УНИВЕРСИТЕТ ИТМО

# Разработка программной для шифрования и дешифрования текста

Автор: Григорьев В. Е.

Руководитель: Атурина В.А, Меленчук М.А

Санкт-Петербург, 2019

# Введение

Во время прохождения практики на тему «Шифрование и дешифрование матрицы с использованием ключа, размер которого может быть меньше, чем шифруемый текст» были рассмотрены следующие этапы:

- Постановка цели и задач.
- Формирование шагов к созданию.
- Выбор механизма шифрования.
- Проектирование модели разработки.
- Произведение реализации продукта.
- Выполнения тестирования программы.
- Совершения отладки продукта.

# Цели и задачи

Целью практики является разработаться систему шифрование удовлетворяющею следующим требованиям:

- Шифрование и дешифрование выполнять с использованием ключа.
- Задача должна быть реализована как законченное приложение со скрытыми формулами и открытыми полями ввода.
- При реализации учитывать особенности ввода данных так чтобы избежать переполнения или ошибок ввода.

# Шифр Виженера

Шифр Виженера — это последовательность шифров Цезаря с различными значениями сдвига. То есть к первой букве текста применяется преобразование, например, ROT5, ко второй, например, ROT17, и так далее. Последовательность применяемых преобразований определяется ключевой фразой, в которой каждая буква слова обозначает требуемый сдвиг, например, фраза ГДЕ ОН задает такую последовательность шифров Цезаря: ROT3-ROT4-ROT5-ROT15-ROT14, которая повторяется, пока не будет зашифрован весь текст сообщения.

