

Name, Vorname:

Hilfsmittel:

Aufschriebe, Skripte, keine Rechner

Hinweise (unbedingt beachten):

- Alle Aufgaben müssen bearbeitet werden.
- Der Name und Vorname muß in DRUCKSCHRIFT auf jedes Aufgabenblatt und auf jedes Lösungsblatt geschrieben werden.
- Aufgabenblätter bitte auch abgeben.
- Die Lösungsblätter müssen in folgender Form durchnummeriert werden. Beispiel: 1/4 2/4 3/4 4/4
- Die rote Farbe darf nicht benutzt werden.
- Lassen Sie bitte auf der linken Seite einen mindestens 3cm breiten Rand.
- Selbsterklärende Variablennamen benutzen.
- Programme müssen benutzerfreundlich sein.
- EVA-Prinzip muss benutzt werden.
- Einrücken der entsprechenden Programmteile.
- Sichtbarkeits- und Zugriffsmodifizierer wie public, usw. dürfen nicht weggelassen werden.
- In allen Klassen dürfen in keiner Methode Ein- oder Ausgaben (auf dem Bildschirm) vorkommen, außer es handelt sich um reine Ausgabe- oder Eingabe Methoden.
- Bei Nichtbeachtung dieser Hinweise gibt es einen Punktabzug !!!!

AUFGABEN

1)

- a) Was ist der Dateizeiger ?
- b) Schildern Sie einen Fall, wo ihn ein Programmierer benötigt.
- c) Bei welcher Anweisung wird er nicht benötigt (weil er automatisch verschoben wird) ?

2)

- a) Was ist eine Klasse ?
- b) Was ist ein Objekt?
- c) Was enthält eine Klasse?
- d) Was ist der Sinn einer Methode?
- e) Was ist ein Attribut?
- f) Was bedeutet private?
- g) Was bedeutet public?

3) Schreiben Sie **ein** fehlerfreies Java-Programm, das folgendes **nacheinander** macht:

- a) Es legt die Dateien "c:\e3fiqueulle.txt" und "c:\e3fiziel.txt" an.
- b) Es sollen mehrere (z.B. 2) ganze Zahlen in die Datei "c:\e3fiqueulle.txt" geschrieben werden.
- c) Es sollen alle Zahlen von "c:\e3fiqueulle.txt" nach "c:\e3fiziel.txt" kopiert werden.
- d) Es sollen alle Zahlen von "c:\e3fiziel.txt" auf dem Bildschirm ausgegeben werden.

Wichtige Bemerkung:

Das Programm soll so geschrieben werden, dass c) und d) nicht nur für zwei Zahlen funktioniert, sondern für beliebig große Dateien "c:\e3fiqueulle.txt" und "c:\e3fiziel.txt".

Beispielprogramm:

```
import java.io.*;

public class MainTest1 {
    public static void main(String[] args) throws IOException {
        int raus;
        RandomAccessFile datei = null;

        datei=new RandomAccessFile("C:\\test.txt","rw");

        // Datei auslesen
        try{
            do{
                raus=datei.readInt();
            }while(true);
        }
        catch(FileNotFoundException e){
            System.out.println("Datei nicht gefunden"+e.toString());
        }
        catch(EOFException e){
            //Hier kommt man hin, wenn das Dateiende erreicht wurde
        }
        catch(IOException e){
            System.out.println("Lese-Fehler"+e.toString());
        }
        // Datei schliessen fehlt noch
    }
}
```

Lösungen:

1)

- a) Ein Dateizeiger ist ein Zeiger, der auf die aktuelle Position in der Datei zeigt.
- b) Letzte Byte einer Datei lesen.
- c) Beim Lesen und Schreiben.

2)

- a) Ein Bauplan
- b) Ein Objekt wird nach dem Bauplan hergestellt und im Arbeitsspeicher abgelegt.
- c) Daten und Methoden.
- d) auf ein Attribut zugreifen.
- e) Attribute sind die Daten der Klasse.
- f) private Bereich in der Klasse.
- g) öffentliche Bereich in der Klasse.

