

Отчет по лабораторной работе по предмету Математические основы защиты информации и информационной безопасности

Лабораторная работа №1. Шифры простой замены

Никита Андреевич Топонен

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
3.1	Шифр Цезаря	7
3.2	Шифр Атбаш	7
4	Выполнение лабораторной работы	8
4.1	Шифр Цезаря	8
4.2	Шифр Атбаш	10
5	Выводы	14
	Список литературы	15

Список иллюстраций

4.1	Шифрование шифром Цезаря	10
4.2	Шифрование шифром Атбаш	13

Список таблиц

1 Цель работы

Цель работы — познакомиться с шифрами простой замены, а также реализовать шифр Цезаря и шифр Атбаш.

2 Задание

1. Реализовать шифр Цезаря с произвольным ключом k .
2. Реализовать шифр Атбаш.

3 Теоретическое введение

3.1 Шифр Цезаря

Шифр Цезаря — это вид шифра подстановки, в котором каждый символ в открытом тексте заменяется символом, находящимся на некотором постоянном числе позиций левее или правее него в алфавите. Например, в шифре со сдвигом вправо на 3, А была бы заменена на Г, Б станет Д, и так далее.

Шифр назван в честь римского полководца Гая Юлия Цезаря, использовавшего его для секретной переписки со своими генералами.

Шифрование с использованием ключа $k = 3$. Буква «А» «сдвигается» на три буквы вперёд и становится буквой «D», буква «Z», перемещённая на три буквы вперёд, становится буквой «B», и так далее.

3.2 Шифр Атбаш

Шифр Атбаш — простой шифр подстановки для алфавитного письма. Правило шифрования состоит в замене i -й буквы алфавита буквой с номером $n - i + 1$, где n — число букв в алфавите.

Впервые встречается в древнееврейском тексте Библии / Танаха.

Таким образом, латинский алфавит с пробелом будет сопоставляться следующим образом: букве «А» соответствует « », буква «В» соответствует букве «Z», буква «Z» соответствует букве «B», и « » соответствует «А».

4 Выполнение лабораторной работы

4.1 Шифр Цезаря

В рамках данной лабораторной работы я реализовал шифрование шифром Цезаря на языке Java для латинского алфавита без учета пробела в алфавите. Ниже приведен код с подробными комментариями:

```
public class Cesar {  
    public static void main(String[] args) {  
        // Подготавливаем сообщение  
        String testMessage = "checking the cesar code on real example";  
  
        // Кодировем сообщение с k = 3  
        String cesarEncodedTestMessage = encode(testMessage, 3);  
  
        // Выводим зашифрованное сообщение для проверки  
        System.out.println("Encoded message: " + cesarEncodedTestMessage);  
  
        // Расшифровываем сообщение  
        String decodedTestMessage = decode(cesarEncodedTestMessage, 3);  
  
        // Проверяем, что расшифрованное сообщение соответствует ожиданиям  
        System.out.println("Decoded message: " + decodedTestMessage);  
    }  
}
```



```

// =====
// Implementation
// =====

// Шифрование сообщения шифром Цезаря с произвольным сдвигом offset
private static String encode(String message, int offset) {
    // Проверяем, что сдвиг не выходит за пределы алфавита
    // Если сдвиг не попадает в рамки алфавита, возвращаем
    // сообщение об ошибке
    if (offset < 1 || offset > 26) {
        return "Could not encode your message. Please check offset.";
    }

    StringBuilder result = new StringBuilder();

    // В цикле шифруем сообщение с помощью сдвига по таблице ASCII
    for (char character : message.toCharArray()) {
        // Пробелы не шифруются
        if (character != ' ') {
            // ASCII код буквы - ASCII код а
            int originalAlphabetPosition = character - 'a';
            // Находим смещение в зависимости от offset
            int newAlphabetPosition =
                (originalAlphabetPosition + offset) % 26;
            // Достаем символ ASCII, прибавляя смещение
            char newCharacter = (char) ('a' + newAlphabetPosition);
            // Записываем в результат
            result.append(newCharacter);
        }
    }
}

