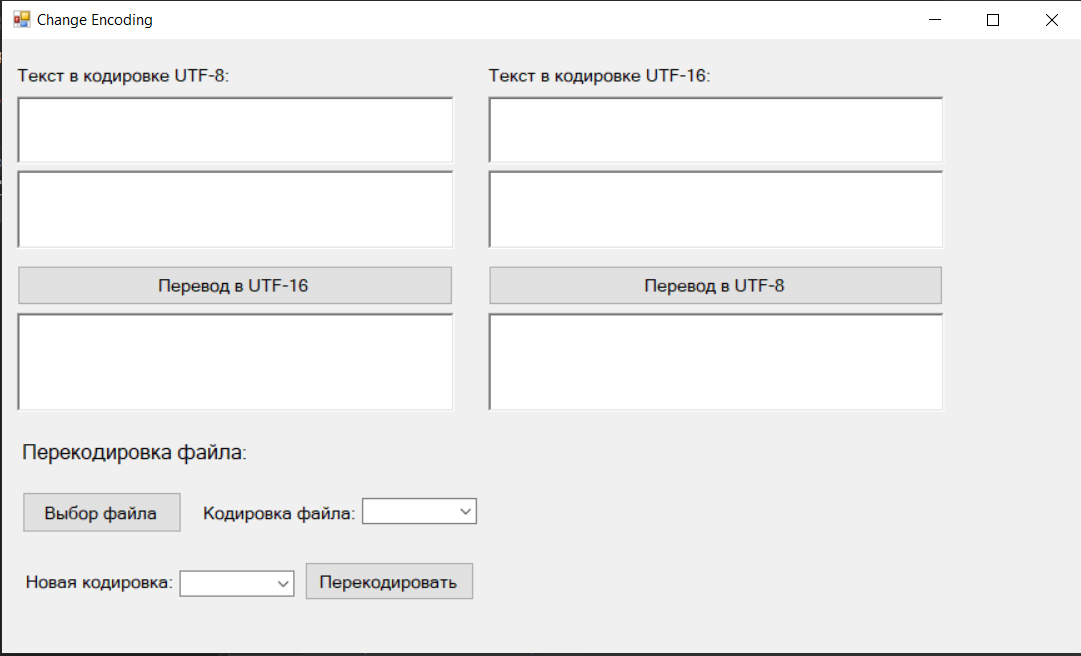


Рисунок 8 – Пользовательский интерфейс

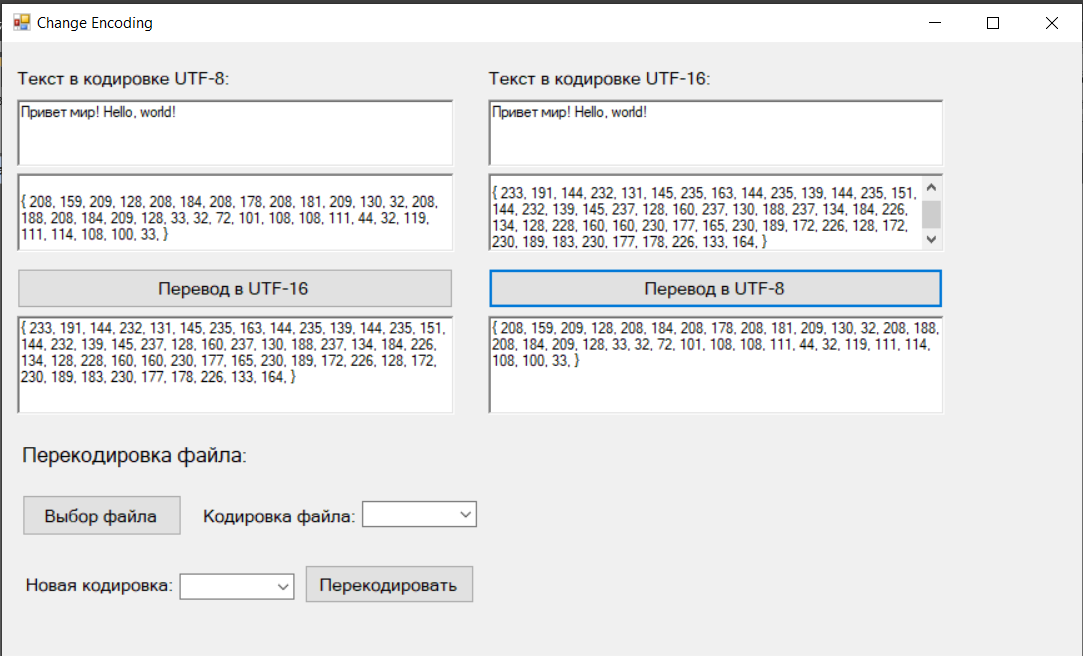


Рисунок 9 – Результат перевода кодировки UTF-8 в UTF-16 и обратно

На рисунке 9 продемонстрирован перевод кодировки UTF-8 в UTF-16 и обратно, здесь следует обратить внимание на значения байтов. При конвертации UTF-8 в UTF-16 выводятся значения байтов этой строки в кодировке UTF-8, а затем байты этой строки в кодировке UTF-16. При переводе кодировки UTF-16 в UTF-8 вывод реализуется аналогичным образом (рассматривая байты можно заметить, что перевод корректен).

