Донецкий Национальный Технический Университет

Лабораторная работа № 2

«Создание и отправка SMS»

Выполнил:

ст. группы ПИ-13

Лысенко А. С.

Проверил:

ст. препод. Костин В.И.

Покровск 2017

Задание. Предоставить пользователю возможность вводить SMS. Адрес получателя выбирается из списка (в дальнейшем контактной книги). Если телефона получателя SMS нет в списке, дать возможность пользователю его добавить. Также предоставить возможность пользователю отправить SMS без сохранения номера телефона получателя. Список последних отправленных десяти SMS должен быть сохранён. Кроме этого организовать работу с шаблонами SMS (создание, просмотр и вставка шаблона в текст SMS).

Отправка смс происходит через интернет службу sms-центр, с помощью программы пользователь получает доступ к аккаунту в службе и через него отправляет сообщения.

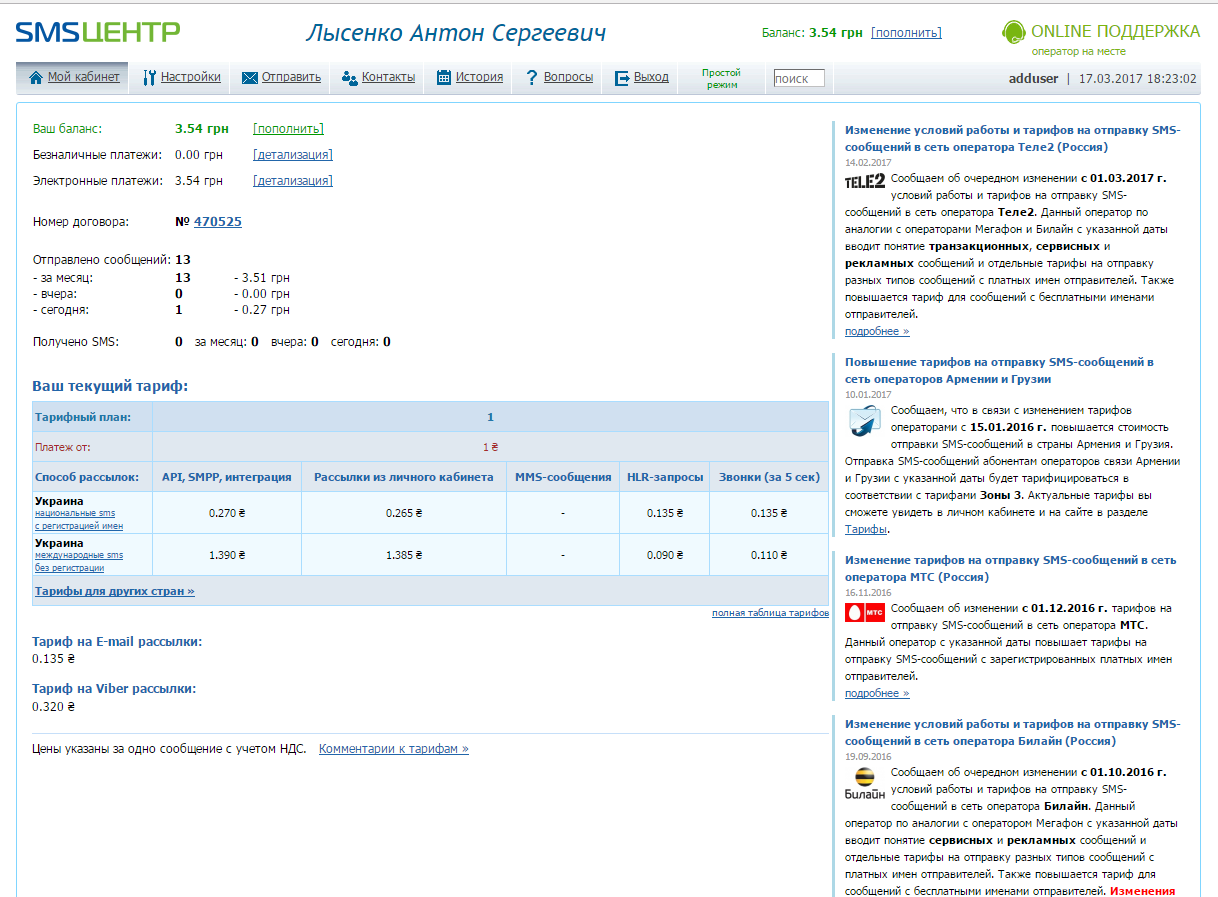


Рис. 1 – кабинет службы

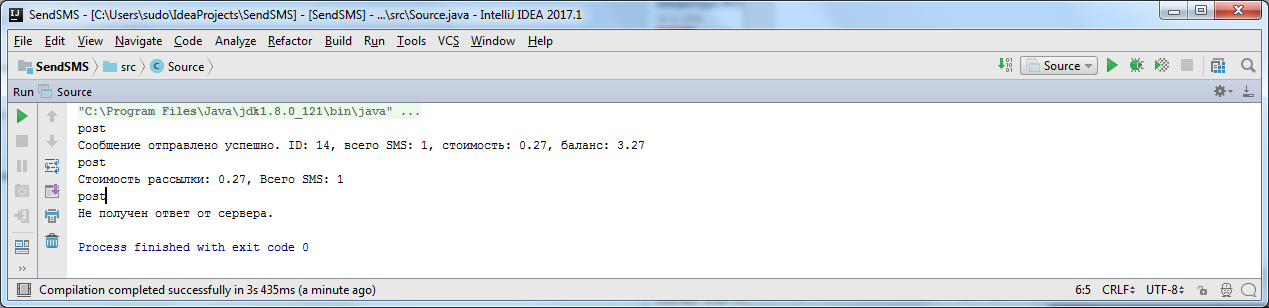


Рис. 2 – работа программы



Рис. 3 – приём сообщения

Source.java

*/\*  
 \* SMSC.RU API (smsc.ru) версия 1.2 (19.02.2017) smsc's sms sender package  
 \*/***import** java.net.\*;  
**import** java.io.\*;  
**import** java.lang.Math;  
  
**public class** Source {  
 **public static void** main(String[] args) {  
  
 Source sd= **new** Source(**"adduser"**, **"Lys\_1996"**, **"utf-8"**, **true**);  
  
