Разработка программы для проведения шифрования и дешифрования текста при помощи ключа.

Выполнил: Соловей Г.И.
Руководитель: Атурина В.А.

Санкт-Петербург, 2019

Введение

Во время прохождения практики на тему «Шифрование и дешифрование с использованием ключа» были рассмотрены следующие этапы:

1. Постановка цели.
2. Формирование шагов к созданию.
3. Выбор механизма шифрования.
4. Произведение реализации продукта.
5. Выполнение тестирования программы.
6. Исправление ошибок продукта.

Цели

Целью практики являлась разработка программы шифрования, которая:

1. Выполняет шифрование и дешифрование при помощи ключа;
2. Является законченным приложением со скрытыми формулами и открытыми полями ввода;
3. Учитывает особенности ввода данных во избежание ошибок;

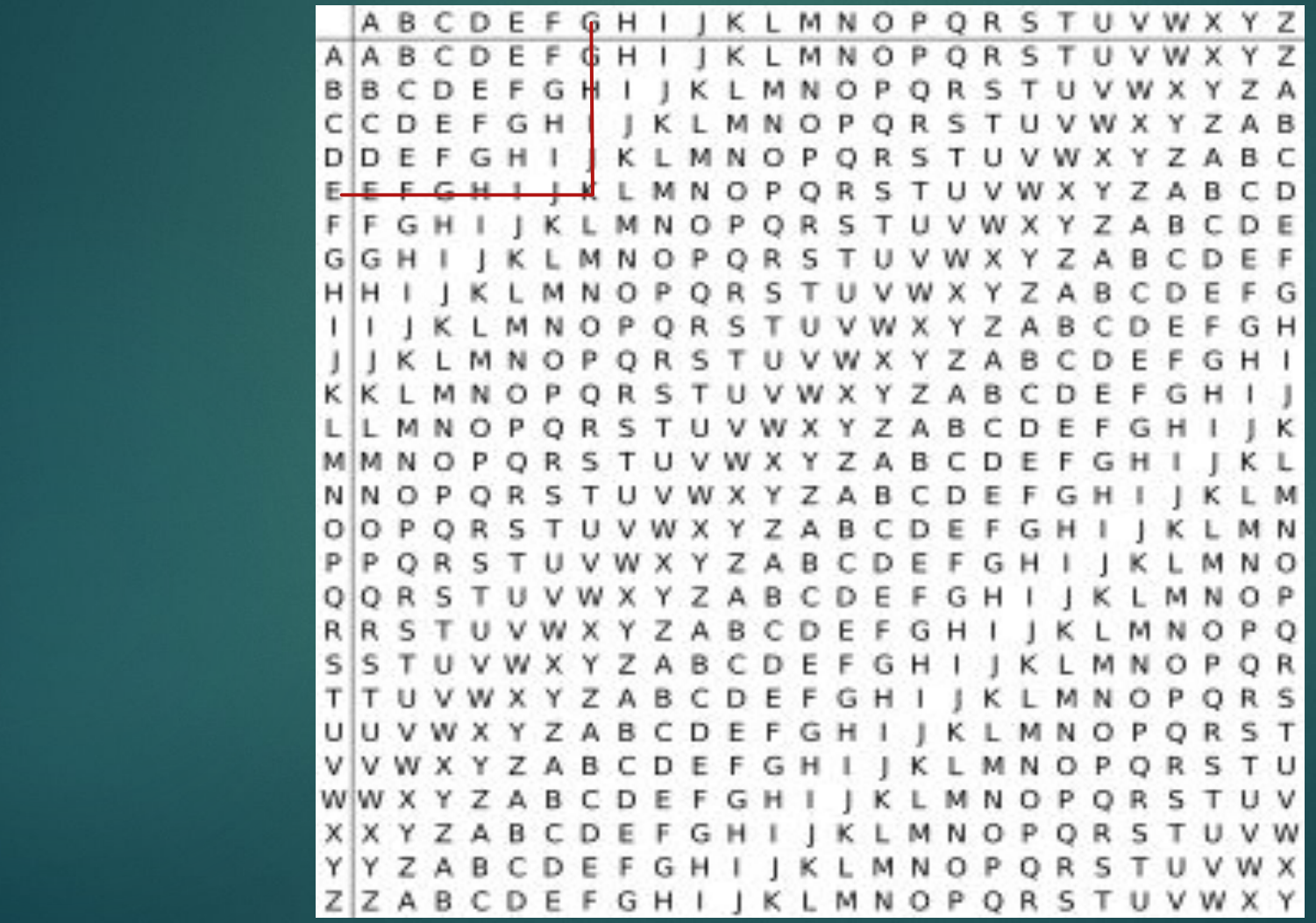
Выбор метода шифрования

Для реализации данного продукта были выбраны три наиболее подходящих метода шифрования:

1. Шифр Виженера;
2. Шифр Цезаря;
3. Шифр Полибия;

Выбор метода шифрования

1.Шифр Виженера

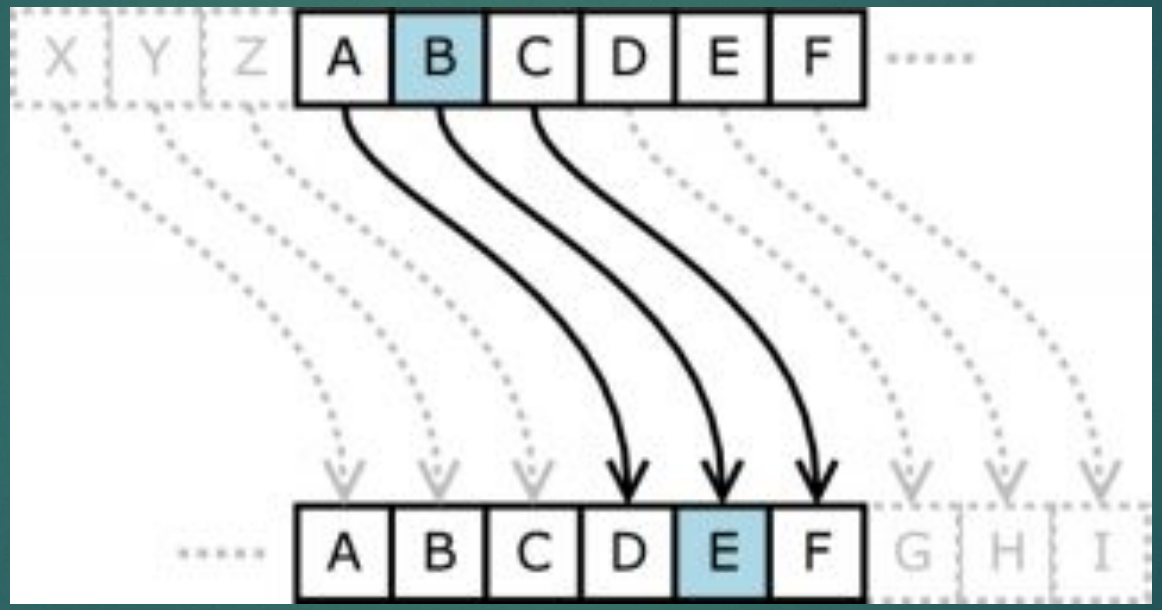


	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

6(G) + 4(E) = 10(K)

Выбор метода шифрования

2.Шифр Цезаря



Выбор метода шифрования

3.Шифр Полибия

БУКВА	К	О	Т
ВЕРТ. КООРДИНАТА	1	5	2
ГОРИЗ. КООРДИНАТА	3	3	4

ВЕРТ. КООРДИНАТА	1	2	3
ГОРИЗ. КООРДИНАТА	5	3	4
БУКВА	Ш	Л	У

	1	2	3	4	5	6
1	Ц	Е	З	А	Р	Ь
2	Б	В	Г	Д	Ж	И
3	К	Л	М	Н	О	П
4	С	Т	У	Ф	Х	Ч
5	Ш	Щ	Ъ	Ы	Э	Ю
6	Я	-	-	-	-	-

Реализация

Исходный текст может содержать только строчные буквы кириллицы (исключая ё), арабские цифры, пробелы, точки и запятые.

