

Akademia Nauk Stosowanych

Teoretyczne i technologiczne podstawy multimediiów

– IS rok 3

Imię i Nazwisko: Emilian Kochanek

Grupa: L2

Data: 29.10.2022

Symbol: TiTPM_04

Na początku użytkownik proszony jest o podanie wiadomości do zaszyfrowania. Następnie zliczane są wystąpienia poszczególnych znaków w wiadomości i sortowane w kolejności alfabetycznej.

```
1 usage  Emilian Kochanek
public void start(){
    System.out.print("Podaj wiadomosc do zaszyfrowania: ");
    wiadomosc = strings.nextLine();
    System.out.println(zliczWystapienia(wiadomosc));
}
```

```
1 usage  Emilian Kochanek
private LinkedHashMap<Character, ArrayList<Float>> zliczWystapienia(String ciagZnakow){
    LinkedHashMap<Character, Float> sorted = new LinkedHashMap<>();
    Map<Character, Long> mapStringToCount = ciagZnakow
        .chars().mapToObj(c -> (char) c)
        .collect(Collectors.groupingBy(c -> c, Collectors.counting()));

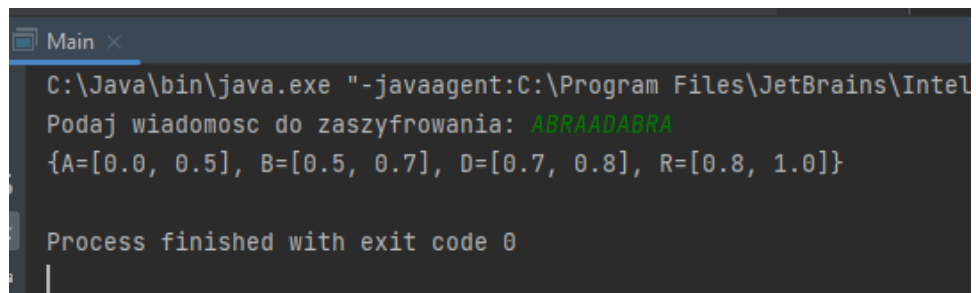
    mapStringToCount.entrySet().stream().sorted(Map.Entry.comparingByKey())
        .forEachOrdered(entry -> sorted.put(entry.getKey(), Float.valueOf(entry.getValue())));

    return wyznaczPrzedzial(sorted);
}
```

Tak wyliczone wystąpienia dla poszczególnych znaków alfabetu przekazywane są do funkcji, która wyznacza przedziały dla tych znaków. W kolejności alfabetycznej sprawdza ilość wystąpień danego znaku i dzieli go przez długość całej wiadomości. Następny znak jest tworzony na tej samej podstawie tylko jego dolną granicą jest górna granica poprzednika, więc ta granica jest dodawana do wyliczonej wartości.

```
private LinkedHashMap<Character, ArrayList<Float>> wyznaczPrzedzial(Map<Character, Float> wystapienia){
    boolean isFirst = true;
    LinkedHashMap<Character, ArrayList<Float>> result = new LinkedHashMap<>();
    for(Map.Entry<Character, Float> iter : wystapienia.entrySet()){
        ArrayList<Float> newPrzedzial = new ArrayList<>();
        if (isFirst){
            newPrzedzial.add((float) 0);
            newPrzedzial.add(iter.getValue()/wiadomosc.length());
            result.put(iter.getKey(), newPrzedzial);
            isFirst=false;
        }else {
            float lastDigit = result.entrySet().stream().sorted(Map.Entry.comparingByKey(Comparator.reverseOrder())).findFirst().get().get(1);
            newPrzedzial.add(lastDigit);
            newPrzedzial.add(lastDigit+iter.getValue()/wiadomosc.length());
            result.put(iter.getKey(), newPrzedzial);
        }
    }
    return result;
}
```

Na koniec program pokazuje wyznaczone granice dla poszczególnych znaków alfabetu podanych w wiadomości do zaszyfrowania.

A screenshot of a Java IDE terminal window. The window title is "Main x". The command prompt shows the execution of a Java program: `C:\Java\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA\bin\idea-agent.jar" Podaj wiadomosc do zaszyfrowania: ABRAADABRA {A=[0.0, 0.5], B=[0.5, 0.7], D=[0.7, 0.8], R=[0.8, 1.0]}`. The output shows the message "Podaj wiadomosc do zaszyfrowania: ABRAADABRA" where the input "ABRAADABRA" is highlighted in green. Below this, the ranges for the alphabet are displayed: `{A=[0.0, 0.5], B=[0.5, 0.7], D=[0.7, 0.8], R=[0.8, 1.0]}`. At the bottom, it says "Process finished with exit code 0".

```

Main x
C:\Java\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA\bin\idea-agent.jar"
Podaj wiadomosc do zaszyfrowania: ABRAADABRA
{A=[0.0, 0.5], B=[0.5, 0.7], D=[0.7, 0.8], R=[0.8, 1.0]}

Process finished with exit code 0
```