3)

```
import java.io.*;
public class MainDateiKopieren1 {
    public static void main(String[] args) throws IOException {
        RandomAccessFile dateiZiel = null;
        RandomAccessFile dateiQuelle = null;
        int z1, z2;
        int zahlQuelle, zahlZiel;
        int c;

        // Quell- und Zieldatei anlegen
        dateiQuelle=new RandomAccessFile("C:\\E3FIquelle.txt","rw");
        dateiZiel=new RandomAccessFile("C:\\E3FIziel.txt","rw");

        // Quelldatei beschreiben
        z1= 182;
        z2= 170;
        dateiQuelle.writeInt(z1);
        dateiQuelle.writeInt(z2);

        // Quelldatei in Zieldatei kopieren
        try{
            do{
                zahlQuelle=dateiQuelle.readInt();
                dateiZiel.writeInt(zahlQuelle);

                }while(true);
        }
        catch(FileNotFoundException e){
            System.out.println("Datei nicht gefunden"+e.toString());
        }
        catch EOFException e){
            //Hier kommt man hin, wenn das Dateiende erreicht wurde
        }
        catch(IOException e){
            System.out.println("Lese-Fehler"+e.toString());
        }
        }

        // Inhalt der Zieldatei auf Bildschirm ausgeben
        try{
            dateiQuelle.seek(0);
            dateiZiel.seek(0);

            System.out.println("Dateiinhalt=");
            do{
                zahlZiel=dateiZiel.readInt();
                System.out.println(zahlZiel);
            }while(true);
        }
        catch(FileNotFoundException e){
            System.out.println("Datei nicht gefunden"+e.toString());
        }
        catch EOFException e){
            //Hier kommt man hin, wenn das Dateiende erreicht wurde
        }
        catch(IOException e){
            System.out.println("Lese-Fehler"+e.toString());
        }
        }

        // Dateien schliessen
        try{
            dateiQuelle.close();
            dateiZiel.close();
        }
        catch(IOException e){
            System.out.println("Close-Fehler"+e.toString());
        }
        return;
    }
}
```

Name, Vorname:

Hilfsmittel:

Aufschriebe, Skripte, keine Rechner

Hinweise (unbedingt beachten):

- Alle Aufgaben müssen bearbeitet werden.
- Der Name und Vorname muß in DRUCKSCHRIFT auf jedes Aufgabenblatt und auf jedes Lösungsblatt geschrieben werden.
- Aufgabenblätter bitte auch abgeben.
- Die Lösungsblätter müssen in folgender Form durchnummeriert werden. Beispiel: 1/4 2/4 3/4 4/4
- Die rote Farbe darf nicht benutzt werden.
- Lassen Sie bitte auf der linken Seite einen mindestens 3cm breiten Rand.
- Selbsterklärende Variablennamen benutzen.
- Programme müssen benutzerfreundlich sein.
- EVA-Prinzip muss benutzt werden.
- Einrücken der entsprechenden Programmteile.
- Sichtbarkeits- und Zugriffsmodifizierer wie public, usw. dürfen nicht weggelassen werden.
- In allen Klassen dürfen in keiner Methode Ein- oder Ausgaben (auf dem Bildschirm) vorkommen, außer es handelt sich um reine Ausgabe- oder Eingabe Methoden.
- Bei Nichtbeachtung dieser Hinweise gibt es einen Punkteabzug !!!!

AUFGABEN

1) 20 P

Schreiben Sie ein Java-Programm (mit main-Methode) mit einer Klasse Ihrer Wahl (mit mindestens einem Konstruktor, mindestens einem Attribut, mindestens einer get-Methode und mindestens einer set-Methode.

(Das Java-Programm soll nichts mit Dateien zu tun haben).

2) 15 P

Gegeben ist die Datei "c:\e3fi.txt", die aus einer Anzahl von Zeichen (als Bytes realisiert) besteht, wie zum Beispiel den 4 Bytes "aBBa".

Überall, wo das Zeichen 'a' in der Datei vorkommt, soll es durch das Zeichen 'A' ersetzt werden.

Erstellen Sie dazu ein Struktogramm, **kein** Programm!

