Echo-Client

Aufgabe 1
Echo-Client

Führe** nacheinander folgende Schritte **aus.

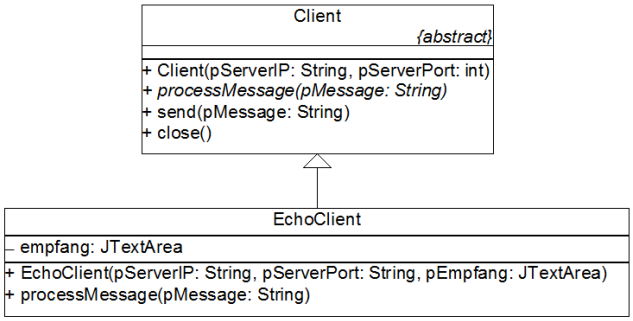
a) ***Erstelle** ein Unterverzeichnis und **kopiere** in dieses Verzeichnis die Abiturklassen `Connection` und `Client`.*

b) ***Erstelle** ein geeignetes Layout für den Echo-Client.*

TextArea!

```
name.append(„Text“);
```

c) ***Erstelle** eine Unterklasse `EchoClient` der Klasse `Client`.*



d) ***Implementiere** den Konstruktor und **überschreibe** die `processMessage(...)`-Methode.* Der Konstruktor der Klasse ruft nur den geerbten Konstruktor auf und speichert das mitgegebene Textfeld in der Variablen empfang.

```
public EchoClient(..., JTextArea pEmpfang) {
  super(pServerIP, pServerPort);
  empfang = pEmpfang;
}
```

Die Methode `processMessage(...)` protokolliert sämtliche Nachrichten in der TextArea `empfang`.

```
public void processMessage(String pMessage){
  empfang.append(pMessage+"\r\n");
}
```

Quellen: Logo (www.vielfalt-bewegt-frankfurt.de).

e) Nun fehlt nur noch die Anbindung der neuen Klasse `EchoClient` an die Oberfläche.
***Erstelle** ein Datenfeld/Attribut in der Klasse zur Oberfläche.*

```
   private EchoClient echoClient;
```

Die Ereignismethode des Buttons "mit Server verbinden" erzeugt nun das Objekt `EchoClient`.

```
public void jbVerbinden_ActionPerformed(ActionEvent evt) {
  echoClient = new EchoClient(jtfIP.getText(),
           Integer.parseInt(jtfPort.getText()),
           jtaEmpfang);
}
```

Die Ereignismethode des Buttons "Senden" gibt den Auftrag an das Objekt `echoClient` weiter.

```
public void jbSenden_ActionPerformed(ActionEvent evt) {
  echoClient.send(jtfSenden.getText());
}
```

Aufgabe 2
Testen

Nun soll dein Client getestet werden.

***Starte** zunächst deinen selbst programmierten Echo-Server auf dem Port 1000.*
***Starte** nun deinen selbst programmierten Echo-Client auf dem gleichen Port und **teste**, ob er funktioniert.*