 sd.send\_sms(**"380505917399"**, **"Ваш пароль: 123"**, 1, **""**, **""**, 0, **""**, **""**);  
 sd.get\_sms\_cost(**"380505917399"**, **"Вы успешно зарегистрированы!"**, 0, 0, **""**, **""**);  
 sd.get\_balance();  
 }  
  
 String **SMSC\_LOGIN** = **"adduser"**; *// логин клиента* String **SMSC\_PASSWORD** = **"Lys\_1996"**; *// пароль или MD5-хеш пароля в нижнем регистре* **boolean SMSC\_HTTPS** = **false**; *// использовать HTTPS протокол* String **SMSC\_CHARSET** = **"utf-8"**; *// кодировка сообщения: koi8-r, windows-1251 или utf-8 (по умолчанию)* **boolean SMSC\_DEBUG** = **false**; *// флаг отладки* **boolean SMSC\_POST** = **true**; *// Использовать метод POST  
  
 /\*\*  
 \* constructors  
 \*/* **public** Source(String login, String password) {  
 **SMSC\_LOGIN** = login;  
 **SMSC\_PASSWORD** = password;  
 }  
  
 **public** Source(String login, String password, String charset) {  
 **SMSC\_LOGIN** = login;  
 **SMSC\_PASSWORD** = password;  
 **SMSC\_CHARSET** = charset;  
 }  
  
 **public** Source(String login, String password, String charset, **boolean** debug) {  
 **SMSC\_LOGIN** = login;  
 **SMSC\_PASSWORD** = password;  
 **SMSC\_CHARSET** = charset;  
 **SMSC\_DEBUG** = debug;  
 }  
  