Wichtige Bemerkung:

Das Programm soll so geschrieben werden, daß es nicht nur für eine bestimmte Menge von Zeichen (z.B. 4 Zeichen funktioniert), sondern für eine beliebig große Datei

"c:\e3fi.txt".

3) 15 P

Schreiben Sie das zur Aufgabe 2) gehörige Java-Programm

Beachten Sie unbedingt die Beschreibung der Java-Doku zur Methode read (siehe Rückseite).

read

public int **read**()

throws IOException

Reads a byte of data from this file. The byte is returned as an integer in the range 0 to 255 (0x00-0x0ff). This method blocks if no input is yet available.

Although `RandomAccessFile` is not a subclass of `InputStream`, this method behaves in exactly the same way as the `InputStream.read()` method of `InputStream`.

Returns:

the next byte of data, or -1 if the end of the file has been reached.

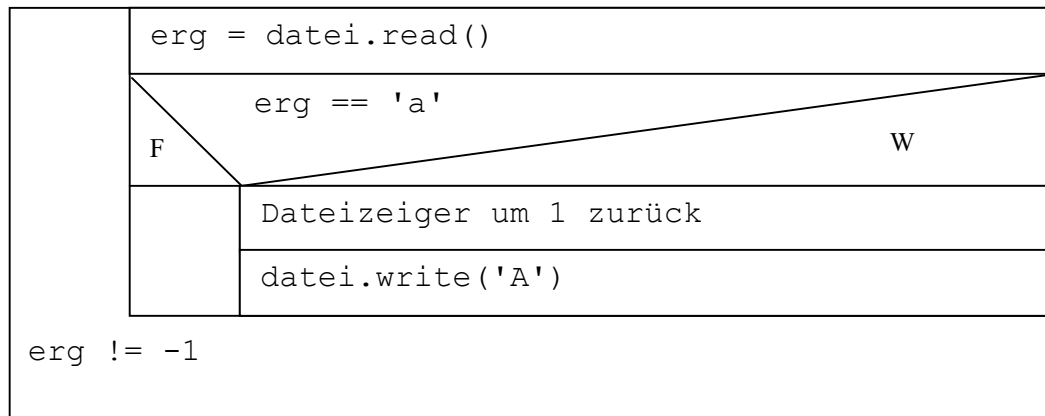
Throws:

IOException - if an I/O error occurs. Not thrown if end-of-file has been reached.

Lösung:

1)

2)



3) siehe Rückseite

```

import java.io.EOFException;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.RandomAccessFile;

public class MainTest2 {
    public static void main(String[] args) throws IOException {
        RandomAccessFile datei = null;
        int erg;
        long index;
        // Datei anlegen
        datei=new RandomAccessFile("C:\\E3FI.txt", "rw");
        // Quelldatei beschreiben
        datei.write('a');
        datei.write('b');

        datei.seek(0);

        try{
            do{
                erg=datei.read();
                if(erg=='a'){
                    // Dateizeiger um 1 zurück;
                    index=datei.getFilePointer();
                    datei.seek(index-1);
                    datei.write('A');
                }
            }while(erg!=-1);
        }
        catch(FileNotFoundException e){
            System.out.println("Datei nicht gefunden"+e.toString());
        }
        catch(EOFException e){
            //Hier kommt man hin, wenn das Dateiende erreicht wurde
        }
        catch(IOException e){
            System.out.println("Lese-Fehler"+e.toString());
        }
        }

        try{
            datei.seek(0);
            do{
                erg=datei.read();
                if(erg!=-1)
                    System.out.println(erg);
            }while(erg!=-1);
        }
        catch(FileNotFoundException e){
            System.out.println("Datei nicht gefunden"+e.toString());
        }
        catch(EOFException e){
            //Hier kommt man hin, wenn das Dateiende erreicht wurde
        }
        catch(IOException e){
            System.out.println("Lese-Fehler"+e.toString());
        }
        }

        // Dateien schliessen
        try{
            datei.close();
        }
        catch(IOException e){
            System.out.println("Close-Fehler"+e.toString());
        }
        return;
    }
}

```


Name, Vorname:

Hilfsmittel:

Bücher und Skripte eigener Wahl.

