

Akademia Nauk Stosowanych
Teoretyczne i technologiczne podstawy multimediiów
– IS rok 3

Imię i Nazwisko: Emilian Kochanek

Grupa: L2

Data: 01.11.2022

Symbol: TiTPM_05

Na początku użytkownik podaje wiadomość do zaszyfrowania i na podstawie tej wiadomości tworzony jest słownik podstawowy. Z wiadomości wybierane są znaki, z których jest zbudowana, bez powtórzeń i układane w kolejności alfabetycznej.

```
1 usage  ➤ Emilian Kochanek
public void zapiszWiadomosc(){
    System.out.print("Podaj wiadomosc: ");
    wiadomosc = scanner.nextLine();

    slowinkPodstawowy();
}

1 usage  ➤ Emilian Kochanek
private void slowinkPodstawowy(){
    List<Character> collect = wiadomosc.chars().mapToObj(o -> (char) o).distinct().collect(Collectors.toList());
    Collections.sort(collect);

    for(Character value : collect){
        i++;
        mapaPodstawowa.put(i, value.toString());
    }
    rozszerzSloownik();
}
```

Następnie tak zbudowany słownik wykorzystywany jest do kodowania wiadomości na podstawie występujących znaków w wiadomości oraz przy okazji powiększany o następne patterny. Patterny tworzone są poprzez sklejanie dwóch kolejnych znaków wiadomości, jeżeli taki pattern nie występuje wkładany jest do słownika oraz prany jest indeks pierwszego znaku z patternu, w przeciwnym razie budowany jest nowy pattern i wyciągany odpowiedni indeks dla pasujących znaków nowego patternu.

```

private void rozszerzSloownik(){
    for(int j = 0; j<=wiadomosc.length()-1; j++){
        if(j == wiadomosc.length()-1){
            String lastChar = String.valueOf(wiadomosc.charAt(j));
            List<Integer> lastPattern = mapaPodstawowa.entrySet().stream()
                .filter(v -> v.getValue().equals(lastChar))
                .map(v -> v.getKey()).collect(Collectors.toList());
            encryptedMessage.add(lastPattern.get(0));
        }else {
            pattern += wiadomosc.substring(j, j + 2);
            if (mapaPodstawowa.containsValue(pattern)) {
                int indexToConcat = pattern.length() - 1;
                pattern = pattern.substring(0, indexToConcat);
            } else {
                int index = pattern.length() - 1;
                String shortedMessage = pattern.substring(0, index);
                List<Integer> indexPattern = mapaPodstawowa.entrySet().stream()
                    .filter(v -> v.getValue().equals(shortedMessage))
                    .map(v -> v.getKey()).collect(Collectors.toList());
                encryptedMessage.add(indexPattern.get(0));
                i++;
                mapaPodstawowa.put(i, pattern);

                pattern = "";
            }
        }
    }
    czytajSzyfr();
}

```

Wiadomość jest zlepkiem indeksów odpowiednich patternów. Taki słownik możemy zapisać do pliku oraz go z niego odczytać i wykorzystać do odszyfrowania wiadomości.

```

public void zapiszDoPlikuONazwie() {
    if (mapaPodstawowa.isEmpty()){
        System.out.println("\nSłownik jest pusty\nZapis do pliku przerwany");
        return;
    }
    try{
        System.out.print("\nPodaj nazwe pliku: ");
        String nazwaPliku = scanner.nextLine();
        File file = new File(nazwaPliku);
        if(!file.exists()){
            System.out.println("Plik został utworzony");
            file.createNewFile();
        }
        if(file.canWrite()) {
            FileWriter fileWriter = new FileWriter(file);
            Formatter formatter = new Formatter(fileWriter);

            for (Map.Entry<Integer, String> mapka : mapaPodstawowa.entrySet()){
                formatter.format("%d | %s\r\n", mapka.getKey(), mapka.getValue());
            }
            formatter.close();
            fileWriter.close();
        }
        System.out.println("Plik został zapisany");
    }catch (Exception e){
        System.out.println("Wystąpił problem podczas zapisu do pliku");
        System.out.println(e.getMessage());
    }
}
}

```

```

public void odczytajZPliku(){
    try{
        mapaPodstawowa.clear();
        System.out.print("Podaj nazwe pliku: ");
        String nazwaPliku = scanner.nextLine();
        File file = new File(nazwaPliku);
        String odczytZpliku;
        if (file.exists()){
            Scanner fileScanner = new Scanner(file);
            while(fileScanner.hasNextLine()){
                odczytZpliku = fileScanner.nextLine();
                String[] split = odczytZpliku.split( regex: "[|\\s]");
                int index = Integer.parseInt(split[0].replace( target: " ", replacement: ""));
                String values = split[1].replace( target: " ", replacement: "");
                mapaPodstawowa.put(index, values);
            }
            fileScanner.close();
        }
        else{
            System.out.println("Plik nie istnieje");
        }
    }catch (Exception e){
        System.out.println("Wystąpił błąd podczas odczytu pliku");
    }
}
}

```

```

public void decryptMessage(){
    String decryptedMessage = "";
    for(Integer index : encryptedMessage){
        decryptedMessage += mapaPodstawowa.get(index);
    }
    System.out.println("\nOdszyfrowana wiadomosc: " + decryptedMessage);
}

```

Podaj wiadomosc: wabbawabba

3 1 2 2 1 4 6 1

Podaj nazwe pliku: Słownik

Plik zostal utworzony

Plik zostal zapisany

Podaj nazwe pliku: Słownik

Odszyfrowana wiadomosc: wabbawabba

1		a
2		b
3		w
4		wa
5		ab
6		bb
7		ba
8		aw
9		wab
10		bba