Буквы исходного текста

| Буквы ключа | А | Б | В | Г | Д | Е | Ж | З | И | Й | К | Л | М | Н | О | П | Р | С | Т | У | Ф | Х | Ц | Ч | Ш | Щ | Ъ | Ы | Ь | Э | Ю | Я |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| А | Б | В | Г | Д | Е | Ж | З | И | Й | К | Л | М | Н | О | П | Р | С | Т | У | Ф | Х | Ц | Ч | Ш | Щ | Ъ | Ы | Ь | Э | Ю | Я | А |
| Б | В | Г | Д | Е | Ж | З | И | Й | К | Л | М | Н | О | П | Р | С | Т | У | Ф | Х | Ц | Ч | Ш | Щ | Ъ | Ы | Ь | Э | Ю | Я | А | Б |
| В | Г | Д | Е | Ж | З | И | Й | К | Л | М | Н | О | П | Р | С | Т | У | Ф | Х | Ц | Ч | Ш | Щ | Ъ | Ы | Ь | Э | Ю | Я | А | Б | В |
| Г | Д | Е | Ж | З | И | Й | К | Л | М | Н | О | П | Р | С | Т | У | Ф | Х | Ц | Ч | Ш | Щ | Ъ | Ы | Ь | Э | Ю | Я | А | Б | В | Г |
| Д | Е | Ж | З | И | Й | К | Л | М | Н | О | П | Р | С | Т | У | Ф | Х | Ц | Ч | Ш | Щ | Ъ | Ы | Ь | Э | Ю | Я | А | Б | В | Г | Д |
| Е | Ж | З | И | Й | К | Л | М | Н | О | П | Р | С | Т | У | Ф | Х | Ц | Ч | Ш | Щ | Ъ | Ы | Ь | Э | Ю | Я | А | Б | В | Г | Д | Е |
| Ж | З | И | Й | К | Л | М | Н | О | П | Р | С | Т | У | Ф | Х | Ц | Ч | Ш | Щ | Ъ | Ы | Ь | Э | Ю | Я | А | Б | В | Г | Д | Е | Ж |
| З | И | Й | К | Л | М | Н | О | П | Р | С | Т | У | Ф | Х | Ц | Ч | Ш | Щ | Ъ | Ы | Ь | Э | Ю | Я | А | Б | В | Г | Д | Е | Ж | З |
| И | Й | К | Л | М | Н | О | П | Р | С | Т | У | Ф | Х | Ц | Ч | Ш | Щ | Ъ | Ы | Ь | Э | Ю | Я | А | Б | В | Г | Д | Е | Ж | З | И |
| Й | К | Л | М | Н | О | П | Р | С | Т | У | Ф | Х | Ц | Ч | Ш | Щ | Ъ | Ы | Ь | Э | Ю | Я | А | Б | В | Г | Д | Е | Ж | З | И | Й |
| К | Л | М | Н | О | П | Р | С | Т | У | Ф | Х | Ц | Ч | Ш | Щ | Ъ | Ы | Ь | Э | Ю | Я | А | Б | В | Г | Д | Е | Ж | З | И | Й | К |
| Л | М | Н | О | П | Р | С | Т | У | Ф | Х | Ц | Ч | Ш | Щ | Ъ | Ы | Ь | Э | Ю | Я | А | Б | В | Г | Д | Е | Ж | З | И | Й | К | Л |
| М | Н | О | П | Р | С | Т | У | Ф | Х | Ц | Ч | Ш | Щ | Ъ | Ы | Ь | Э | Ю | Я | А | Б | В | Г | Д | Е | Ж | З | И | Й | К | Л | М |
| Н | О | П | Р | С | Т | У | Ф | Х | Ц | Ч | Ш | Щ | Ъ | Ы | Ь | Э | Ю | Я | А | Б | В | Г | Д | Е | Ж | З | И | Й | К | Л | М | Н |
| О | П | Р | С | Т | У | Ф | Х | Ц | Ч | Ш | Щ | Ъ | Ы | Ь | Э | Ю | Я | А | Б | В | Г | Д | Е | Ж | З | И | Й | К | Л | М | Н | О |
| П | Р | С | Т | У | Ф | Х | Ц | Ч | Ш | Щ | Ъ | Ы | Ь | Э | Ю | Я | А | Б | В | Г | Д | Е | Ж | З | И | Й | К | Л | М | Н | О | П |
| Р | С | Т | У | Ф | Х | Ц | Ч | Ш | Щ | Ъ | Ы | Ь | Э | Ю | Я | А | Б | В | Г | Д | Е | Ж | З | И | Й | К | Л | М | Н | О | П | Р |
| С | Т | У | Ф | Х | Ц | Ч | Ш | Щ | Ъ | Ы | Ь | Э | Ю | Я | А | Б | В | Г | Д | Е | Ж | З | И | Й | К | Л | М | Н | О | П | Р | С |
| Т | У | Ф | Х | Ц | Ч | Ш | Щ | Ъ | Ы | Ь | Э | Ю | Я | А | Б | В | Г | Д | Е | Ж | З | И | Й | К | Л | М | Н | О | П | Р | С | Т |
| У | Ф | Х | Ц | Ч | Ш | Щ | Ъ | Ы | Ь | Э | Ю | Я | А | Б | В | Г | Д | Е | Ж | З | И | Й | К | Л | М | Н | О | П | Р | С | Т | У |
| Ф | Х | Ц | Ч | Ш | Щ | Ъ | Ы | Ь | Э | Ю | Я | А | Б | В | Г | Д | Е | Ж | З | И | Й | К | Л | М | Н | О | П | Р | С | Т | У | Ф |
| Х | Ц | Ч | Ш | Щ | Ъ | Ы | Ь | Э | Ю | Я | А | Б | В | Г | Д | Е | Ж | З | И | Й | К | Л | М | Н | О | П | Р | С | Т | У | Ф | Х |
| Ц | Ч | Ш | Щ | Ъ | Ы | Ь | Э | Ю | Я | А | Б | В | Г | Д | Е | Ж | З | И | Й | К | Л | М | Н | О | П | Р | С | Т | У | Ф | Х | Ц |
| Ч | Ш | Щ | Ъ | Ы | Ь | Э | Ю | Я | А | Б | В | Г | Д | Е | Ж | З | И | Й | К | Л | М | Н | О | П | Р | С | Т | У | Ф | Х | Ц | Ч |
| Ш | Щ | Ъ | Ы | Ь | Э | Ю | Я | А | Б | В | Г | Д | Е | Ж | З | И | Й | К | Л | М | Н | О | П | Р | С | Т | У | Ф | Х | Ц | Ч | Ш |
| Щ | Ъ | Ы | Ь | Э | Ю | Я | А | Б | В | Г | Д | Е | Ж | З | И | Й | К | Л | М | Н | О | П | Р | С | Т | У | Ф | Х | Ц | Ч | Ш | Щ |
| Ъ | Ы | Ь | Э | Ю | Я | А | Б | В | Г | Д | Е | Ж | З | И | Й | К | Л | М | Н | О | П | Р | С | Т | У | Ф | Х | Ц | Ч | Ш | Щ | Ъ |
| Ы | Ь | Э | Ю | Я | А | Б | В | Г | Д | Е | Ж | З | И | Й | К | Л | М | Н | О | П | Р | С | Т | У | Ф | Х | Ц | Ч | Ш | Щ | Ъ | Ы |
| Ь | Э | Ю | Я | А | Б | В | Г | Д | Е | Ж | З | И | Й | К | Л | М | Н | О | П | Р | С | Т | У | Ф | Х | Ц | Ч | Ш | Щ | Ъ | Ы | Ь |
| Э | Ю | Я | А | Б | В | Г | Д | Е | Ж | З | И | Й | К | Л | М | Н | О | П | Р | С | Т | У | Ф | Х | Ц | Ч | Ш | Щ | Ъ | Ы | Ь | Э |
| Ю | Я | А | Б | В | Г | Д | Е | Ж | З | И | Й | К | Л | М | Н | О | П | Р | С | Т | У | Ф | Х | Ц | Ч | Ш | Щ | Ъ | Ы | Ь | Э | Ю |
| Я | А | Б | В | Г | Д | Е | Ж | З | И | Й | К | Л | М | Н | О | П | Р | С | Т | У | Ф | Х | Ц | Ч | Ш | Щ | Ъ | Ы | Ь | Э | Ю | Я |

