Ödev 2 - Rapor

17253509
Bedrettin Bora Tanrıkulu
Bilgisayar Mühendisliği - 1.Sınıf

Vigenere Cipher

- Bu proje 3 dosyadan oluşmaktadır. Bunlar;
  1) Vigenere.java (main)
  2) Encryption.java
  3) Decryption.java

1) Vigenere.java

- Bu class projenin asıl dosyasıdır. Program bu dosya çalıştırılır. **Main** haricinda başka method'u yoktur. Kullanıcı iletişime bu main method'unda geçilir.

  İçerdiği menu ile Encryption ve Decryption işlemleri gerçekleştirilir ya da programdan çıkış sağlanır.

```java
import java.util.Scanner;

public class Vigenere {

    public static void main(String[] args) {
        String plaintext; // sifrelenecek metin
        String ciphertext; // sifrelenmis metin
        String key; // anahtar

        // Objects
        Scanner scan = new Scanner(System.in);
        Encryption encrypt = new Encryption();
        Decryption decrypt = new Decryption();

        while(true){
            // Menu
            System.out.println("################################");
            System.out.println("#       VIGENERE CIPHER        #");
            System.out.println("#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~#");
            System.out.println("# 1) Encryption (sifreleme)    #");
            System.out.println("# 2) Decryption (sifre çözme) #");
            System.out.println("# 9) Cikis                     #");
            System.out.println("################################");
            System.out.print("\t\t Secenek: "); int menu = scan.nextInt();

            // Encrypt and Decrypt
            switch(menu) {
                // Encryption (Şifreleme)
                case 1:
                    // Sifrelenecek metninin kullanicidan alinmasi
                    System.out.print("Sifrelenecek Metin: ");
                    scan.nextLine(); // Fazladan bir "enter" aldigi icin bu sekilde absorbe ettim

                    plaintext = scan.nextLine();
                    encrypt.setPlainText(plaintext.toLowerCase());
```

2) Encryption.java

- Bu class projenin şifreleme işlemini gerçekleştiridiği class'tır.
  Main tarafından **setPlainText** ile şifrelenmek üzerine bu class'a veri yollanır. Ardından bu veri **Encrypt** method'u ile şifrelenir.

  Encrypt method'u kullanıcı tarafından yollanan **Plain Text** ve **Key**'in harflerinin alfabede kaçıncı sıraya geldiğini belirler ardından bu iki değeri toplar. Bu toplamın modunu alır. Modu alınan bu değerin alfabede hangi harfe denk geldiğini bulur. Ardından **getCipherText** method'u ile şifreli veri main'e geri yollanır.

```java
public class Encryption{
    public final String alphabet = "abcçdefgğhıijklmnoöprsştuüvyz";
    public static String plaintext; // sifrelenecek metin
    public static StringBuilder ciphertext = new StringBuilder();

    public void encrypt(String key){
        int rankofplaintext;
        int rankofkey;

        ciphertext = new StringBuilder(""); // ciphertext'in icini temizler
        plaintext = plaintext.replace(" ", ""); // bosluk karakterini yok sayar
        key = key.replace(" ", "");

        for (int i=0; i<(plaintext.length()); i++) {
            rankofkey = alphabet.indexOf(key.charAt(i % key.length())); // key'in "i" sirasindaki harfinin a
            rankofplaintext = alphabet.indexOf(plaintext.charAt(i)); // plaintext'in "i" sirasindaki harfini

            int sum = rankofkey + rankofplaintext;
            sum = sum % alphabet.length();

            ciphertext.append(alphabet.charAt(sum));
        }
    }

    public void setPlainText(String plaintext) {
        this.plaintext = plaintext;
    }

    public String getCipherText() {
```

3) Decryption.java

- Bu class'da Encryption.java'da yapılan işlemin tam tersi yapılır.

  Main tarafından **setCipherText** ile şifresi çözülmek üzerine bu class'a veri yollanır. Ardından bu veri **Decrypt** method'u ile şifrelenir.