```

```

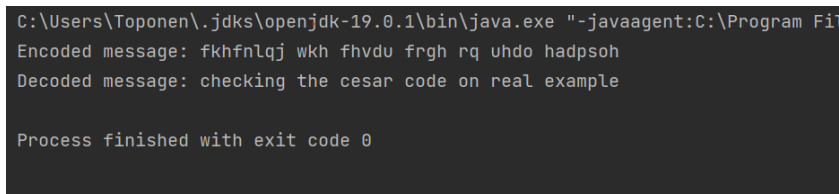
        } else {
            result.append(character);
        }
    }

    return result.toString();
}

// Расшифровываем сообщения, зашифрованное шифром Цезаря,
// с заранее известным offset
private static String decode(String encodedMessage, int offset) {
    // Сдвигаем так, чтобы алфавит оказался в начальном положении
    return encode(encodedMessage, 26 - (offset % 26));
}
}

```

Результаты выполнения программы на иллюстрации (рис. 4.1).



```

C:\Users\Toponen\.jdk\openjdk-19.0.1\bin\java.exe "-javaagent:C:\Program Fi
Encoded message: fkhfnlqj wkh fhvdu frgh rq uhdo hadpsoh
Decoded message: checking the cesar code on real example

Process finished with exit code 0

```

Рис. 4.1: Шифрование шифром Цезаря

4.2 Шифр Атбаш

В рамках данной лабораторной работы я реализовал шифрование шифром Атбаш на языке Java для латинского алфавита с пробелом в алфавите. Ниже приведен код с подробными комментариями:

```

import java.util.HashMap;

public class Atbash {
    public static void main(String[] args) {
        // Подготавливаем сообщение
        String testMessage = "checking the atbash code on real example";

        // Кодировем сообщение
        String atbashEncodedTestMessage = atbash(testMessage);

        // Выводим зашифрованное сообщение для проверки
        System.out.println("Encoded message: " + atbashEncodedTestMessage);

        // Расшифровываем сообщение
        String decodedTestMessage = atbash(atbashEncodedTestMessage);

        // Проверяем, что расшифрованное сообщение соответствует ожиданиям
        System.out.println("Decoded message: " + decodedTestMessage);
    }

    // =====
    // Implementation
    // =====

    // Таблица соответствия латинского алфавита с пробелом с шифром Атбаш
    private static final HashMap<Character, Character> ATBASH_TABLE =
        new HashMap<>(){
            put('a', 'z'); put('b', 'y'); put('c', 'x');
            put('e', 'w'); put('f', 'v'); put('g', 'u'); put('h', 't');
        }
}

```

```

        put('i', 's'); put('j', 'r'); put('k', 'q'); put('l', 'p');
        put('m', 'o'); put('n', 'n'); put('o', 'm'); put('p', 'l');
        put('q', 'k'); put('r', 'j'); put('s', 'i'); put('t', 'h');
        put('u', 'g'); put('v', 'f'); put('w', 'e'); put('x', 'd');
        put('y', 'c'); put('z', 'b'); put(' ', 'a');
    });

    // Шифрование и расшифрование сообщения шифром Атбаш
    public static String atbash(String message)
    {
        StringBuilder result = new StringBuilder();

        // В цикле находим соответствующий символ в таблице
        // и записываем в результат
        for(char letter : message.toCharArray()) {
            result
                .append(Character
                    .toLowerCase(ATBASH_TABLE.get(letter)))
        }

        return result.toString();
    }
}

```

Результаты выполнения программы на иллюстрации (рис. 4.2).

```
C:\Users\Toponen\.jdk\openjdk-19.0.1\bin\java.exe "-javaagent:  
Encoded message: ytwyqsuahtwa hz itaymxwamnajw pawd olpw  
Decoded message: checking the atbash code on real example  
  
Process finished with exit code 0
```

Рис. 4.2: Шифрование шифром Атбаш

5 Выводы

В рамках данной лабораторной работы я познакомился с шифрами простой замены, такими как шифр Цезаря и шифр Атбаш. Также реализовал оба шифра на языке Java.

Список литературы