Ключ: по длине <= тексту.

Зашифровать

Дешифровать

Зашифрованный текст:

Дешифрованный текст:

Реализация

1.1

```
function varemptier()
{
    userkey_number = [];
    startertext = [];
    userkey = [];
    caesarkey = 0;
    oneinkey = 0;
    zeroinkey = 0;
    cipheredtextfirst = [];
    cipheredtextsecond = [];
    cipheredtextthird = [];
    decipheredtext = [];
    keyfortext = [];
    keyfortext_number = [];
    keybinary = [];
    table = [[],[],[],[],[],[],[],[],[]];
    keyforpolibius = [];
}
```

1.2

```
function start()
{
    startertext = document.getElementById('usertext').value;
    userkey = document.getElementById('key').value;
    if (userkey.length == 0)
    {
        alert('Введите ключ!');
        return 0;
    }
    if (userkey.length > startertext.length)
    {
        alert('Некорректный ключ');
        return 0;
    }
    if (userkey.length == startertext.length)
    {
        keyfortext = userkey;
    }
    if (userkey.length < startertext.length)
    {
        keyfortext = userkey;
        for (i = keyfortext.length; keyfortext.length != startertext.length; i++)
        {
            keyfortext += keyfortext[i - userkey.length];
        }
    }
    for(i = 0; i < keyfortext.length; i++)
    {
        for(j = 0; j < symbolalphabet.length; j++)
        {
            if(keyfortext[i] == symbolalphabet[j])
            {
                keyfortext_number[i] = j;
            }
        }
    }
    for(i = 0; i < userkey.length; i++)
    {
        for(j = 0; j < symbolalphabet.length; j++)
        {
            if(userkey[i] == symbolalphabet[j])
            {
                userkey_number[i] = j;
            }
        }
    }
    firstmethodcipher_vigener();
    secondmethodcipher_caesar();
    polibiustable();
    thirdmethodcipher_polibius();
    document.getElementById('cipheredtext').innerHTML = cipheredtextthird.join('');
}
```

Реализация

2.1

```
function firstmethodcipher_vigener()
{
    for (i = 0; i < startertext.length; i++)
    {
        for(j = 0; j < symbolalphabet.length; j++)
        {
            if (startertext[i] == symbolalphabet[j])
            {
                cipheredtextfirst[i] = j;
            }
        }
    }
    for(i = 0; i < cipheredtextfirst.length; i++)
    {
        cipheredtextfirst[i] += keyfortext_number[i];
    }
    for(i = 0; i < cipheredtextfirst.length; i++)
    {
        for(j = 0; j < symbolalphabet.length; j++)
        {
            if(cipheredtextfirst[i] > 76)
            {
                cipheredtextfirst[i] = cipheredtextfirst[i] - 77;
            }
        }
    }
}
```

2.2

```
function secondmethodcipher_caesar()
{
    for(i = 0; i < userkey_number.length; i++)
    {
        keybinary[i] = userkey_number[i].toString(2);
    }
    oneinkey = 0;
    zeroinkey = 0;
    for (i = 0; i < keybinary.length; i++)
    {
        for (j = 0; j < keybinary[i].length; j++)
        {
            if (keybinary[i][j] == '1') {oneinkey++;}
            if (keybinary[i][j] == '0') {zeroinkey++;}
        }
    }
    caesarkey = oneinkey*zeroinkey;
    while(caesarkey - 77 > 0) {caesarkey = caesarkey - 77;}
    for(i = 0; i < cipheredtextfirst.length; i++)
    {
        cipheredtextsecond[i] = cipheredtextfirst[i] + caesarkey;
        if (cipheredtextsecond[i] > 76) {cipheredtextsecond[i] -= 77;}
    }
}
```

