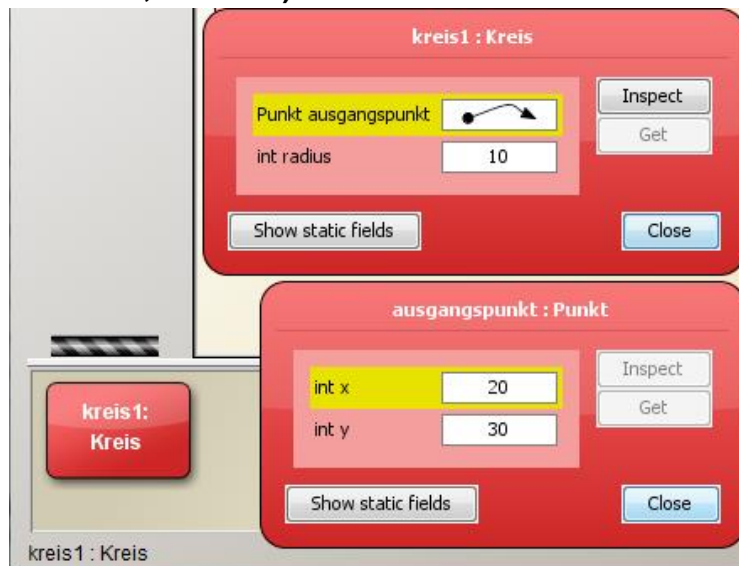


Ihre Ergebnisse werden ab 30.10.23 im Praktikum besprochen.

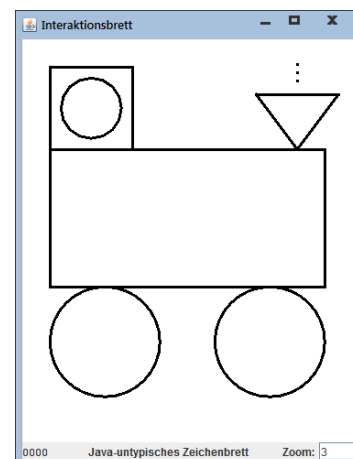
9. Aufgabe (Konstruktoren, get- und set-Methoden, 4 Punkte)

- a) Ergänzen Sie in jeder Klasse aus Aufgabe 8 mindestens einen Konstruktor, mit dem die Werte aller Objektvariablen gesetzt werden können.
- b) Ergänzen Sie in jeder Klasse aus Aufgabe 8 get- und set-Methoden für alle Objektvariablen.
- c) Machen Sie für jede Klasse jeweils ein Bildschirmfoto von den "Innereien" eines Objektes (also Inspektion aller Objektvariablen, wie rechts im folgenden Bild beispielhaft für einen Kreis gezeigt, dabei wird für Eigenschaften, die wieder mit Objekten beschrieben sind, wieder die „Inspect“-Funktion genutzt)



10. Aufgabe (6 Punkte, Klassennutzung, Aufruf von Methoden direkt für das Objekt, mit Code Pad und in einer eigenen Methode)

- a) Laden Sie sich die Klasse Interaktionsbrett.java von der Veranstaltungsseite in ein neues Projekt. Erzeugen Sie sich ein Objekt dieser Klasse direkt mit BlueJ. Nutzen Sie die folgenden Methoden, um ein beliebiges Bild mit mindestens sieben Elementen zu zeichnen und machen Sie einen Screenshot mit dem Objekt und dem Ergebnis Ihrer Zeichenkünste. Ein mögliches Ergebnis ist im Bild auf der rechten Seite dargestellt. Weitere Informationen zur Klasse Interaktionsbrett stehen unter <http://home.edvsz.hs-osnabrueck.de/skleuker/querschnittlich/Interaktionsbrett/Version1/index.html>.



Constructor Summary

[Interaktionsbrett](#) ()

Konstruktor zum Erzeugen eines Interaktionsbretts.

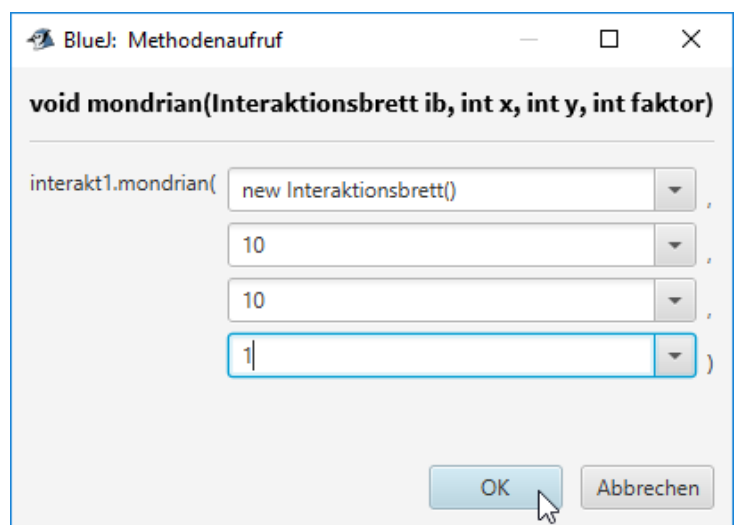
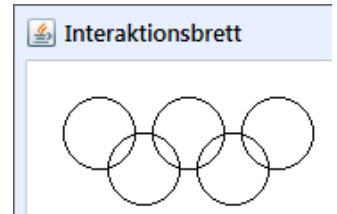
Method Summary

void	neueLinie (int x1, int y1, int x2, int y2) Methode zum Zeichnen einer neuen Linie.
void	neuerKreis (int x, int y, int radius) Methode zum Zeichnen eines neuen Kreises.
void	neuerPunkt (int x, int y) Methode zum Zeichnen eines neuen Punktes.
void	neuesRechteck (int x, int y, int breite, int hoehe) Methode zum Zeichnen eines neuen Rechtecks.

