

*Name, Vorname:*

Hilfsmittel:  
keine

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) 14P

- 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?

2) 10P

a) 2P

Wann und wie wird ein Konstruktor aufgerufen?

Bitte Beispiel geben (nur Aufruf angeben, keine Klasse implementieren).

b) 3P

Herr X behauptet : "Man braucht keine set-Methoden, da der Konstruktor genau das gleiche machen kann". Nehmen Sie dazu Stellung.

c) 3P

Angenommen, ein Programmierer hat keinen Konstruktor erzeugt.

Wann gibt es eine Fehlermeldung beim Kompilieren?

d) 2P

Was geschieht, wenn die Attribute einer Klasse nicht initialisiert werden und ein Objekt erzeugt wird? (Fehlermeldung?, Wert der Attribute ?, ...). Bitte genaue Beschreibung.

Auf der Rückseite geht es weiter !!!!!!!

3) (Alle Teilaufgaben müssen in **einem** Programm realisiert werden) 28P

a) Erstellen Sie die Klasse Konto, (mit genau 2 Attributen, genau 2 set-Methoden, genau 2 get-Methoden und genau 2 Konstruktoren), die ein Sparkonto mit einem Kontostand und einem Zinssatz repräsentieren soll.

b) Erzeugen Sie ein Konto mit dem Kontostand von 5 (Euro) und einem Zinssatz von 3%

c) Angenommen, man kennt nicht den aktuellen Kontostand und den Zinssatz .

Verändern Sie (nur mit Hilfe der get-bzw. set-Methoden) den Kontostand um den Wert +1000 (Euro) und den Zinssatz um -2 Prozentpunkte.

d) Ermitteln Sie mit Hilfe der entsprechenden Methoden den neuen Kontostand und den neuen Zinssatz und geben diese auf dem Bildschirm aus.

e) Erzeugen Sie ein anderes, neues Konto mit dem Kontostand von 10 (Euro) und einem Zinssatz von 5%.

Lösung:

1) 14 Punkte (jede Teilaufgabe 2 Punkte)

- a) Ein Bauplan, nach dem ein Objekt erstellt wird
- b) Ein nach einem Bauplan im Arbeitsspeicher angelegter, reservierter Speicher.
- c) Attribute und Methoden.
- d) Attribute zu lesen und zu beschreiben.
- e) Daten (Eigenschaften) in einer Klasse.
- f) Auf private Member darf nur innerhalb einer Klasse zugegriffen werden.
- g) Auf public Member darf innerhalb und außerhalb einer Klasse zugegriffen werden.

2)

10P

a) 2P

Ein Konstruktor wird aufgerufen, wenn ein Objekt mit new erzeugt wird.

Hund hund1 = new Hund("rex",45);

b) 3P

Diese Ansicht ist falsch:

Mit einem Konstruktor kann man nur einmal (beim Erzeugen des Objekts) die Attribute dieses Objekts festlegen. Will man diese später verändern, braucht man eine set-Methode.

c) 3P

Wenn der Programmierer einen Konstruktor mit mindestens einem Parameter erstellt und damit ein dazu entsprechendes Objekt erzeugt.

d) 2P

Es gibt keine Fehlermeldung, denn die Attribute des angelegten Objekts werden standardmäßig mit 0 vorbelegt.

3)

```
public class MainKonto1 {
    public static void main(String[] args) {
        Konto k1, k2;
        // 2 P
        k1= new Konto(5,3);
        // 6 P
        k1.setKontostand(k1.getKontostand()+1000);
        k1.setZinssatz(k1.getZinssatz()-2);
        // 4 P
        System.out.println("neuer Kontostand="
                           "+k1.getKontostand());
        System.out.println("neuer Zinssatz="
                           "+k1.getZinssatz());

        // 2 P
        k2= new Konto(10,5);
    }
}
```

```
class Konto{
    // 2 Punkte
    private double kontostand;
    private double zinssatz;

    // 2 Punkte
    public Konto(){
        kontostand = 0;
        zinssatz = 0;
    }

    // 2 Punkte
    public Konto(double pKontostand, double pZinssatz){
        kontostand = pKontostand;
        zinssatz = pZinssatz;
    }

    // 2 Punkte
    public double getKontostand() {
        return kontostand;
    }

    // 2 Punkte
    public double getZinssatz() {
        return zinssatz;
    }

    // 2 Punkte
    public void setKontostand(double kontostand) {
        this.kontostand = kontostand;
    }

    // 2 Punkte
    public void setZinssatz(double zinssatz) {
        this.zinssatz = zinssatz;
    }
}
```