 */\*\*  
 \* Отправка SMS  
 \*  
 \** ***@param phones*** *- список телефонов через запятую или точку с запятой  
 \** ***@param message*** *- отправляемое сообщение  
 \** ***@param translit*** *- переводить или нет в транслит (1,2 или 0)  
 \** ***@param time*** *- необходимое время доставки в виде строки (DDMMYYhhmm, h1-h2, 0ts, +m)  
 \** ***@param id*** *- идентификатор сообщения. Представляет собой 32-битное число в диапазоне от 1 до 2147483647.  
 \** ***@param format*** *- формат сообщения (0 - обычное sms, 1 - flash-sms, 2 - wap-push, 3 - hlr, 4 - bin, 5 - bin-hex, 6 - ping-sms, 7 - mms, 8 - mail, 9 - call)  
 \** ***@param sender*** *- имя отправителя (Sender ID). Для отключения Sender ID по умолчанию необходимо в качестве имени передать пустую строку или точку.  
 \** ***@param query*** *- строка дополнительных параметров, добавляемая в URL-запрос ("valid=01:00&maxsms=3&tz=2")  
 \** ***@return*** *array (<id>, <количество sms>, <стоимость>, <баланс>) в случае успешной отправки  
 \* или массив (<id>, -<код ошибки>) в случае ошибки  
 \*/* **public** String[] send\_sms(String phones, String message, **int** translit, String time, String id, **int** format, String sender, String query)  
 {  
 String[] formats = {**""**, **"flash=1"**, **"push=1"**, **"hlr=1"**, **"bin=1"**, **"bin=2"**, **"ping=1"**, **"mms=1"**, **"mail=1"**, **"call=1"**};  
 String[] m = {};  
  
 **try** {  
 m = \_smsc\_send\_cmd(**"send"**, **"cost=3&phones="** + URLEncoder.*encode*(phones, **SMSC\_CHARSET**)  
 + **"&mes="** + URLEncoder.*encode*(message, **SMSC\_CHARSET**)  
 + **"&translit="** + translit + **"&id="** + id + (format > 0 ? **"&"** + formats[format] : **""**)  
 + (sender == **""** ? **""** : **"&sender="** + URLEncoder.*encode*(sender, **SMSC\_CHARSET**))  
 + (time == **""** ? **""** : **"&time="** + URLEncoder.*encode*(time, **SMSC\_CHARSET**) )  
 + (query == **""** ? **""** : **"&"** + query));  
 }  
 **catch** (UnsupportedEncodingException e) {  
  
 }  
  
 **if** (m.**length** > 1) {  
 **if** (**SMSC\_DEBUG**) {  
 **if** (Integer.*parseInt*(m[1]) > 0) {  
 System.***out***.println(**"Сообщение отправлено успешно. ID: "** + m[0] + **", всего SMS: "** + m[1] + **", стоимость: "** + m[2] + **", баланс: "** + m[3]);  
 }  
 **else** {  
 System.***out***.print(**"Ошибка №"** + Math.*abs*(Integer.*parseInt*(m[1])));  
 System.***out***.println(Integer.*parseInt*(m[0])>0 ? (**", ID: "** + m[0]) : **""**);  
 }  
 }  
 }  
 **else** {  
 System.***out***.println(**"Не получен ответ от сервера."**);  
 }  
  
 **return** m;  
 };  
  
 */\*\*  
 \* Получение стоимости SMS  
 \*  
 \** ***@param phones*** *- список телефонов через запятую или точку с запятой  
 \** ***@param message*** *- отправляемое сообщение.  
 \** ***@param translit*** *- переводить или нет в транслит (1,2 или 0)  
 \** ***@param format*** *- формат сообщения (0 - обычное sms, 1 - flash-sms, 2 - wap-push, 3 - hlr, 4 - bin, 5 - bin-hex, 6 - ping-sms, 7 - mms, 8 - mail, 9 - call)  
 \** ***@param sender*** *- имя отправителя (Sender ID)  
 \** ***@param query*** *- строка дополнительных параметров, добавляемая в URL-запрос ("list=79999999999:Ваш пароль: 123\n78888888888:Ваш пароль: 456")  
 \** ***@return*** *array(*<*стоимость*>*,* <*количество sms*>*) либо (0, -<код ошибки>) в случае ошибки  
 \*/* **public** String[] get\_sms\_cost(String phones, String message, **int** translit, **int** format, String sender, String query)  
 {  
 String[] formats = {**""**, **"flash=1"**, **"push=1"**, **"hlr=1"**, **"bin=1"**, **"bin=2"**, **"ping=1"**, **"mms=1"**, **"mail=1"**, **"call=1"**};  
 String[] m = {};  
  