Hinweise (unbedingt beachten):

- Alle Aufgaben müssen bearbeitet werden.
- Der Name und Vorname muß in DRUCKSCHRIFT auf jedes Aufgabenblatt und auf jedes Lösungsblatt geschrieben werden.
- Aufgabenblätter bitte auch abgeben.
- Die Lösungsblätter müssen in folgender Form durchnummeriert werden. Beispiel: 1/4 2/4 3/4 4/4
- Die rote Farbe darf nicht benutzt werden.
- Lassen Sie bitte auf der linken Seite einen mindestens 3cm breiten Rand.
- Selbsterklärende Variablennamen benutzen.
- Programme müssen benutzerfreundlich sein.
- EVA-Prinzip muss benutzt werden.
- Einrücken der entsprechenden Programmteile.
- Sichtbarkeits- und Zugriffsmodifizierer wie public, usw. dürfen nicht weggelassen werden.
- In allen Klassen dürfen in keiner Methode Ein- oder Ausgaben (auf dem Bildschirm) vorkommen, außer es handelt sich um reine Ausgabe- oder Eingabe Methoden.
- Bei Nichtbeachtung dieser Hinweise gibt es einen Punktabzug !!!!

AUFGABEN

1) Entwickeln Sie ein Programm, das den Umfang und den Flächeninhalt von Kreisen berechnet. Das Programmdesign und die Lösung soll **objektorientiert** sein.

$$U = 2\pi r$$

$$A = \pi r^2$$

Bemerkung:

In der Klasse Kreis dürfen in keiner Methode Ein- oder Ausgaben (auf dem Bildschirm) vorkommen.

Konkret:

a) 15P

Erzeugen Sie die Klasse Kreis mit genau dem Attribut (und nur dem Attribut) "radius" und den zu "radius" gehörigen get-und set-Methoden und mit 2 Konstruktoren
Erzeugen Sie die Methode berechneFlaeche und berechneUmfang.

b) 6P

Erzeugen Sie in main zwei Kreise:
k1 mit Radius 2 und k2 mit Radius 10.

c) 9P

Geben Sie die Daten (Radius, Umfang, Fläche) von k2 auf dem Bildschirm aus

d) 4P

Aus Testgründen wird der über Tastatur eingegebene Radius des Objekts k1 verdoppelt und in dem Objekt k2 gespeichert. Dies soll in **einem** Ausdruck geschehen.

e) 9P

Geben Sie die Daten (Radius, Umfang, Fläche) von k2 nach der Verdopplung aus.

f) 7P

Berechnen Sie, um das Wievielfache sich dann der Flächeninhalt erhöht hat und geben Sie dieses Ergebnis auf dem Bildschirm aus.

Lösung:

```
public class MainTest4 {
    public static void main(String[] args){
        double verhaeltnis;
        // Teilaufgabe b)
        Kreis k1 = new Kreis(2);
        Kreis k2 = new Kreis(10);
        // Teilaufgabe c)
        System.out.println("k2.getRadius()= "+k2.getRadius());
        System.out.println("k2.berechneUmfang()= "+k2.berechneUmfang());
        System.out.println("k2.berechneFlaeche()= "+k2.berechneFlaeche());
        // Teilaufgabe d)
        k2.setRadius(k1.getRadius()*2);
        // Teilaufgabe e)
        System.out.println("k2.getRadius()= "+k2.getRadius());
        System.out.println("k2.berechneUmfang()= "+k2.berechneUmfang());
        System.out.println("k2.berechneFlaeche()= "+k2.berechneFlaeche());
        // Teilaufgabe f)
        verhaeltnis = k2.berechneFlaeche()/k1.berechneFlaeche();
        System.out.println("verhaeltnis= "+verhaeltnis);
    }
}

class Kreis{
    private double radius;
    public static final double pi = 3.14;

    public Kreis(double pRadius){
        radius = pRadius;
    }

    public Kreis(){
    }

    public void setRadius(double pRadius){
        radius = pRadius;
    }

    public double getRadius(){
        return(radius);
    }

    public double berechneUmfang(){
        return(2*pi*radius);
    }

    public double berechneFlaeche(){
        return(pi*radius*radius);
    }
}
```