*Name, Vorname:*Hilfsmittel:  
keine

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

Eine natürliche Zahl  $n > 0$  heißt vollkommene Zahl (perfekte Zahl), wenn sie gleich der Summe aller ihrer positiven Teiler (außer sich selbst) ist.

Beispiel:

Die Summe der Teiler vom 6 (außer sich selbst):  $1 + 2 + 3 = 6$ , also ist 6 vollkommen.

Schreiben Sie ein Programm, das bestimmt, ob eine natürliche Zahl  $n > 0$  vollkommen ist.

2) 25 P

Zu Beginn eines Jahres wird ein Anfangskapital  $K_0$  zum Zinssatz  $p$  angelegt. Welchen Wert hat es nach

1, 2, 3, ...  $n$  Jahren, wenn die Zinsen auf dem Sparbuch bleiben ?

Mathematische Anleitung anhand eines Beispiels:

Voraussetzungen:

Anfangskapital  $K_0 = 1000$

$p=10\%$

n	Kapital K	Zinsen	neues Kapital
0	1000	$1000 * 10/100 = 100$	$1000 + 100 = 1100$
1	1100	$1100 * 10/100 = 110$	$1100 + 110 = 1210$
2	1210	$1210 * 10/100 = 121$	$1210 + 121 = 1331$
3	...	...	...

Schreiben Sie ein Programm, das das Endkapital  $K_{\text{end}}$  berechnet, das ein Anfangskapital  $K_0$  bewirkt, wenn es zu einem Zinssatz von  $p\%$  insgesamt  $n$  Jahre auf dem Konto angelegt wird (und die Zinsen auf dem Sparbuch bleiben) ?

5P

*Name, Vorname:*

Hilfsmittel:  
keine

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

1a) 2P

Was ist falsch an folgendem Java-Programmausschnitt? (mit Begründung!)

```
...  
int[] v;  
v = new int[3];  
v[3] = -3;  
...
```

b) 2P

Was ist falsch an folgendem Java-Programmausschnitt? (mit Begründung!)

```
...  
double[] w, zahlen;  
zahlen[0] = 12;  
...
```

c) 2P

Ist der folgende Java-Programmausschnitt syntaktisch korrekt?

Wenn ja, welche Werte haben die Elemente der Variablen w?

```
...  
int[] w;  
w = new int[2];  
w[0] = -123;  
...
```

2) 4 P

Die Klasse Schaf soll schon existieren (mit genau einem zweiparametrischen Konstruktor aus integer Parametern, die das Gewicht und das Alter festlegen).

Erstellen Sie das Feld "schafstall" mit 2 Schafen.

Das erste Schaf wiegt 20 Kilo und ist 2 Jahre alt,

das zweite Schaf wiegt 30 Kilo und ist 3 Jahre alt.

3)

a)

18P

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

Erzeugen Sie die Methode berechneFlaeche und berechneUmfang.

b)

22P

Machen Sie in main() hintereinander Folgendes

b1) Erstellen Sie in dem Feld "kreise" 100 Kreise mit den Radien 0, 1, 2, ...99

b2) Geben Sie die Flächeninhalte dieser Kreise auf dem Bildschirm aus.

b3) Verdoppeln Sie die Radien dieser Kreise

b4) Speichern Sie diese Kreise (mit doppeltem Radius) zur Sicherheit in einem neuen Feld mit dem Namen "kreisSicherung".

b5) Angenommen, jemand fügt an das Ende in main()die 2 folgenden Programmierzeilen:

```
kreisSicherung[10].setRadius(13);  
System.out.println("Radius =" + kreise[10].getRadius());
```

Was wird auf dem Bildschirm ausgegeben?

Begründen Sie!