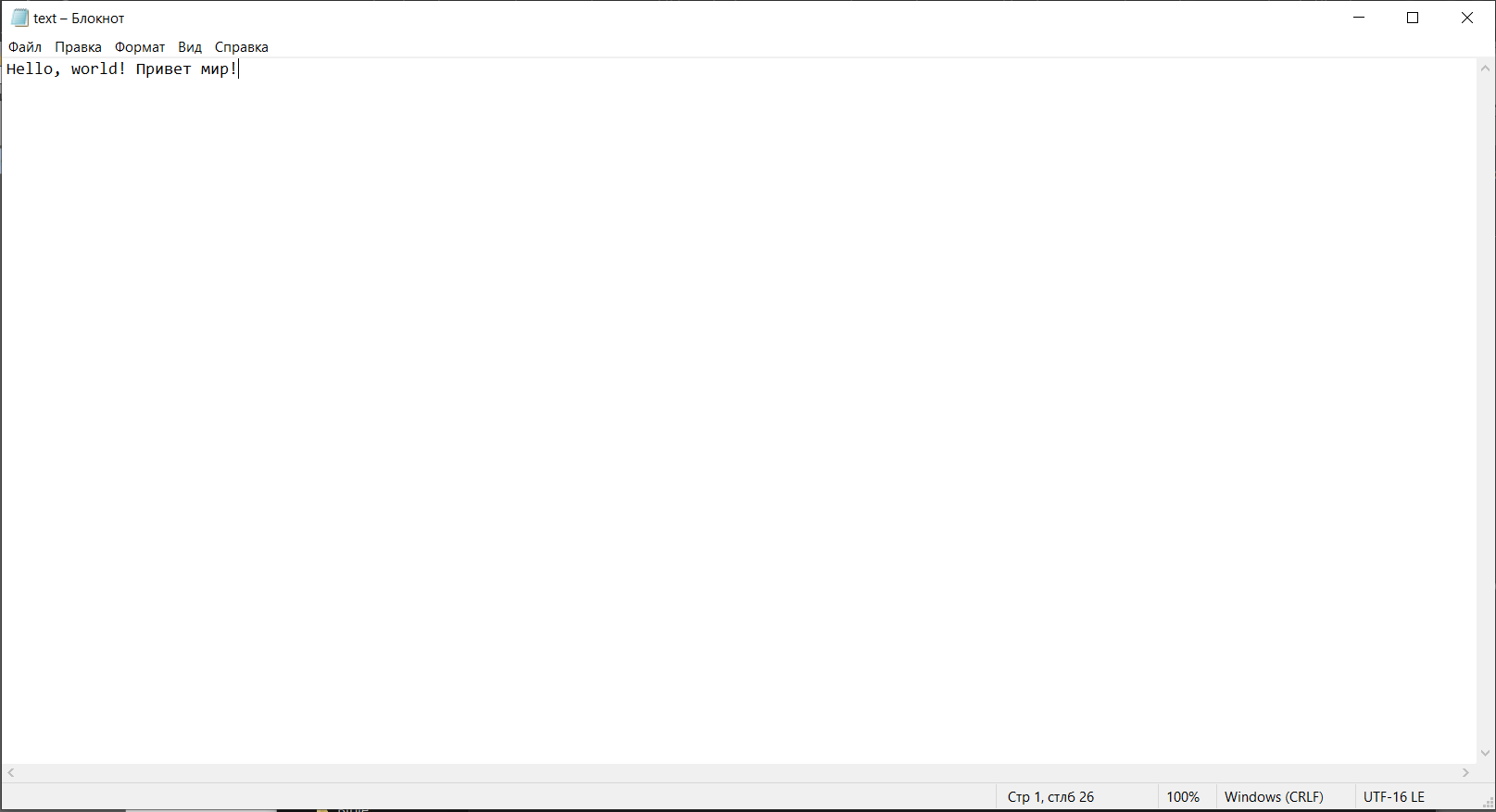


Рисунок 10 – Содержимое файла text.txt до преобразования к кодировке UTF-8

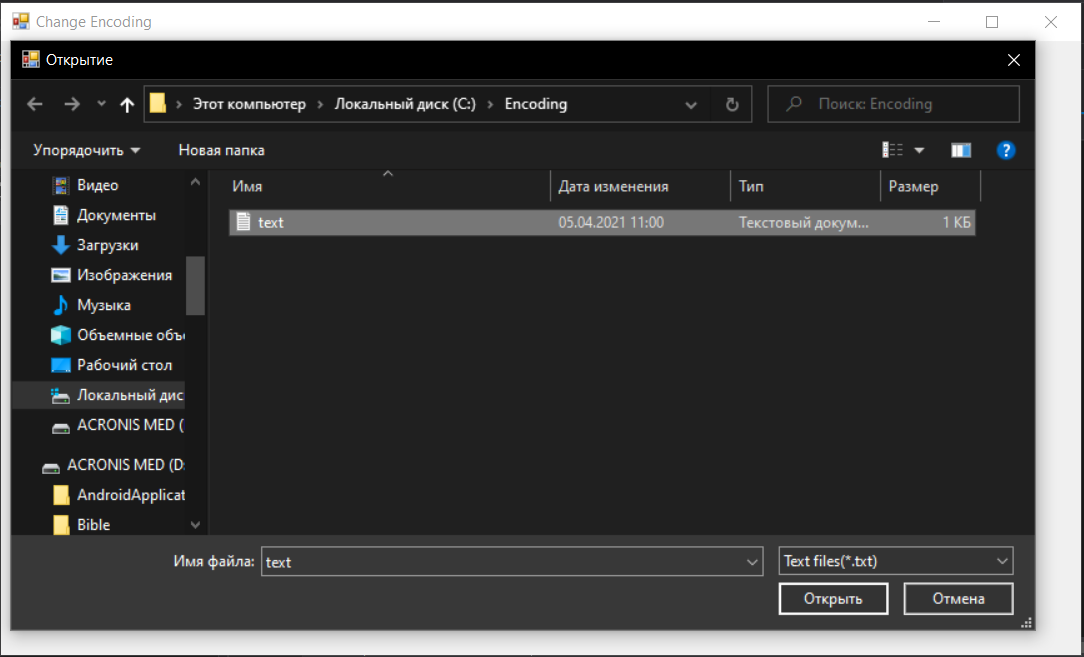