 **try** {  
 m = \_smsc\_send\_cmd(**"send"**, **"cost=1&phones="** + URLEncoder.*encode*(phones, **SMSC\_CHARSET**)  
 + **"&mes="** + URLEncoder.*encode*(message, **SMSC\_CHARSET**)  
 + **"&translit="** + translit + (format > 0 ? **"&"** + formats[format] : **""**)  
 + (sender == **""** ? **""** : **"&sender="** + URLEncoder.*encode*(sender, **SMSC\_CHARSET**))  
 + (query == **""** ? **""** : **"&"** + query));  
 }  
 **catch** (UnsupportedEncodingException e) {  
  
 }  
 *// (cost, cnt) или (0, -error)* **if** (m.**length** > 1) {  
 **if** (**SMSC\_DEBUG**) {  
 **if** (Integer.*parseInt*(m[1]) > 0)  
 System.***out***.println(**"Стоимость рассылки: "** + m[0] + **", Всего SMS: "** + m[1]);  
  
 **else** System.***out***.print(**"Ошибка №"** + Math.*abs*(Integer.*parseInt*(m[1])));  
 }  
 }  
 **else** System.***out***.println(**"Не получен ответ от сервера."**);  
  
 **return** m;  
 }  
  
 */\*\*  
 \* Проверка статуса отправленного SMS или HLR-запроса  
 \*  
 \** ***@param id*** *- ID cообщения  
 \** ***@param phone*** *- номер телефона  
 \** ***@param all*** *- дополнительно возвращаются элементы в конце массива:  
 \* (<время отправки>, <номер телефона>, <стоимость>, <sender id>, <название статуса>, <текст сообщения>)  
 \** ***@return*** *array  
 \* для отправленного SMS (<статус>, <время изменения>, <код ошибки sms>)  
 \* для HLR-запроса (<статус>, <время изменения>, <код ошибки sms>, <код страны регистрации>, <код оператора абонента>,  
 \* <название страны регистрации>, <название оператора абонента>, <название роуминговой страны>, <название роумингового оператор  
 \* <код IMSI SIM-карты>, <номер сервис-центра>)  
 \* либо array(0, -<код ошибки>) в случае ошибки  
 \*/* **public** String[] get\_status(**int** id, String phone, **int** all)  
 {  
 String[] m = {};  
 String tmp;  
  
 **try** {  
 m = \_smsc\_send\_cmd(**"status"**, **"phone="** + URLEncoder.*encode*(phone, **SMSC\_CHARSET**) + **"&id="** + id + **"&all="** + all);  
  
 **if** (m.**length** > 1) {  
 **if** (**SMSC\_DEBUG**) {  
 **if** (m[1] != **""** && Integer.*parseInt*(m[1]) >= 0) {  
 java.sql.Timestamp currentTimestamp = **new** java.sql.Timestamp(Integer.*parseInt*(m[1]));  
 System.***out***.println(**"Статус SMS = "** + m[0]);  
 }  
 **else** System.***out***.println(**"Ошибка №"** + Math.*abs*(Integer.*parseInt*(m[1])));  
 }  
  
 **if** (all == 1 && m.**length** > 9 && (m.**length** < 14 || m[14] != **"HLR"**)) {  
 tmp = *\_implode*(m, **","**);  
 m = tmp.split(**","**, 9);  
 }  
 }  
 **else** System.***out***.println(**"Не получен ответ от сервера."**);  
  