  Decrypt method'u kullanıcı tarafından yollanan **CipherText** ve **Key**'in harflerinin alfabede kaçıncı sıraya geldiğini belirler ardından bu iki değeri birbirinden çıkarır. Sonuç negatif çıkarsa alfabenin boyutu kadar üzerine ekleme yapılır. Bu sonucun modunu alır.

  Modu alınan bu değerin alfabede hangi harfe denk geldiğini bulur.
  Ardından **getPlainText** method'u ile şifreli veri main'e geri yollanır.

```java
public class Decryption{
    public final String alphabet = "abcçdefgğhıijklmnoöprsştuüvyz";
    public static String ciphertext; // sifrelenmis metin
    public static StringBuilder plaintext = new StringBuilder("");

    public void decrypt(String key){
        int rankofciphertext;
        int rankofkey;

        plaintext = new StringBuilder(""); // plaintext'in icini temizler
        ciphertext = ciphertext.replace(" ", ""); // bosluk karakterini yok sayar
        key = key.replace(" ", "");

        for (int i=0; i<(ciphertext.length()); i++) {
            rankofkey = alphabet.indexOf(key.charAt(i % key.length())); // key'in "i" sirasindaki harfinin alfa
            rankofciphertext = alphabet.indexOf(ciphertext.charAt(i)); // ciphertext'in "i" sirasindaki harfin

            int sum = rankofciphertext - rankofkey;
            if(sum < 0)
                sum += alphabet.length();
            sum = sum % alphabet.length();

            plaintext.append(alphabet.charAt(sum));
        }
    }

    public void setCipherText(String ciphertext) {
        this.ciphertext = ciphertext;
    }
```

Programın Çalıştırılması

- Programı çalıştırmak için aşağıdaki komutu kullanıyoruz;

```
cd /path/to/HW2Vigenere/
javac *.java && java Vigenere
```

- Aşağıdaki ekran görüntüsünde de gözüktüğü gibi, Şifrelenmek üzere **"şifrelenecekveri"** metnini ve **"anahtar"** parolasını programa girdiğimizde **"şyfazlünsclgvüry"** olan şifrelenmiş veriyi alırız.
- **"şyfazlünsclgvüry"** şifreli metinin şifresini çözmek için, **"anahtar"** parolası ile şifre çözme yaparsak, metnimiz olan "şifrelenecekveri" 'yi elde ederiz.

```
[fsutil@BoraT HW2Vigenere]$ javac *.java && java Vigenere
##############################
#       VIGENERE CIPHER      #
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~#
# 1) Encryption (sifreleme)  #
# 2) Decryption (sifre çözme) #
# 9) Cikis                   #
##############################
                Secenek: 1
Sifrelenecek Metin: şifrelenecekveri
Sifreleme Icin Anahtar: anahtar
Sifrelenmis Metin: şyfazlünsclgvüry


##############################
#       VIGENERE CIPHER      #
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~#
# 1) Encryption (sifreleme)  #
# 2) Decryption (sifre çözme) #
# 9) Cikis                   #
##############################
                Secenek: 2
Sifrelenmis Metin: şyfazlünsclgvüry
Sifre Acma Icin Anahtar: anahtar
Sifresi Cozulmus Metin: şifrelenecekveri
```

Dikdörtgen

- Bu proje 2 dosyadan oluşmaktadır. Bunlar;
  1) Dikdortgen.java (main)
  2) DikdortgenTes.java

1) Dikdortgen.java

- Bu class'ta bir dikdörtgenin koordinatları, yüksekliği ve genişliğini içerir. Bu değerler ile dikdörtgenin alanını, çevresini bulabilir.

