

```

1  import java.io.*;
2  import java.util.Scanner;
3
4  public class Main {
5      public static void main(String[] args) throws IOException {
6          metinOzellikleri metinOzellikleri = new metinOzellikleri();
7          Scanner scanner = new Scanner(System.in);
8          int secim;
9          do {
10             System.out.println("1-Metin yaz\2-Metin oku\3-Metnin kelimelerini say\4-Çıkış\Yapmak istediğiniz işlemi seçiniz: ");
11             secim = scanner.nextInt();
12             switch (secim){
13                 case 1:
14                     metinOzellikleri.metinYaz();
15                     break;
16                 case 2:
17                     metinOzellikleri.metinOku();
18                     break;
19                 case 3:
20                     metinOzellikleri.kelimeSayma();
21                     break;
22             }
23         } while (secim != 4);
24     }
25 }

```

Solda görünen resim bizim main fonksiyonumuzdur. Sade ve kalabalıktan uzak, do-while içine switch-case yapısının konduğu bir yapıdır. Yapmak istediğimiz işlemi böylelikle bize seçtirir.

Sağda görünen çıktı 1. ödevin çıktısıdır. Yapmak işlemin seçimine göre size metin yazdırma, yazılan okuma, yazdığınız metni kaydettiğiniz .txt uzantılı içindeki kelimeleri sayma gibi fonksiyonları sunar.

```

1-Metin yaz
2-Metin oku
3-Metnin kelimelerini say
4-Çıkış
Yapmak istediğiniz işlemi seçiniz:
1
Metin giriniz:
Yusuf Asan 1306220078
1-Metin yaz
2-Metin oku
3-Metnin kelimelerini say
4-Çıkış
Yapmak istediğiniz işlemi seçiniz:
2
Yusuf Asan 1306220078

```

istediğiniz metni dosyanın

```
Main.java x metinOzellikleri.java x MetinBelgesi.txt x
1 import java.io.*;
2 import java.util.Scanner;
3
4 public class metinOzellikleri {
5     File file = new File("MetinBelgesi.txt");
6     String metin;
7     public void metinYaz() throws IOException {
8         Scanner girdi = new Scanner(System.in);
9
10        System.out.println("Metin giriniz: ");
11        metin = girdi.nextLine();
12
13        if(!file.exists()) {
14            file.createNewFile();
15        }
16        FileWriter fileWriter = new FileWriter(file, true);
17        BufferedWriter bufferedWriter = new BufferedWriter(fileWriter);
18        bufferedWriter.write(metin);
19        bufferedWriter.write(" ");
20        bufferedWriter.close();
21    }
22    public void metinOku() throws IOException {
23        FileReader fileReader = new FileReader(file);
24        BufferedReader br = new BufferedReader(fileReader);
25        String metinOku;
26
27        while((metinOku = br.readLine()) != null){
28            System.out.println(metinOku);
29        }
30        br.close();
31    }
32 }
```

Bu kod main fonksiyondan ayrılan kullanmak istediğimiz özellikleri class biçiminde içinde tutan bir yapıdır. İstediğimiz fonksiyonlar birbirinden ayrı yazılmıştır. Kullanıcıdan metin almak için scanner kütüphanesinden yararlanılmış, yeni bir .txt dosyası açılmak için File'dan yardım alınmıştır. Yazmak için FileWriter ve BufferedWriter ve okumak için FileReader ve BufferedReader sınıflarından yararlanılmıştır. bufferedWriter içindeki write fonksiyonu yardımıyla kullanıcıdan alınan metin yazdırılır ve sonra kapatılır. Sonradan eklenicek metinler için de sonuna boşluk

koydum. `metinOku()` fonksiyonunun içindeki `while` döngüsü ise okunacak dosyanın içindeki yazıyı (eğer boş değilse) ekrana yazdırır.

```
public void kelimeSayma(){
    String dosyaYolu = "MetinBelgesi.txt";

    int kelimeSayisi = 0;

    try (BufferedReader br = new BufferedReader(new FileReader(dosyaYolu))) {
        String satir;
        while ((satir = br.readLine()) != null) {
            String[] kelimeler = satir.split(regex: "\\s+");
            kelimeSayisi += kelimeler.length;
        }
    } catch (IOException e) {
        e.printStackTrace();
    }

    System.out.println("Dosyadaki toplam kelime sayısı: " + kelimeSayisi);
}
```

İkinci ödev olan kelime sayma ödevinde ise `dosyaYolu` değişkenine attığım `.txt` dosyasını tekrardan `BufferedReader` ve `FileReader` kullanarak okudum. Try – catch bloğu içinde `while` kullanarak `.txt` belgesi (eğer boş değilse) baştan sona okundu ve kod her boşluk gördüğünde kelimelerden oluşan diziye eleman ekledi. En sonunda ise `.length` fonksiyonuyla eleman sayısı yazdırıldı. Aşağıda görünen çıktı bu fonksiyonun çıktısıdır.

```
Yusuf Asan 1306220078
1-Metin yaz
2-Metin oku
3-Metnin kelimelerini say
4-Çıkış
Yapmak istediğiniz işlemi seçiniz:
3
Dosyadaki toplam kelime sayısı: 3
```