Bemerkung:

Ausgaben auf Bildschirm NIE in "normalen" Methoden machen, sondern nur in eigens dafür erstellten Ausgabe-Methoden realisieren!

In "normalen" Methoden müssen statt Bildschirmausgaben diese Werte über return zurückgeliefert werden oder in entsprechenden Attributen gespeichert werden.

## Lösungen:

1)

10P

a) 2P

`v[3] = -3;` überschreibt nicht reservierten Speicher

b) 2P

Für das Feld `zahlen` wurde kein Speicher reserviert.

`zahlen` ist eine lokale Variable, deren Wert undefiniert ist.

c) 2P

Die Werte des Felds `w` sind nach dem Erstellen mit 0 vorbelegt, `w[1]` wurde verändert, also:

`w[0] = -123`

`w[1] = 0`

2)

4P

`Schaf [] schafstall = {new Schaf (20, 2), new Schaf (30, 3)};`

3)

a)

18P

```
class Kreis{
    private double radius;                // 2P
    // pi=3,14.. gilt auch außerhalb der Klasse Kreis,
    // deshalb keine schöne Lösung !!
    public static final double pi = 3.14;

    public Kreis(double pRadius){         // 2P
        radius = pRadius;
    }

    public Kreis(){                       // 2P
    }

    public void setRadius(double pRadius){ // 3P
        radius = pRadius;
    }

    public double getRadius(){            // 3P
        return(radius);
    }

    public double berechneUmfang(){       // 3P
        return(2*pi*radius);
    }

    public double berechneFlaeche(){      // 3P
        return(pi*radius*radius);
    }
}
```



b)

22P

```
public class MainBKII1_13_11_10_Nr3 {
    public static void main(String[] args) {
        int i;
        // b1
        Kreis[] kreise;
        kreise = new Kreis[100];
        for(i=0;i<100;i++){
            kreise[i]=new Kreis(i);
        }

        // b2
        for(i=0;i<100;i++){
            System.out.println("Flaeche="+kreise[i].berechneFlaeche());
        }

        // b3
        for(i=0;i<100;i++){
            kreise[i].setRadius(kreise[i].getRadius()*2);
        }

        // b4
        Kreis[] kreisSicherung;
        kreisSicherung=kreise;

        //alternativ
        for(i=0;i<100;i++){
            kreisSicherung[i]=kreise[i];
        }

        // b5)
        // Radius=13
        // kreisSicherung[10] und kreise [10] zeigen auf das
        // gleiche Objekt.
    }
}
```

*Name, Vorname:*

Hilfsmittel:  
keine

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) 26 P

Ein Auto besteht aus genau einem Kofferraum. Dies soll modelliert werden.

Beachten Sie: Wenn das Auto verschrottet wird, wird auch der Kofferraum zerstört.

a)

Erzeugen Sie die Klasse Kofferraum mit genau dem Attribut (und nur diesem Attribut) "volumen" und den dazugehörigen get- und set-Methoden und genau einem Konstruktor (mit genau einem Parameter).

Außerdem muß noch die Methode printAll(...) erstellt werden, die sämtliche Daten der Klasse auf dem Bildschirm ausgibt.

b)

Erzeugen Sie die Klasse Auto mit genau 2 Attributen. Ein Attribut davon ist "kfz" und bedeutet das Kraftfahrzeugzeichen des Autos.

Außerdem müssen noch jeweils 2 get- und 2 set-Methoden und ein Konstruktor mit genau 2 Parametern erstellt werden.

Außerdem muß noch die Methode printAll(...) erstellt werden, die sämtliche Daten der Klasse auf dem Bildschirm ausgibt.

c)

Erstellen Sie in main() das Auto myAuto mit dem KFZ "E2 FI 10", das einen Kofferraum mit einem Volumen von 1000 cm<sup>3</sup> hat.

2)

27 P

Ein Professor hat genau einen Assistenten. Dies soll modelliert werden.

