Akademia Nauk Stosowanych Teoretyczne i technologiczne podstawy multimediów – IS rok 3	
Imię i Nazwisko: Emilian Kochanek	**Grupa:** L2
Data: 15.11.2022	**Symbol:** TiTPM_06

Użytkownik podaje treść wiadomości do zaszyfrowania oraz wielkość bufforu dla słownika. Na podstawie podanej wiadomości tworzony jest słownik szyfrujący wiadomość. Na jest to pusty wyraz oznaczony indeksem (0, 0, $pierwszaLiteraWiadomosci).

```
public void zapiszWiadomosc(){
    System.out.print("Podaj wiadmosc: ");
    wiadomosc = scanner.nextLine();

    System.out.println("Podaj wielkość slownika: ");
    wielkoscSlownika = scanner.nextInt();

    slowinkPodstawowy();
}
```

```
private void slowinkPodstawowy() {
    char firstLetter = wiadomosc.charAt(0);

    mapKey = String.format("(0, 0, %s)", firstLetter);
    mapaPodstawowa.put(mapKey, pattern);

    pattern = pattern.substring( beginIndex: 1) + firstLetter;
    rozszerzSlownik();
}
```

Następnie program przystępuje do dalszej budowy słownika, na podstawie algorytmu opisanego w pdf.

```
private void rozszerzSlownik(){
    for(int j = 1; j<=wiadomosc.length()+wielkoscSlownika; j++){
        if(j >= wiadomosc.length()-1){
            pattern = buffor;
            buffor = String.valueOf(wiadomosc.charAt(wiadomosc.length()-1));
            if (conatinsValue(buffor)) {
                mapKey = String.format("(%d, %d, %s)", indexStart, indexEnd, "_");
                mapaPodstawowa.put(mapKey, pattern);
                break;
            }
        }else {
            if(buffor.length() == 0){
                buffor = wiadomosc.substring(j, j+4);
            }else {
                pattern = pattern.substring( beginIndex: indexEnd+1) + buffor.substring(0, indexEnd+1);
                buffor = buffor.substring( beginIndex: indexEnd+1) + wiadomosc.substring(j, j+indexEnd+1);
            }
            mapaPodstawowa.put("#", pattern);
            if (conatinsValue(buffor)){
                mapaPodstawowa.remove( key: "#");
                char firstLetter = buffor.charAt(indexEnd);
                mapKey = String.format("(%d, %d, %s)", indexStart, indexEnd, firstLetter);
                mapaPodstawowa.put(mapKey, pattern);
                if(indexEnd + 1 == wielkoscSlownika){
                    j += indexEnd-1;
                }else {
                    j += indexEnd + 1;
                }
            }
            else {
                char firstLetter = buffor.charAt(0);
                mapKey = String.format("(0, 0, %s)", firstLetter);
                mapaPodstawowa.put(mapKey, pattern);
                pattern = pattern.substring( beginIndex: 1) + firstLetter;
                buffor = "";
            }
        }
```

Po zakończeniu bodowy słownika, użytkownik ma możliwość zapisania go do pliku o podanej przez niego nazwie.

```java
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 = scannerPlik.nextLine();
        File file = new File(nazwaPliku);
        if(!file.exists()){
            System.out.println("Plik zostal utworzony");
            file.createNewFile();
        }
        if(file.canWrite()) {
            FileWriter fileWriter = new FileWriter(file);
            Formatter formatter = new Formatter(fileWriter);

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

Następnie na potrzeby testów słownik jest czyszczony i odczytywany z pliku, który wcześniej został utworzony.

```java
public void odczytajZPliku(){
    try{
        mapaPodstawowa.clear();
        System.out.print("Podaj nazwe pliku: ");
        String nazwaPliku = scannerPlik.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: "[|]");
                String index = 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("Wystapil blad podczas odczytu pliku");
    }
    finally {
        scannerPlik.close();
        scanner.close();
    }
}
```

Tak utworzony ponownie słownik wykorzystywany jest do odszyfrowania zakodowanej wiadomości.

Dekodowanie polega na poprawnym odczytaniu indeksu, który mówi jak od którego znaku wziąć pattern, przypisany do niego i ile wynosi długość tego patternu która doklejana jest do finalnej wiadomości. Podczas budowania wiadomości dodawany jest zawsze znak zawarty w indeksie. Jeżeli znak indeksu to “_”, program kończy budowę wiadomości i wyświetla odkodowaną wiadomość.

```
Podaj wiadmosc: ababcbababaaaaaa
Podaj wielkość slownika:
4
(0, 0, a) (0, 0, b) (2, 2, c) (0, 3, a) (0, 2, a) (2, 2, a) (0, 1, _)
Podaj nazwe pliku: Plik
Plik zostal zapisany
Podaj nazwe pliku: Plik

Odszyfrowna wiadomosc: ababcbababaaaaa
```

```
(0, 0, a) | ____
(0, 0, b) | ___a
(2, 2, c) | __ab
(0, 3, a) | babc
(0, 2, a) | baba
(2, 2, a) | abaa
(0, 1, _) | aaaa
```