Рисунок 11 – Выбор файла text.txt для преобразования

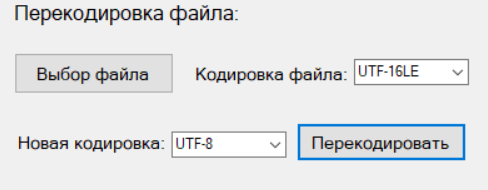


Рисунок 12 – Перекодирование файла text.txt в кодировку UTF-8

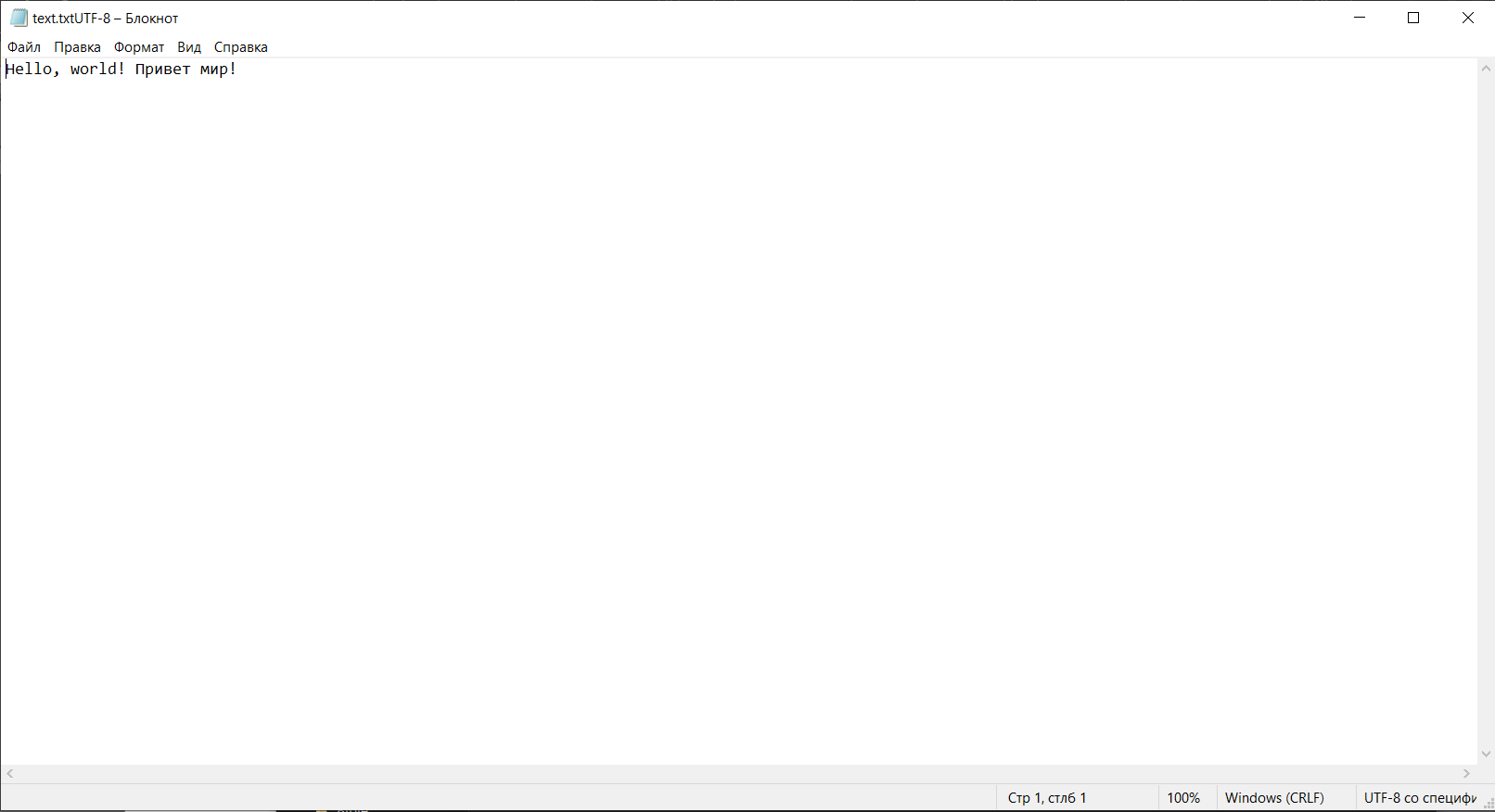


Рисунок 13 – Результат перевода UTF-16 в UTF-8



Рисунок 14 – Результат перевода UTF-8 в UTF-16 (обратное преобразование рисунку 13)