Beachten Sie: Wenn der Professor stirbt, stirbt nicht notwendig sein Assistent (sondern lebt normalerweise weiter).

a)

Erzeugen Sie die Klasse Assistent mit genau dem Attribut (und nur diesem Attribut) "name" und den dazugehörigen get- und set-Methoden und genau einem Konstruktor (mit genau einem Parameter).

b)

Erzeugen Sie die Klasse Professor mit genau 2 Attributen. Ein Attribut davon ist "fach" und bedeutet das Fachgebiet, auf dem der Professor arbeitet (z.B. Informatik)

Außerdem müssen noch jeweils 2 get- und 2 set-Methoden und ein Konstruktor mit genau 2 Parametern erstellt werden.

Außerdem muß noch die Methode printAll(...) erstellt werden, die sämtliche Daten der Klasse auf dem Bildschirm ausgibt.

c)

Erstellen Sie in main() den Professor myProf mit dem Fachgebiet "ED", der einen Assistenten mit Namen Maier hat.

## Lösungen:

```
class Kofferraum{                                     // 9P
    private int volumen;

    public Kofferraum(int volumen) {
        this.volumen = volumen;
    }

    public int getVolumen() {
        return volumen;
    }

    public void setVolumen(int volumen) {
        this.volumen = volumen;
    }

    public void printAll(){
        System.out.println("Volumen des Kofferraums="+volumen);
    }
}

class Auto{                                           // 14P
    private String kfz;
    private Kofferraum kofferraum;

    public Auto(String kfz, int volumen) {
        this.kfz = kfz;
        kofferraum = new Kofferraum(volumen);
    }

    public String getKfz() {
        return kfz;
    }

    public void setKfz(String kfz) {
        this.kfz = kfz;
    }

    public Kofferraum getKofferraum() {
        return kofferraum;
    }

    public void setKofferraum(Kofferraum kofferraum) {
        this.kofferraum = kofferraum;
    }

    public void printAll(){
        System.out.println("KFZ="+kfz);
        System.out.println("Volumen des
                               Kofferraum="+kofferraum.getVolumen());
    }
}

public class Startklasse {
    public static void main(String[] args) {
        Auto myAuto = new Auto("E2-FS 10", 2000);
        myAuto.printAll();
    }
}
```

// 2P

// 1P

```

class Assistent{
    String name;

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public Assistent(String name) {
        this.name = name;
    }

    public void printAll(){
        System.out.println("Name des Assistenten="+name);
    }

}

class Professor{
    String fach;
    Assistent assistant;

    public String getFach() {
        return fach;
    }

    public void setFach(String fach) {
        this.fach = fach;
    }

    public Assistent getAssistent() {
        return assistant;
    }

    public void setAssistent(Assistent assistant) {
        this.assistant = assistant;
    }

    public Professor(String fach, Assistent assistant) {
        this.fach = fach;
        this.assistant = assistant;
    }

    public void printAll(){
        System.out.println("Fachgebiet="+fach);
        System.out.println("Namen des Assistenten="+assistant.getName());
    }

}

public class Startklasse {
    public static void main(String[] args) {
        Assistent a = new Assistent("Maier");
        Professor prof = new Professor("EDV", a);

    }

}

```

// 9P

// 14P

// 2P

// 2P

*Name, Vorname:*

Hilfsmittel:  
keine

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

Bemerkungen:

Falls in der Lösung Felder verwendet werden, dürfen diese keine dynamischen Felder sein (wie z.B. ArrayList).

Die Programme müssen den Prinzipien der OOP entsprechen.

Alle Teilaufgaben müssen in **einem** Programm realisiert werden und die einzelnen Aufgabenteile als Kommentar eingefügt werden, wie z.B.

// 1)

bzw.

// 1a)

1)

50 P

Ein Dokument (wie z.B. ein Buch) besteht aus mehreren Kapiteln.

Dies soll modelliert werden.

Beachten Sie: Wenn das Dokument zerstört (z.B. durch Abbrennen) wird, werden auch die einzelnen Kapitel des Dokuments zerstört.

a) 9P

Erzeugen Sie die Klasse Kapitel mit genau dem Attribut (und nur diesem Attribut) "name" und den dazugehörigen get- und set-Methoden und genau einem Konstruktor (mit genau einem Parameter).