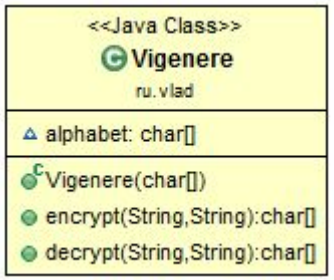
# Шифр Вернама

Шифр является разновидностью криптосистемы одноразовых блокнотов. В нём используется булева функция «Исключающее ИЛИ». Шифр Вернама является примером системы с абсолютной криптографической стойкостью. При этом он считается одной из простейших криптосистем.

Для получения шифротекста открытый текст объединяется операцией «исключающее ИЛИ» с секретным ключом. Так, например, при применении ключа (11101) на букву «А» (11000) получаем зашифрованное сообщение (00101). Зная, что для принимаемого сообщения имеем ключ (11101), легко получить исходное сообщение той же операцией.

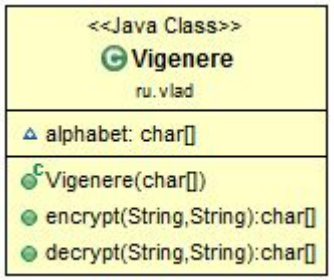
# Реализация (1/5)

```
for(int i = 0; i < data.length; i++) {
        x = -1;
        y = -1;
        f = -1;
        if(data[i] == ' ') {
                code[i] = ' ';
                continue;
        }
        for(int j = 0; j < alphabet.length; j++) {
                if(data[i] == alphabet[j]) {
                        x = j;
                }
        }
        if(x == -1) {
                code[i] = data[i];
                continue;
        }
        hide[i] = key[k];
        k++;
        k = (k > key.length - 1) ? 0 : k;
        for(int j = 0; j < alphabet.length; j++) {
                if(hide[i] == alphabet[j]) {
                        y = j;
                }
        }
        f = (x + y >= alphabet.length) ? x + y - alphabet.length : x + y;
        code[i] = alphabet[f];
}
return code;
```