Реализация мобильного приложения

**Вёрстка:**

<?xml version="1.0" encoding="utf-8"?>

<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".MainActivity">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="306dp"

android:background="@color/cardview\_shadow\_start\_color"

android:orientation="horizontal">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="#009955"

android:orientation="vertical">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="58dp"

android:orientation="horizontal">

<TextView

android:layout\_width="82dp"

android:layout\_height="match\_parent"

android:layout\_marginTop="20px"

android:gravity="center\_horizontal"

android:text="Текст UTF-8"

android:textAlignment="center"

android:textColor="@color/black"

android:textSize="30px"></TextView>

<EditText

android:id="@+id/inputUtf8"

android:layout\_width="152dp"

android:layout\_height="wrap\_content"

android:textSize="30px"></EditText>

<Button

android:id="@+id/convertToUtf16"

android:layout\_width="wrap\_content"

android:layout\_height="38dp"

android:gravity="center\_horizontal"

android:text="UTF-16"

android:textAlignment="center"

android:textSize="30px"></Button>

</LinearLayout>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="58dp"

android:orientation="horizontal">

<TextView

android:layout\_width="400px"

android:layout\_height="match\_parent"

android:paddingTop="20px"

android:gravity="center\_horizontal"

android:text="Текст в кодировке UTF-16:"

android:textAlignment="center"

android:textColor="@color/black"

android:textSize="30px"></TextView>

<EditText

android:id="@+id/resultConvertUtf816"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:textSize="30px"></EditText>

</LinearLayout>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="55dp"

android:background="#565656"></LinearLayout>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="125px"

android:orientation="horizontal">

<TextView

android:layout\_width="82dp"

android:layout\_height="match\_parent"

android:paddingTop="20px"

android:gravity="center\_horizontal"

android:text="Текст UTF-16"

android:textAlignment="center"

android:textColor="@color/black"

android:textSize="30px"></TextView>

<EditText

android:id="@+id/inputUtf16"

android:layout\_width="156dp"

android:layout\_height="wrap\_content"

android:textSize="30px"></EditText>

<Button

android:id="@+id/convertToUtf8"

android:layout\_width="wrap\_content"

android:layout\_height="37dp"

android:gravity="center\_horizontal"

android:text="UTF-8"

android:textAlignment="center"

android:textSize="30px"></Button>

</LinearLayout>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="60dp"

android:orientation="horizontal">

<TextView

android:layout\_width="400px"

android:layout\_height="match\_parent"

android:paddingTop="20px"

android:gravity="center\_horizontal"

android:text="Текст в кодировке UTF-8:"

android:textAlignment="center"

android:textColor="@color/black"

android:textSize="30px"></TextView>

<EditText

android:id="@+id/resultConvertUtf168"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:textSize="30px"></EditText>

</LinearLayout>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="58dp">

<Button

android:id="@+id/chooseFile"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:gravity="center\_horizontal"

android:text="Выбор файла"

android:textAlignment="center"

android:textSize="30px"></Button>

<Spinner

android:id="@+id/spinner1"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:entries="@array/coding"

android:gravity="center\_horizontal"

android:textAlignment="center" />

</LinearLayout>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="58dp">

<Button

android:id="@+id/coding"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:gravity="center\_horizontal"

android:text="Перекодировать"

android:textAlignment="center"

android:textSize="30px"></Button>

<Spinner

android:id="@+id/spinner2"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:entries="@array/coding"

android:gravity="center\_horizontal"

android:textAlignment="center" />

</LinearLayout>

</LinearLayout>

</LinearLayout>

</ScrollView>

**Файл манифеста:**

<?xml version="1.0" encoding="utf-8"?>

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

package="com.game.changeencoding">

<uses-permission android:name="android.permission.READ\_EXTERNAL\_STORAGE" />

<uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE" />

<application

android:allowBackup="true"

android:icon="@mipmap/ic\_launcher"

android:label="@string/app\_name"

android:roundIcon="@mipmap/ic\_launcher\_round"

android:supportsRtl="true"

android:theme="@style/Theme.ChangeEncoding">

<activity android:name=".MainActivity">

<intent-filter>

<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />

</intent-filter>

</activity>

</application>

</manifest>

**Код приложения:**

package com.game.changeencoding;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Context;

import android.content.Intent;

import android.net.Uri;

import android.os.Bundle;

import android.os.Environment;

import android.os.FileUtils;

import android.util.Log;

import android.view.View;

import android.widget.AdapterView;

import android.widget.Button;