Außerdem muß noch die Methode printAll(...) erstellt werden, die sämtliche Daten der Klasse auf dem Bildschirm ausgibt.

b)

Erzeugen Sie die Klasse Dokument mit genau 2 Attributen. Ein Attribut davon ist "titel" und bedeutet den Titel des Dokuments.

Außerdem müssen noch jeweils 2 get- und 2 set-Methoden und ein Konstruktor mit genau 2 Parametern erstellt werden.

Die sich auf einen Raum beziehende get- und set-Methode muß an einer bestimmten Stelle des entsprechenden Feldes einen Raum einfügen bzw. diesen zurückliefern.

Dazu muß nachgeprüft werden, ob sich dieser Raum innerhalb der Feldgrenzen bewegt.

Ist dies nicht der Fall, muß bei der get-Methode null zurückgeliefert werden und bei der set-Methode darf kein Raum eingefügt werden.

c)

Erstellen Sie die Methode `... deleteAllKapitals()` ... die alle Kapitel des Dokuments löscht (auf "null" setzt).

d)

Erstellen Sie die Methode `... fillAllKapitals()` ... die alle Kapitel des Dokuments erzeugt und mit den Standardnamen "X0", "X1", "X2", usw. vorbelegt.

e)

Außerdem muß noch die Methode `printAll(...)` erstellt werden, die sämtliche Daten der Klasse auf dem Bildschirm ausgibt.

f)

Implementieren Sie die nötigen Anweisungen in `main()`, damit Folgendes realisiert wird:

Ein V3-Experte erstellt das Dokument mit Namen "V3", das maximal 1000 Kapitel besitzt.

Alle Kapitel des Dokuments müssen mit den Standardnamen "X0", "X1", "X2", ... vorbelegt werden. Nachdem der Autor des Dokuments Spontaneingebungen hatte, nennt er das

Das 1. Kapitel des Dokuments "Warum V3" und das

Das 2. Kapitel des Dokuments "Der V3 heilt".

Nachdem der V3-Experte selbst einmal im V3 gesessen ist, hat er nach neuen tranceähnlichen Zuständen alle Kapitel des V3-Dokumnets gelöscht.

## Lösungen:

```
class Kapitel{    // 9P
    private String name;

    public Kapitel(String pName){
        name=pName;
    }

    public void setName(String pName){
        name=pName;
    }

    public String getName(){
        return name;
    }

    public void printAllAttributs(){
        System.out.println("Name des Kapitels="+name);
    }
}

class Dokument{    // 29P
    private String name;                // 1P
    private Kapitel[] seineRäume;       // 1P

    public Dokument(String pName, int maxAnzahl){    // 4P
        name=pName;
        seineRäume = new Kapitel[maxAnzahl];
    }

    public void setName(String pName){    // 2P
        name=pName;
    }

    public String getName(){    // 2P
        return name;
    }

    public void setKapitel(Kapitel kapitel, int index){    // 3P
        seineRäume[index]=kapitel;
    }

    public Kapitel getKapitel(int index){    // 3P
        return seineRäume[index];
    }

    public void fillAllKapitels(){    // 4P
        int i;
        for(i=0;i<seineRäume.length;i++){
            seineRäume[i]=new Kapitel("X"+i);
        }
    }

    public void deleteAllKapitals(){    // 4P
        int i;
        for(i=0;i<seineRäume.length;i++){
            seineRäume[i]=null;
        }
    }

    public void printAllAttributs(){    // 5P
        int i;
        System.out.println("Name des Dokumentes="+name);
        for(i=0;i<seineRäume.length;i++){
            if(seineRäume[i]!=null){
                seineRäume[i].printAllAttributs();
            }
        }
    }
}
```



```
public class Startklasse {  
    public static void main(String[] args) {  
        Dokument haus=new Dokument("V3", 10);           // 2P  
        Kapitel kapitel1=new Kapitel("Kapitel_1");       // 2P  
        Kapitel kapitel2=new Kapitel("Kapitel_2");       // 2P  
        haus.fillAllKapitels();                           // 2P  
        haus.setKapitel(kapitel1,0);                     // 2P  
        haus.setKapitel(kapitel2,1);                     // 2P  
        haus.deleteAllKapitals();                         // 2P  
    }  
}
```