 }  
 **catch** (UnsupportedEncodingException e) {  
  
 }  
  
 **return** m;  
 }  
  
 */\*\*  
 \* Получениe баланса  
 \*  
 \** ***@return*** *String баланс или пустую строку в случае ошибки  
 \*/* **public** String get\_balance() {  
 String[] m = {};  
  
 m = \_smsc\_send\_cmd(**"balance"**, **""**); *// (balance) или (0, -error)* **if** (m.**length** > 1) {  
 **if** (**SMSC\_DEBUG**) {  
 **if** (m.**length** == 1)  
 System.***out***.println(**"Сумма на счете: "** + m[0]);  
 **else** System.***out***.println(**"Ошибка №"** + Math.*abs*(Integer.*parseInt*(m[1])));  
 }  
 }  
 **else** {  
 System.***out***.println(**"Не получен ответ от сервера."**);  
 }  
 **return** m.**length** == 2 ? **""** : m[0];  
 }  
  
 */\*\*  
 \* Формирование и отправка запроса  
 \** ***@param cmd*** *- требуемая команда  
 \** ***@param arg*** *- дополнительные параметры  
 \*/* **private** String[] \_smsc\_send\_cmd(String cmd, String arg){  
 */\* String[] m = {}; \*/* String ret = **","**;  
  
 **try** {  
 String \_url = (**SMSC\_HTTPS** ? **"https"** : **"http"**) + **"://smsc.ru/sys/"** + cmd +**".php?login="** + URLEncoder.*encode*(**SMSC\_LOGIN**, **SMSC\_CHARSET**)  
 + **"&psw="** + URLEncoder.*encode*(**SMSC\_PASSWORD**, **SMSC\_CHARSET**)  
 + **"&fmt=1&charset="** + **SMSC\_CHARSET** + **"&"** + arg;  
  
 String url = \_url;  
 **int** i = 0;  
 **do** {  
 **if** (i++ > 0) {  
 url = \_url;  
 url = url.replace(**"://smsc.ru/"**, **"://www"** + (i) + **".smsc.ru/"**);  
 }  
 ret = \_smsc\_read\_url(url);  
 }  
 **while** (ret == **""** && i < 5);  
 }  
 **catch** (UnsupportedEncodingException e) {  
  
 }  
  
 **return** ret.split(**","**);  
 }  
  
 */\*\*  
 \* Чтение URL  
 \** ***@param url*** *- ID cообщения  
 \** ***@return*** *line - ответ сервера  
 \*/* **private** String \_smsc\_read\_url(String url) {  
  
 String line = **""**, real\_url = url;  
 String[] param = {};  
 **boolean** is\_post = (**SMSC\_POST** || url.length() > 2000);  
  
 **if** (is\_post) {  
 param = url.split(**"\\?"**,2);  
 real\_url = param[0];  
 }  
  
 **try** {  
 URL u = **new** URL(real\_url);  
 InputStream is;  
  
 **if** (is\_post){  
 URLConnection conn = u.openConnection();  
 conn.setDoOutput(**true**);  
 OutputStreamWriter os = **new** OutputStreamWriter(conn.getOutputStream(), **SMSC\_CHARSET**);  
 os.write(param[1]);  
 os.flush();  
 os.close();  
 System.***out***.println(**"post"**);  
 is = conn.getInputStream();  
 }  
 **else** {  
 is = u.openStream();  
 }  
  
 InputStreamReader reader = **new** InputStreamReader(is, **SMSC\_CHARSET**);  
  
 **int** ch;  
 **while** ((ch = reader.read()) != -1) {  
 line += (**char**)ch;  
 }  
  
 reader.close();  
 }  
 **catch** (MalformedURLException e) { *// Неверно урл, протокол...* }  
 **catch** (IOException e) {  
  
 }  
  
 **return** line;  
 }  
  
 **private static** String \_implode(String[] ary, String delim) {  
 String out = **""**;  
  
 **for** (**int** i = 0; i < ary.**length**; i++) {  
 **if** (i != 0)  
 out += delim;  
 out += ary[i];  
 }  
  
 **return** out;  
 }  
}  
  
*// Examples:  
/\*  
 Smsc sd= new Smsc();  
 // or  
 Smsc sd= new Smsc("login", "password");  
  
 sd.send\_sms("79999999999", "Ваш пароль: 123", 1, "", "", 0, "", "");  
 sd.get\_sms\_cost("79999999999", "Вы успешно зарегистрированы!", 0, 0, "", "");  
 sd.get\_status(sms\_id, "79999999999");  
 sd.get\_balanse();  
\*/*