import android.widget.EditText;

import android.widget.Spinner;

import android.widget.Toast;

import java.io.File;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.FileOutputStream;

import java.io.IOException;

import java.io.InputStreamReader;

import java.io.OutputStreamWriter;

import java.io.Reader;

import java.io.UnsupportedEncodingException;

import java.io.Writer;

public class MainActivity extends AppCompatActivity {

private EditText \_inputUtf8;

private Button \_convertToUtf16;

private EditText \_resultConvertUtf816;

private EditText \_inputUtf16;

private Button \_convertToUtf8;

private EditText \_resultConvertUtf168;

private Button \_chooseFile;

private Button \_coding;

private Spinner \_spinner1;

private Spinner \_spinner2;

private String fileCoding = "";

private String fileEncoding = "";

private static final int FILE\_SELECT\_CODE = 0;

private static String filePath = "";

private void showFileChooser() {

Intent intent = new Intent(Intent.ACTION\_GET\_CONTENT);

intent.setType("\*/\*");

intent.addCategory(Intent.CATEGORY\_OPENABLE);

try {

startActivityForResult(

Intent.createChooser(intent, "Выбор файла"),

FILE\_SELECT\_CODE);

} catch (android.content.ActivityNotFoundException ex) {

Toast.makeText(this, "Требуется установка файлового менеджера!",

Toast.LENGTH\_SHORT).show();

filePath = "";

}

}

private Uri uriUser = null;

@Override

protected void onActivityResult(int requestCode, int resultCode, Intent data) {

switch (requestCode) {

case FILE\_SELECT\_CODE:

if (resultCode == RESULT\_OK) {

Uri uri = data.getData();

filePath = data.getData().getPath();

uriUser = uri;

}

break;

}

super.onActivityResult(requestCode, resultCode, data);

}

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

\_inputUtf8 = (EditText) findViewById(R.id.inputUtf8);

\_convertToUtf16 = (Button) findViewById(R.id.convertToUtf16);

\_resultConvertUtf816 = (EditText) findViewById(R.id.resultConvertUtf816);

\_inputUtf16 = (EditText) findViewById(R.id.inputUtf16);

\_convertToUtf8 = (Button) findViewById(R.id.convertToUtf8);

\_resultConvertUtf168 = (EditText) findViewById(R.id.resultConvertUtf168);

\_chooseFile = (Button) findViewById(R.id.chooseFile);

\_coding = (Button) findViewById(R.id.coding);

\_spinner1 = (Spinner) findViewById(R.id.spinner1);

\_spinner2 = (Spinner) findViewById(R.id.spinner2);

\_spinner1.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {

public void onItemSelected(AdapterView<?> parent,

View itemSelected, int selectedItemPosition, long selectedId) {

String[] choose = getResources().getStringArray(R.array.coding);

fileCoding = choose[selectedItemPosition];

}

public void onNothingSelected(AdapterView<?> parent) {

}

});

\_spinner2.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {

public void onItemSelected(AdapterView<?> parent,

View itemSelected, int selectedItemPosition, long selectedId) {

String[] choose = getResources().getStringArray(R.array.coding);

fileEncoding = choose[selectedItemPosition];

}

public void onNothingSelected(AdapterView<?> parent) {

}

});

\_chooseFile.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

showFileChooser();

}

});

\_coding.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

File sdPath = Environment.getExternalStorageDirectory();

try (final FileInputStream fileInputStream = new FileInputStream(sdPath + "/" +

filePath.split(":")[1]);

final InputStreamReader inputStreamReader = new InputStreamReader(fileInputStream, fileCoding);

final FileOutputStream fileOutputStream = new FileOutputStream(sdPath + "/" +

filePath.split(":")[1] + fileEncoding + ".txt");

final OutputStreamWriter outputStreamWriter = new OutputStreamWriter(fileOutputStream, fileEncoding))

{

final int BUFFER\_SIZE = 2048;

final char[] buffer = new char[BUFFER\_SIZE];

int readed;

while ((readed = inputStreamReader.read(buffer, 0, BUFFER\_SIZE)) > 0)

outputStreamWriter.write(buffer, 0, readed);