Name, Vorname:

Hilfsmittel:

Aufschriebe, Skripte, keine Rechner

Hinweise (unbedingt beachten):

- Alle Aufgaben müssen bearbeitet werden.
- Der Name und Vorname muß in DRUCKSCHRIFT auf jedes Aufgabenblatt und auf jedes Lösungsblatt geschrieben werden.
- Aufgabenblätter bitte auch abgeben.
- Die Lösungsblätter müssen in folgender Form durchnummeriert werden. Beispiel: 1/4 2/4 3/4 4/4
- Die rote Farbe darf nicht benutzt werden.
- Lassen Sie bitte auf der linken Seite einen mindestens 3cm breiten Rand.
- Selbsterklärende Variablennamen benutzen.
- Programme müssen benutzerfreundlich sein.
- EVA-Prinzip muss benutzt werden.
- Einrücken der entsprechenden Programmteile.
- Sichtbarkeits- und Zugriffsmodifizierer wie public, usw. dürfen nicht weggelassen werden.
- In allen Klassen dürfen in keiner Methode Ein- oder Ausgaben (auf dem Bildschirm) vorkommen, außer es handelt sich um reine Ausgabe- oder Eingabe Methoden.
- Bei Nichtbeachtung dieser Hinweise gibt es einen Punkteabzug !!!!

AUFGABEN

1) 25 P

In einer Datei sollen alle vorkommenden Bindestriche durch Doppelpunkte ersetzt werden. Entwickeln Sie dazu ein Struktogramm!

Wichtige Bemerkung:

Das Struktogramm soll so entwickelt werden, dass es nicht nur für eine Datei einer speziellen, bestimmten Größe funktioniert, sondern für eine beliebig große Datei.

2) 25 P

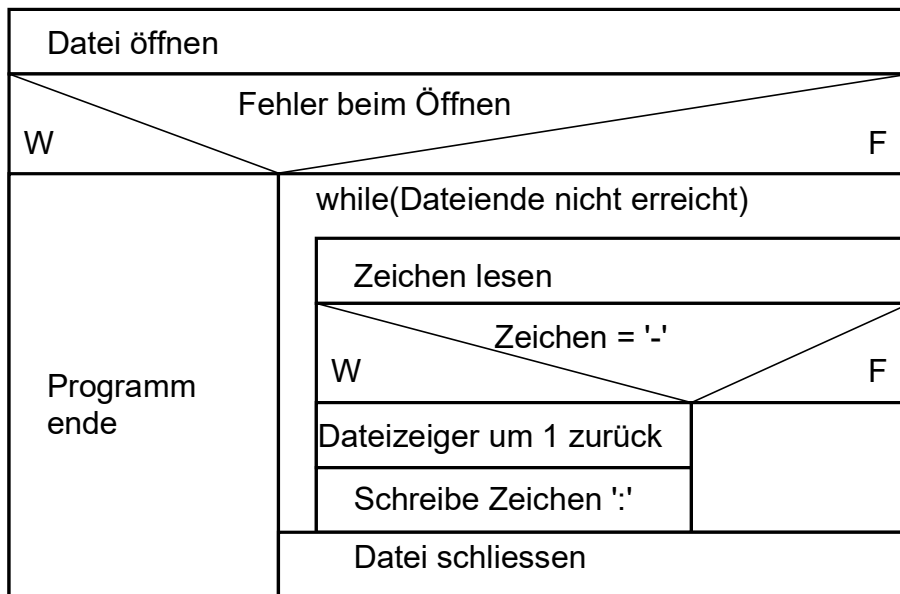
Ersetzen Sie die Umlaute Ä und ä der Datei "C:\test1.txt" durch "Ae" bzw. "ae" und speichern Sie den konvertierten Text in der Datei "C:\test2.txt" ab.

Wichtige Bemerkung:

Das Programm soll so geschrieben werden, dass es nicht nur für eine Datei einer speziellen, bestimmten Größe funktioniert, sondern für eine beliebig große Datei.