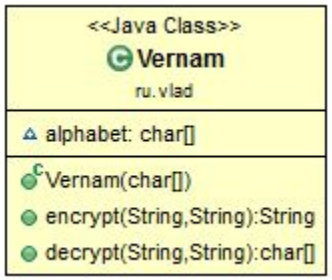
# Реализация (2/5)

```
for(int i = 0; i < code.length; i++) {
        x = -1;
        y = -1;
        f = -1;
        if(code[i] == ' ') {
                data[i] = ' ';
                continue;
        }
        for(int j = 0; j < alphabet.length; j++) {
                if(code[i] == alphabet[j]) {
                        f = j;
                }
        }
        if(f == -1) {
                data[i] = code[i];
                continue;
        }
        hide[i] = key[k];
        k++;
        k = (k > key.length - 1) ? 0 : k;
        for(int j = 0; j < alphabet.length; j++) {
                if(hide[i] == alphabet[j]) {
                        y = j;
                }
        }
        x = (f - y < 0) ? alphabet.length - (y - f) : f - y;
        data[i] = alphabet[x];
}
return data;
```



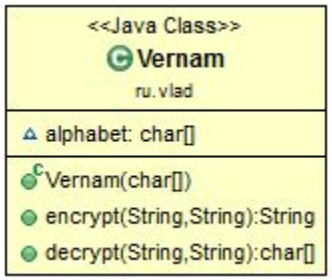
# Реализация (3/5)

```
for(int i = 0; i < data.length; i++) {
        a = -1;
        b = -1;
        c = -1;
        if(data[i] == ' ') {
                code += ' ';
                continue;
        }
        for(int j = 0; j < alphabet.length; j++) {
                if(data[i] == alphabet[j]) {
                        a = j;
                }
        }
        if(a == -1) {
                code += data[i];
                continue;
        }
        hide[i] = key[k];
        k++;
        k = (k > key.length - 1) ? 0 : k;
        for(int j = 0; j < alphabet.length; j++) {
                if(hide[i] == alphabet[j]) {
                        b = j;
                }
        }
    c = a ^ b;
    char[] convert = Integer.toBinaryString(c).toCharArray();
    char[] letterBin = {'0', '0', '0', '0', '0', '0'};
    int d = 0;
        for(int j = letterBin.length - convert.length; j < letterBin.length; j++) {
            letterBin[j] = convert[d];
                d++;
}
    code += new String(letterBin);
}
return code;
```



# Реализация (4/5)

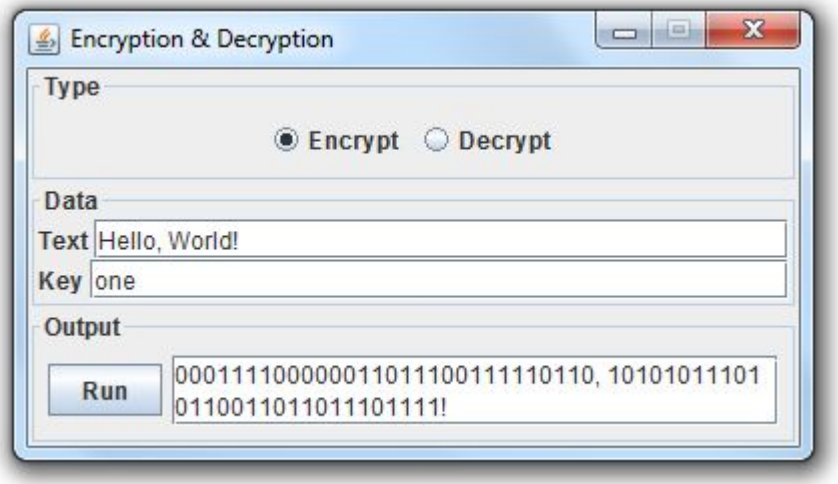
```
for(int i = 0; i < code.length; i++) {
bin = "";
        a = -1;
        b = -1;
        c = -1;
        if(code[i] == ' ') {
                data[index] = ' ';
        index++;
                continue;
        }
        if(code[i] != '1' && code[i] != '0') {
        data[index] = code[i];
        index++;
          continue;
}
        for(int j = i; j < i + 6; j++) {
        bin += code[j];
}
c = Integer.parseInt(bin, 2);
hide[i] = key[k];
        k++;
        k = (k > key.length - 1) ? 0 : k;
for(int j = 0; j < alphabet.length; j++) {
                if(hide[i] == alphabet[j]) {
                        b = j;
                }
        }
a = c ^ b;
        data[index] = alphabet[a];
index++;
i+=5;
}
return data;
```