  Bu class'ta get ve set işlemlerini kolaylaştırmak için **getter** ve **setter** isminde iki method vardır. Bu sayede aynı anda hem koordinat hem de yükseklik-genişlik belirlenip, geri alınabilir.
  shift method'u ile koordinat düzelminde dikdörtgen kaydırılabilir.

```java
public class Dikdortgen {
    private static int x; // noktanin x koordinati
    private static int y; // noktanin y koordinati
    private static int height; // yukseklik
    private static int width; // genislik

    public Dikdortgen(int x, int y, int height, int width) {
        this.x = x;
        this.y = y;
        this.height = height;
        this.width = width;
    }

    public void shift(int x, int y) {
        this.x = this.x + x;
        this.y = this.y + y;
    }

    public void setHeightWidth(int height, int width) {
        this.height = height;
        this.width = width;
    }

    public void setXY(int x, int y) {
        this.x = x;
        this.y = y;
    }

    public int getAlan() {
        return (this.width * this.height);
    }

    public int getCevre() {
        return (2*this.width + 2*this.height);
    }
```

2) DikdortgenTest.java

- Bu class sayesinde, Dikdortgen.java 'yı test edebiliriz. Test için 3 adet nesne oluşturulmuştur.

  İlk nesne oluşturulduğu gibi gösterilmiştir.
  İkinci nesne oluşturulmuş, ardından gözlenmiş ve üzerinden kaydırmak işlemi uygulanıp değişime bakılmıştır.

  Üçüncü nesnede ise, ikinci nesneye yapılan işlemlerin aynısı yapılmış ama ek olarak **setter** method'u ile değerler güncellenip tekrar sonuca bakılmıştır.

```java
import java.util.Scanner;

public class DikdortgenTest {

    public static void main(String[] args) {
        System.out.println();
        System.out.println("# Dikdortgen_1 #");
        Dikdortgen dikdortgen1 = new Dikdortgen(0, 0, 1, 11);
        dikdortgen1.getter(); // nesnenin degerleri gosterilir
        System.out.println();

        System.out.println("# Dikdortgen_2 #");
        Dikdortgen dikdortgen2 = new Dikdortgen(9, 2, 80, 38);
        dikdortgen2.getter(); // nesnenin degerleri gosterilir
        dikdortgen2.shifter(5,-3); // 5 birim yukari, 3 birim sola kaydirma islemi yapilir
        dikdortgen2.getter(); // yeni sonuc gozlemlenmek icin nesnenin degerleri tekrar gosterilir
        System.out.println();

        System.out.println("# Dikdortgen_3 #");
        Dikdortgen dikdortgen3 = new Dikdortgen(3, 1, 877, 21);
        dikdortgen3.getter(); // nesnenin degerleri gosterilir
        dikdortgen3.shifter(-41,41); // 41 birim asagi, 41 saga kaydirma islemi yapilir
        dikdortgen3.getter(); // yeni sonuc gozlemlenmek icin nesnenin degerleri tekrar gosterilir
        // Dikdortgen 3 icin setter parametresi ile degerlerin degistirilmesi
        System.out.println("Dikdortgen 3'un degerleri (5, 2, 23, 65) olarak guncellenirse;");
        dikdortgen3.setter(5, 2, 23, 65);
        dikdortgen3.getter();
        System.out.println();
    }
}
```

Programın Çalıştırılması

- Programı çalıştırmak için aşağıdaki komutu kullanıyoruz;

```
cd /path/to/HW2Dikdortgen/
javac *.java && java DikdortgenTest
```

  Programın çıktısı aşağıdaki gibidir.

```
[fsutil@BoraT HW2Dikdortgen]$ javac *.java && java DikdortgenTest

# Dikdortgen_1 #
[0,0] koordinatlarında bulunan; alani "11", cevresi "24" olan dikdortgen.

# Dikdortgen_2 #
[9,2] koordinatlarında bulunan; alani "3040", cevresi "236" olan dikdortgen.
Dikdortgen [5,-3] birim otelenirse;
[14,-1] koordinatlarında bulunan; alani "3040", cevresi "236" olan dikdortgen.

# Dikdortgen_3 #
[3,1] koordinatlarında bulunan; alani "18417", cevresi "1796" olan dikdortgen.
Dikdortgen [-41,41] birim otelenirse;
[-38,42] koordinatlarında bulunan; alani "18417", cevresi "1796" olan dikdortgen.
Dikdortgen 3'un degerleri (5, 2, 23, 65) olarak guncellenirse;
[5,2] koordinatlarında bulunan; alani "1495", cevresi "176" olan dikdortgen.
```