Lösungen

1)



2)

```
import java.io.*;
public class MainW_99_00_aufg2 {
    public static void main(String[] args){
        int wert;
        long dateiZeiger=0;
        RandomAccessFile myDateiQuelle = null;
        RandomAccessFile myDateiZiel = null;
        try{
            myDateiQuelle = new RandomAccessFile("C:\\test1.txt","r");
            myDateiZiel = new RandomAccessFile("C:\\test2.txt","rw");
            myDateiZiel.setLength(0);
            do{
                wert = myDateiQuelle.read();
                if(wert=='Ä'){
                    dateiZeiger = dateiZeiger-1;
                    myDateiZiel.write('A');
                    myDateiZiel.write('e');
                    dateiZeiger = dateiZeiger+2;
                }
                else if(wert=='ä'){
                    dateiZeiger = dateiZeiger-1;
                    myDateiZiel.write('a');
                    myDateiZiel.write('e');
                    dateiZeiger = dateiZeiger+2;
                }
                else{
                    dateiZeiger = dateiZeiger+1;
                    myDateiZiel.write(wert);
                }
            }
            while(myDateiQuelle.getFilePointer() <= myDateiQuelle.length()-1);
        }
        catch(FileNotFoundException e){
            System.out.println("Fehler: Datei nicht da: "+e.toString());
        }
        catch(IOException e){
            System.out.println("Fehler: Dateizugriff: "+e.toString());
        }
    }
}
```

KLAUSUR 3 SAE E3FI Nachtermin 1 Zeit: 45 Minuten

Name, Vorname:

Hilfsmittel:

Aufschriebe, Skripte, keine Rechner

Hinweise (unbedingt beachten):

- Alle Aufgaben müssen bearbeitet werden.
- Der Name und Vorname muß in DRUCKSCHRIFT auf jedes Aufgabenblatt und auf jedes Lösungsblatt geschrieben werden.
- Aufgabenblätter bitte auch abgeben.
- Die Lösungsblätter müssen in folgender Form durchnummeriert werden. Beispiel: 1/4 2/4 3/4 4/4
- Die rote Farbe darf nicht benutzt werden.
- Lassen Sie bitte auf der linken Seite einen mindestens 3cm breiten Rand.
- Selbsterklärende Variablennamen benutzen.
- Programme müssen benutzerfreundlich sein.
- EVA-Prinzip muss benutzt werden.
- Einrücken der entsprechenden Programmteile.
- Sichtbarkeits- und Zugriffsmodifizierer wie public, usw. dürfen nicht weggelassen werden.
- In allen Klassen dürfen in keiner Methode Ein- oder Ausgaben (auf dem Bildschirm) vorkommen, außer es handelt sich um reine Ausgabe- oder Eingabe Methoden.
- Bei Nichtbeachtung dieser Hinweise gibt es einen Punktabzug !!!!

AUFGABEN

1) 25 P

In einer Datei soll das 1. Byte entfernt werden.

Beispiel:

Datei vor der Entfernung des 1. Bytes

i	n	f	o	r	m	t	i	k
---	---	---	---	---	---	---	---	---

Datei nach der Entfernung des 1. Bytes

n	f	o	r	m	t	i	k
---	---	---	---	---	---	---	---

Entwickeln Sie dazu ein Struktogramm!

Bemerkung:

Mit

`void setLength(long newLength)`

kann man die Länge einer Datei neu setzen.

2) 25 P

Schreiben Sie **ein** fehlerfreies Java-Programm, das folgendes macht:

Um die Datei "`c:\geheim.txt`" zu verschlüsseln, soll der Dateiinhalte in umgekehrter Reihenfolge angeordnet werden.

Beispiel:

Datei vor der Verschlüsselung

i	n	f	o	r	m	a	t	i	k
---	---	---	---	---	---	---	---	---	---

Datei nach der Verschlüsselung

k	i	t	a	m	r	o	f	n	i
---	---	---	---	---	---	---	---	---	---

Wichtige Bemerkung:

Das Programm soll so geschrieben werden, dass es nicht nur für z.B. eine Datei der Größe 10 Bytes funktioniert, sondern für eine beliebig große Dateien "`c:\geheim.txt`".