Реализация

2.3

```
function polibiustable()
{
    var alphcopy = [];
    alphcopy = symbolalphabet;
    if(userkey.length == 9 || userkey.length < 9) {keyforpolibius = userkey;}
    if(userkey.length > 9) {for(i = 0; i < 9; i++) {keyforpolibius[i] = userkey[i];}}
    for (i = 0; i < alphcopy.length; i++)
    {
        for(j = 0; j < keyforpolibius.length; j++)
        {
        if(alphcopy[i] == keyforpolibius[j])
            {
                for(k = i; k < alphcopy.length; k++)
                {
                    alphcopy[k] = alphcopy[k+1];
                }
            }
        }
    }
    symbolalphabet = ['А','Б','В','Г','Д','Е','Ж','З','И','Й','К','Л','М','Н','О','П',
    alphcopy.length = symbolalphabet.length - keyforpolibius.length + 1;
    var k = 0;
    for (i = 0; i < keyforpolibius.length; i++)
    {
        table[0][i] = keyforpolibius[i];
    }
    if (keyforpolibius.length == 9)
    {
        for(i = 1; i < 9; i++)
        {
            for(j = 0; j < 9; j++)
            {
                table[i][j] = alphcopy[k]; k++;
            }
        }
    }
    if (keyforpolibius.length < 9)
    {
        for(i = keyforpolibius.length; i < 9; i++)
        {
            table[0][i] = alphcopy[k]; k++;
        }
        for(i = 1; i < 9; i++)
        {
            for(j = 0; j < 9; j++)
            {
                table[i][j] = alphcopy[k]; k++;
            }
        }
    }
}
```

2.4

```
function thirdmethodcipher_polibius()
{
    var verticalcoord = [];
    var horizoncoord = [];
    var cipheredcoords = [];
    var parity;
    var halftext = Math.floor(cipheredtextsecond.length / 2);
    if(cipheredtextthird.length%2 == 0) {parity = true} else {parity = false}
    for(i = 0; i < cipheredtextsecond.length; i++)
    {
        for(j = 0; j < symbolalphabet.length; j++)
        {
            if(cipheredtextsecond[i] == j) {cipheredtextsecond[i] = symbolalphabet[j];}
        }
    }
    for(i = 0; i < cipheredtextsecond.length; i++)
    {
        for(j = 0; j < 9; j++)
        {
            for(k = 0; k < 9; k++)
            {
                if(cipheredtextsecond[i] == table[j][k])
                {
                    verticalcoord[i] = j; horizoncoord[i] = k;
                }
            }
        }
    }
    for(i = 0; i < verticalcoord.length; i++)
    {
        verticalcoord[i] = verticalcoord[i].toString(10);
    }
    for(i = 0; i < horizoncoord.length; i++)
    {
        horizoncoord[i] = horizoncoord[i].toString(10);
    }
    var q = 0;
    for(i = 0; i < cipheredtextsecond.length; i++)
    {
        cipheredcoords += verticalcoord[i];
    }
    for(i = 0; i < cipheredtextsecond.length; i++)
    {
        cipheredcoords += horizoncoord[i];
    }
    var cipheredcoordscopy = [];
    var d = 0;
    for(i = 0; i < cipheredtextsecond.length * 2; i+= 2)
    {
        cipheredcoordscopy[d] = cipheredcoords[i] + cipheredcoords[i+1]; d++;
    }
    cipheredcoords = cipheredcoordscopy;
    for(i = 0; i < cipheredcoords.length; i++)
    {
        cipheredtextthird[i] = table[cipheredcoords[i][0]][cipheredcoords[i][1]];
    }
}
```