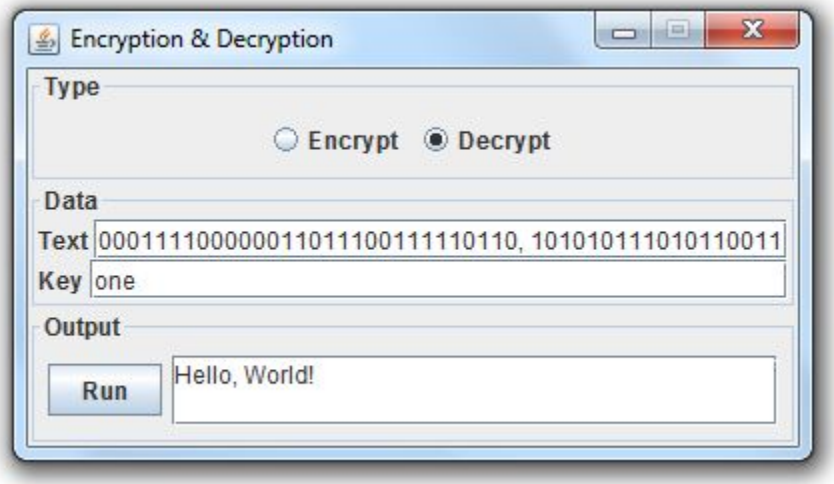
# Реализация (5/5)

```
panel.setLayout(new BoxLayout(panel, BoxLayout.Y_AXIS));
JPanel type = new JPanel();
Border borderType = BorderFactory.createTitledBorder("Type");
type.setBorder(borderType);
ButtonGroup typeButton = new ButtonGroup();
typeButton.add(encrypt);
typeButton.add(decrypt);
encrypt.setSelected(true);
type.add(encrypt);
type.add(decrypt);
panel.add(type);
JPanel dataFields = new JPanel();
dataFields.setLayout(new BoxLayout(dataFields, BoxLayout.Y_AXIS));
Border borderData = BorderFactory.createTitledBorder("Data");
dataFields.setBorder(borderData);
JPanel textField = new JPanel();
textField.setLayout(new BoxLayout(textField, BoxLayout.X_AXIS));
textField.add(label1);
textField.add(data);
JPanel keyField = new JPanel();
keyField.setLayout(new BoxLayout(keyField, BoxLayout.X_AXIS));
keyField.add(label2);
keyField.add(key);
dataFields.add(textField);
dataFields.add(keyField);
panel.add(dataFields);
ActionListener actionListener = new TestActionListener();
run.addActionListener(actionListener);
JPanel outPut = new JPanel();
outPut.setLayout(new FlowLayout());
Border borderOutPut = BorderFactory.createTitledBorder("Output");
outPut.setBorder(borderOutPut);
outPut.add(run);
area2.setLineWrap(true);
area2.setEditable(false);
outPut.add(new JScrollPane(area2));
panel.add(outPut);
frame.setTitle("Encryption & Decryption");
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame.setLocationRelativeTo(null);
frame.setResizable(false);
frame.add(panel);
frame.pack();
frame.setVisible(true);
```

# Интерфейс пользователя



# Интерфейс пользователя



УНИВЕРСИТЕТ ИТМО

# Спасибо за внимание

Санкт-Петербург, 2019