Lösungen:

1)

laenge = Dateilänge bestimmen.
for(i=1; i<laenge; i++)
bewege Dateizeiger an die Stelle i
lese den Wert an der Stelle aus
bewege Dateizeiger an die Stelle i-1
schreibe den Wert an die Stelle

2)

```
import java.io.*;

public class MainAufgabe6 {
    public static void main(String[] args) {
        long laenge;
        int zeichen;
        RandomAccessFile myDatei = null;
        RandomAccessFile myTempDatei = null;

        try{
            myDatei = new RandomAccessFile("c:\\geheim.txt" , "rw");
            myTempDatei = new RandomAccessFile("c:\\temp.txt" , "rw");
            myDatei.seek(0);
            myTempDatei.seek(0);

            laenge=myDatei.length()-1;
            for(long i=laenge-1;i>=0;i--){
                myDatei.seek(i);
                zeichen=myDatei.read();
                myTempDatei.write(zeichen);
            }

            myDatei.seek(0);
            for(long i=0;i<laenge;i++){
                myTempDatei.seek(i);
                zeichen=myTempDatei.read();
                myDatei.write(zeichen);
                // zum Testen
                System.out.println((char)zeichen);
            }

            // Dateien schliessen
            myDatei.close();
            myTempDatei.close();
            // temporäre Datei löschen
            File f = new File("c:\\temp.txt");
            f.createNewFile();
            f.delete();
        }
        catch(FileNotFoundException e){
            System.out.println("Fehler: Datei nicht da: "+e.toString());
        }
        catch(IOException e){
            System.out.println("Fehler: Dateizugriff: "+e.toString());
        }
    }
}
```

KLAUSUR 3 SAE E3FI Nachtermin 2 Zeit: 45 Minuten

Name, Vorname:

Hilfsmittel:

Aufschriebe, Skripte, keine Rechner

Hinweise (unbedingt beachten):

- Alle Aufgaben müssen bearbeitet werden.
- Der Name und Vorname muß in DRUCKSCHRIFT auf jedes Aufgabenblatt und auf jedes Lösungsblatt geschrieben werden.
- Aufgabenblätter bitte auch abgeben.
- Die Lösungsblätter müssen in folgender Form durchnummeriert werden. Beispiel: 1/4 2/4 3/4 4/4
- Die rote Farbe darf nicht benutzt werden.
- Lassen Sie bitte auf der linken Seite einen mindestens 3cm breiten Rand.
- Selbsterklärende Variablennamen benutzen.
- Programme müssen benutzerfreundlich sein.
- EVA-Prinzip muss benutzt werden.
- Einrücken der entsprechenden Programmteile.
- Sichtbarkeits- und Zugriffsmodifizierer wie public, usw. dürfen nicht weggelassen werden.
- In allen Klassen dürfen in keiner Methode Ein- oder Ausgaben (auf dem Bildschirm) vorkommen, außer es handelt sich um reine Ausgabe- oder Eingabe Methoden.
- Bei Nichtbeachtung dieser Hinweise gibt es einen Punktabzug !!!!

AUFGABEN

1) 25 P

In einer Datei soll das 5. Byte entfernt werden.

Beispiel:

Datei vor der Entfernung des 5. Bytes

i	n	f	o	r	m	a	t	i	k
---	---	---	---	---	---	---	---	---	---

Datei nach der Entfernung des 5. Bytes

i	n	f	o	m	a	t	i	k
---	---	---	---	---	---	---	---	---

Entwickeln Sie dazu ein Struktogramm!

Bemerkung:

Mit

```
void setLength(long newLength)
```

kann man die Länge einer Datei neu setzen.

2) 25 P

Schreiben Sie **ein** fehlerfreies Java-Programm, das folgendes macht:

Fügen Sie den Inhalt der Datei "c:\test1.txt" an die Datei "c:\test2.txt" an.

Das Programm soll so geschrieben werden, dass es nicht nur für z.B. eine Datei der Größe 10 Bytes funktioniert, sondern für eine beliebig große Dateien.