} catch (Exception e) {

e.printStackTrace();

}

}

});

\_convertToUtf16.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

if(\_inputUtf8.getText().toString().length() == 0){

Toast.makeText(MainActivity.this, "Ошибка: отсутствует текст, который нужно перевести из " +

"кодировки UTF-8 в UTF-16", Toast.LENGTH\_LONG).show();

return;

}

String textInUtf16 = \_inputUtf8.getText().toString();

try {

byte[] utf8 = textInUtf16.getBytes("UTF-8");

String textInUtf8 = new String(utf8, "UTF-8");

byte[] utf16 = textInUtf8.getBytes("UTF-16LE");

\_resultConvertUtf816.setText(new String(utf16, "UTF-16LE"));

} catch (UnsupportedEncodingException e) {

e.printStackTrace();

}

}

});

\_convertToUtf8.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

if(\_inputUtf16.getText().toString().length() == 0){

Toast.makeText(MainActivity.this, "Ошибка: отсутствует текст, который нужно перевести из " +

"кодировки UTF-16LE в UTF-8", Toast.LENGTH\_LONG).show();

return;

}

String textInUtf16 = \_inputUtf16.getText().toString();

try{

byte[] utf8 = textInUtf16.getBytes("UTF-8");

String textInUtf8 = new String(utf8, "UTF-8");

\_resultConvertUtf168.setText(textInUtf8);

} catch (UnsupportedEncodingException e) {

e.printStackTrace();

}

}

});

}

}



Рисунок 15 – Интерфейс мобильного приложения

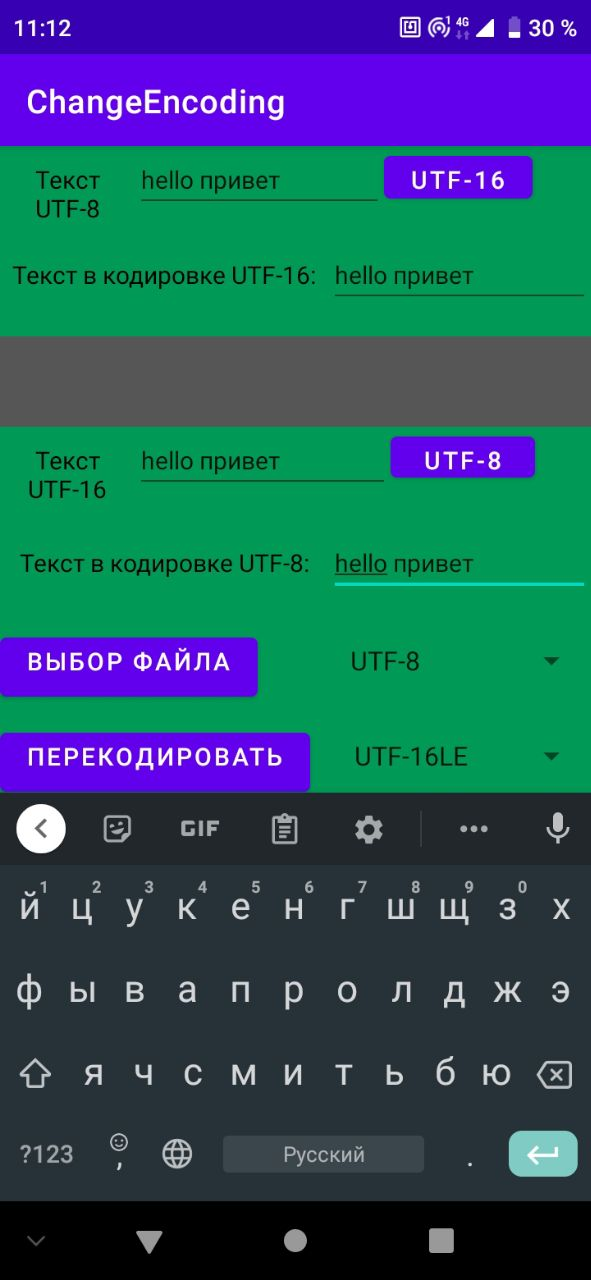


Рисунок 16 – Результат перевода из кодировки UTF-8 в UTF-16 и обратно