Реализация

3.

```
function deciphering()
{
    var verticalcoord = [];
    var horizoncoord = [];
    var cipheredcoords = [];
    var parity;
    var halftext = Math.floor(cipheredtextsecond.length / 2);
    if(cipheredtextthird.length%2 == 0) {parity = true} else {parity = false}
    for(i = 0; i < cipheredtextthird.length; i++)
    {
        for(j = 0; j < 9; j++)
        {
            for(k = 0; k < 9; k++)
            {
                if(cipheredtextthird[i] == table[j][k])
                {
                    cipheredcoords[i] = j.toString(10) + k.toString(10);
                }
            }
        }
    }
    if (parity == true)
    {
        for(i = 0; i < halftext; i++)
        {
            verticalcoord += cipheredcoords[i];
            horizoncoord += cipheredcoords[i+halftext];
        }
    }
    if (parity == false)
    {
        for(i = 0; i < halftext; i++)
        {
            verticalcoord += cipheredcoords[i];
        }
        verticalcoord += cipheredcoords[halftext][0];
        horizoncoord += cipheredcoords[halftext][1];
        for(i = 1; i < halftext+1; i++)
        {
            horizoncoord += cipheredcoords[i+halftext];
        }
    }
    for(i = 0; i < cipheredtextthird.length; i++)
    {
        decipheredtext[i] = table[verticalcoord[i]][horizoncoord[i]];
    }
    for(i = 0; i < decipheredtext.length; i++)
    {
        for(j = 0;  j < symbolalphabet.length; j++)
        {
            if(decipheredtext[i] == symbolalphabet[j]) {decipheredtext[i] = j;}
        }
    }
    for(i = 0; i < decipheredtext.length; i++)
    {
        if(decipheredtext[i] - caesarkey < 0) {decipheredtext[i] += 77}
        decipheredtext[i] -= caesarkey
    }
    for(i = 0; i < decipheredtext.length; i++)
    {
        if (decipheredtext[i] - keyfortext_number[i] < 0) {decipheredtext[i] += 77;}
        decipheredtext[i] -= keyfortext_number[i];
    }
    for (i = 0; i < decipheredtext.length; i++)
    {
        for(j = 0; j < symbolalphabet.length; j++)
        {
            if(decipheredtext[i] == j) {decipheredtext[i] = symbolalphabet[j];}
        }
    };
    document.getElementById('decipheredtext').innerHTML = decipheredtext.join('');
}
```

Тестирование и исправление ошибок

```
function polibiustable()
{
    var alphcopy = [];
    alphcopy = symbolalphabet;
    if(userkey.length == 9 || userkey.length < 9) {keyforpolibius = userkey;}
    if(userkey.length > 9) {for(i = 0; i < 9; i++) {keyforpolibius[i] = userkey[i];}}
    for (i = 0; i < alphcopy.length; i++)
    {
        for(j = 0; j < keyforpolibius.length; j++)
        {
        if(alphcopy[i] == keyforpolibius[j])
            {
                for(k = i; k < alphcopy.length; k++)
                {
                    alphcopy[k] = alphcopy[k+1];


        }
    }
    symbolalphabet = ['А','Б','В','Г','Д','Е','Ж','З','И','Й','К','Л','М','Н','О','П','Р','
    alphcopy.length = symbolalphabet.length - keyforpolibius.length + 1;
    var k = 0;
    for (i = 0; i < keyforpolibius.length; i++)
    {
        table[0][i] = keyforpolibius[i];
    }
```

Заключение

В процессе прохождения практики были получены следующие результаты:

1. Поставлена цель.
2. Сформированы шаги к созданию.
3. Выбран механизм шифрования.
4. Произведена реализация продукта.
5. Выполнено тестирование программы и исправление ошибок.

Спасибо за внимание!