

Name:

Korrektor:

Punkte:

Aufgabe 1

a) Zeichnen Sie zu der nachstehenden Syntax in EBNF die passenden Syntaxdiagramme.

Drei0 ::= '0' | '3' | '6' | '9'

Drei1 ::= '1' | '4' | '7'

Drei2 ::= '2' | '5' | '8'

Drei ::= Drei0 Drei0 | Drei1 Drei2 | Drei2 Drei1

b) Gegeben sei folgende EBNF:

Worte ::= {'a'|'b'|'c'}'x'['y'|'z']

Welche der folgenden Zeichenfolgen lassen sich aus dem Nicht-Terminalsymbol „Worte“ ableiten ?

b1) **xy**

b2) **xyz**

b3) **abx**

b4) **aaxyz**

b5) **abcy**

b6) **abcz**

Name:

Korrektor:

Punkte:

Aufgabe 1 Lösung

Name:

Korrektor:

Punkte:

Aufgabe 2

Schreiben Sie ein Programm, das die ungeraden Quadratzahlen zwischen 1 und 1000 ausgibt. Benutzen Sie die vorgegebene Klasse Aufgabe2 und ergänzen Sie die Methode main entsprechend!

```
public class Aufgabe2 {  
  
    public static void main(String[] args) {  
        // Hier hin mit Ihrer Lösung !  

```

```
    } // main  
}
```

Name:

Korrektor:

Punkte:

Aufgabe 2 Lösung

```
class Aufgabe2 {  
  
    public static void main(String[] args) {  
        for (int i=1; i*i<=1000; i+=2) {  
            System.out.println(i*i);  
        }  
    }  
}
```

Name:

Korrektor:

Punkte:

Aufgabe 3

Schreiben Sie eine Klasse QuadratischeGleichung(double a, double b, double c)

die die quadratische Gleichung $a*x^2+b*x+c=0$ löst. Durch Aufruf

der Methode double x1() resp. double x2() werden die 2 Lösungen

abgerufen. Hat die Gleichung keine reellen Lösungen, soll eine

Ausnahme ausgelöst werden, wenn x1() oder x2() aufgerufen werden.

$$x_{1/2} = -\frac{b}{2a} \pm \sqrt{\frac{b^2}{4a^2} - \frac{c}{a}}$$

Wir geben den Programmrahmen und eine Testklasse für QuadratischeGleichung vor:

```
class QuadratischeGleichung {
    double a,b,c,disk;

    QuadratischeGleichung(double a, double b, double c) {
        this.a=a; this.b=b; this.c=c;
        disk = b*b/(4*(a*a))-c*c/(a*a);
    }

    double x1() throws Exception {
        // hier bitte ergänzen

    }

    double x2() throws Exception {
        // hier bitte ergänzen

    }
}

// Hier wird die Klasse benutzt

class Aufgabe3 {
    public static void main(String[] args) {
        QuadratischeGleichung q=new QuadratischeGleichung(1,2,10);
        try {
            System.out.println(q.x1());
        } catch (Exception e) {
            System.out.println(e); }
    }
}
```

Name:

Korrektor:

Punkte:

Aufgabe 3 Lösung

```
double x1() throws Exception {  
    if (disk<0) throw new Exception("Ergebnis imaginär");  
    return -b/(a+a)+Math.sqrt(disk);  
}
```

```
double x2() throws Exception {  
    if (disk<0) throw new Exception("Ergebnis imaginär");  
    return -b/(a+a)-Math.sqrt(disk);  
}
```

Name:

Korrektor:

Punkte:

Aufgabe 4

Gegeben sei die folgende Klasse LinList, die eine lineare Liste implementiert:

```

class ListItem {
    Object info;
    ListItem next;

    ListItem(Object o, ListItem next) {
        info = o; this.next = next;
    }
}

class LinList {
    int anzahl; ListItem head;

    LinList() {
        anzahl = 0; head = null;
    }

    void addFront(Object o) {
        ListItem h = new ListItem(o,head);
        head = h; anzahl++;
    }

    Object remove(int n) {
        Object h;
        if (n<0 || n>=anzahl) return null;
        if (n==0) {
            h = head.info; head = head.next;
        } else {
            ListItem l = head;
            for (int i=1; i<n; i++) l = l.next;
            h = l.next.info;
            l.next = l.next.next;
        }
        anzahl--;
        return h;
    }

    int nrElements() {
        return anzahl;
    }
}

```

Leiten Sie aus der Klasse LinList eine Klasse MyStack ab, die die üblichen Stackmethoden implementiert:

```

void push(Object o); // schiebt das Object o auf den Stapel
Object pop(); // liefert das oberste Objekt im Stack zurück und entfernt es
// falls der Stapel leer ist, wird null zurück gegeben
boolean empty(); // liefert true, falls der Stapel leer ist, sonst false.

```

Name:

Korrektor:

Punkte:

Ihre Lösung zu Aufgabe 4:

```
class MyStack
```


Name:

Korrektor:

Punkte:

Aufgabe 4 Lösung

```
class MyStack extends LinList {  
    void push(Object o) {  
        addFront(o);  
    }  
  
    Object pop() {  
        return remove(0);  
    }  
  
    boolean empty() {  
        return anzahl==0;  
    }  
}
```

Name:

Korrektor:

Punkte:

Aufgabe 5

Gegeben sei folgende Klasse

```

class Din {
    protected final double dinA0Breite = 0.841;

    protected double[] breite = new double[7];
    protected double[] hoehe = new double[7];

    double dinHoehe(int n) {
        return hoehe[n];
    }

    double dinBreite(int n) {
        return breite[n];
    }

    // Hier ist Platz für den Konstruktor

    Din() {

    }

} // Din()

```

DIN-An Querformat

Ergänzen Sie den Konstruktor der Klasse Din. Die Felder hoehe und breite sollen die Seitenmaße von DIN-A0 bis DIN-A6 im Hochformat aufnehmen. Die Breite und die Höhe des DIN-A0 Bogens können Sie der Konstante dinA0Breite entnehmen, die Höhe ist $\text{dinA0Breite} * \sqrt{2}$.

DIN-An Querformat

Die Höhe des kleineren DIN-A1 Bogens ist gleich der Breite des DIN-A0-Bogens, die DIN-A1-Breite ist gleich der halben DIN-A0-Höhe, usw.

← Din A(n-1) Hochformat

Name:

Korrektor:

Punkte:

Aufgabe 5 Lösung

```
Din() {  
    breite[0] = dinA0Breite;  
    hoehe[0] = dinA0Breite*Math.sqrt(2.0);  
  
    for (int i=1; i<7 ; i++) {  
        hoehe[i] = breite[i-1];  
        breite[i]= hoehe[i-1]/2.0;  
    }  
}
```

Name:

Korrektor:

Punkte:

Aufgabe 6

Leiten Sie aus der Klasse Frame eine Fensterklasse ab, die wie folgt auf einen Mausklick reagiert: an der Klickposition wird ein Kreuz gezeichnet, das aus einem waagrechten und einem senkrechten Balken aus je 5 Pixeln besteht. Wird die Zeichenfläche erneut angeklickt, wird das „alte“ Kreuz gelöscht und das neue gezeichnet. Ergänzen Sie die Methoden paint und mousePressed !

```
import java.awt.*;
import java.awt.event.*;

class Aufgabe6 extends Frame {
    public static void main(String[] args) {
        Aufgabe6 wnd = new Aufgabe6();
    }

    Aufgabe6() {
        super("Where the cross is made");
        addMouseListener(new MyMouseListener());
        setBackground(Color.lightGray);
        setSize(300,200); setVisible(true);
    }

    int crossX = -1; int crossY = -1;

    public void paint(Graphics gc) {

    }

    class MyMouseListener extends MouseAdapter {
        public void mousePressed(MouseEvent event) {

        }
    }
}
```

Name:

Korrektor:

Punkte:

Aufgabe 6 Lösung

```
import java.awt.*;
import java.awt.event.*;

class Aufgabe6 extends Frame {
    public static void main(String[] args) {
        Aufgabe6 wnd =new Aufgabe6();
    }

    Aufgabe6() {
        super("Where the cross is made");
        addWindowListener(new WindowClosingAdapter(true));
        addMouseListener(new MyMouseListener());

        setBackground(Color.lightGray);
        setSize(300,200);
        setVisible(true);
    }

    int crossX =-1; int crossY = -1;

    public void paint(Graphics gc) {
        if (crossX>=0) {
            gc.drawLine(crossX-2,crossY,crossX+2,crossY);
            gc.drawLine(crossX,crossY-2,crossX,crossY+2);
        }
    }

    class MyMouseListener
    extends MouseAdapter
    {
        public void mousePressed(MouseEvent event)
        {
            crossX = event.getX();
            crossY = event.getY();
            repaint();
        }
    }
}
```