- b) Schreiben Sie eine neue Klasse InteraktionsbrettSpielerei mit einer Methode void schoenesBild(), die das gleiche Bild, wie in Aufgabe a) erstellt (Objekterzeugung, Methodenaufrufe). Das Interaktionsbrett soll dabei eine lokale Variable und keine Objektvariable sein. Wenn Sie ein Objekt Ihrer Klasse InteraktionsbrettSpielerei erzeugen und dann darauf schoenesBild() aufrufen, soll sich ein Interaktionsbrett mit dem Bild öffnen.

- c) Ergänzen Sie die Klasse InteraktionsbrettSpielerei um eine Methode, die die Ringe auf der rechten Seite ausgibt, beachten Sie, welche Kreise sich nur überschneiden.

- d) Ergänzen Sie folgende Methode in Ihrer Klasse InteraktionsbrettSpielerei.java und führen Sie diese Methode jeweils einmal mit von Ihnen fest gewählten Werten für x und y und folgenden Werten für faktor aus: 1, 2, 3, 0. Sichern Sie Ihre Ergebnisse mit Bildschirmfotos. Beachten Sie, dass man bei der Objektkonstruktion auch direkt ein neues Objekt als Parameter übergeben kann, wie es auf der rechten Seite mit der Eingabe new Interaktionsbrett() gezeigt wird. (Alternativ kann man das Objekt erst erzeugen, in die Objektleiste legen und dann den Namen des Objektes als Parameter nutzen.)

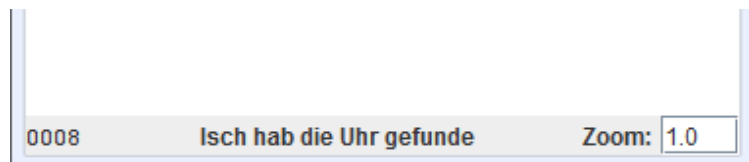


```
void mondrian(Interaktionsbrett ib, int x, int y, int faktor){  
    ib.neuesRechteck(x, y, 60 * faktor, 60 * faktor);  
    ib.neuesRechteck(x, y, 30 * faktor, 30 * faktor);  
    ib.neuesRechteck(x, y, 10 * faktor, 10 * faktor);  
}
```

- e) Überlegen Sie, welches Ergebnis folgende zusätzliche Methode in Ihrer Klasse InteraktionsbrettSpielerei liefern wird. Skizzieren Sie das Ergebnis *zunächst* auf Papier und prüfen Sie das Ergebnis dann durch eine Ausführung.

```
void watNu(){  
    Interaktionsbrett ibrett = new Interaktionsbrett();  
    this.mondrian(ibrett, 10, 10, 2);  
    this.mondrian(ibrett, 10, 10, 1);  
    this.mondrian(ibrett, 40, 40, 1);  
}
```

- f) Nutzen Sie die Methoden starteUhr(), stoppeUhr() und textZeigen() der Klasse



Interaktionsbrett, um ein Bildschirmfoto mit dem gezeigten Inhalt zu erzeugen, es handelt sich dabei um das untere Teil des Interaktionsbretts. Ob Sie dies mit einer weiteren Methode oder direkt im Code Pad umsetzen, können Sie auswählen.

Hinweis: Falls sich bei der Bearbeitung ein „BlueJ: terminal Window“ öffnen sollte, können Sie es ignorieren und schließen.

11. Aufgabe (4 Punkte, Umgang mit elementaren Grundbegriffen der objektorientierten Programmierung)

Gegeben sei die Klasse Konto auf der rechten Seite (auch von der Webseite erhältlich, aber eigentlich nicht benötigt). Bearbeiten Sie folgende Aufgaben schriftlich.

- Geben die Namen aller Objektvariablen an.
- Geben Sie die Parameterliste des Konstruktors aus Zeile 6 an.
- Wenn in Zeile 2 der Name „nummer“ in „kontonummer“ umgeändert werden würde. In welchen anderen Zeilen müssen mindestens Änderungen vorgenommen werden?
- Wenn in Zeile 6 der Name „nummer“ in „kontonummer“ umgeändert werden würde. In welchen anderen Zeilen müssen mindestens Änderungen vorgenommen werden?
- Auf welche Variablen (Objektvariablen, lokale Variablen) kann in Zeile 13 zugegriffen werden?
- Auf welche Variablen kann in Zeile 45 zugegriffen werden?
- Welchen Wert haben die Objektvariablen eines Objekts, das mit `new Konto(42, "ich")` erzeugt wurde?
- Welchen Wert haben die Objektvariablen eines Objekts, das mit `new Konto(42)` erzeugt wurde?
- Was erwarten Sie, was bei einer Objekterzeugung mit `new Konto("ich", 42)` passiert?
- Gegeben sei das folgende Programm, geben Sie mit Hilfe einer Tabelle nach jeder Zeile der Methode `spielerei()` an, welche Werte alle Variablen (lokale Variablen und evtl. deren Objektvariablen) haben.

```
class KontoAnalyse{
    void spielerei(){
        Konto k1 = new Konto(42, "ich");
        Konto k2 = new Konto(43);
        k1.einzahlen(200.00);
        k2.setBetrag(100.00);
        double diff = k1.getBetrag()
                    - k2.getBetrag();
        k2.abbuchen(diff);
        k1.einzahlen(k1.getBetrag());
        k2.abbuchen(k1.getBetrag());
    }
}
```

```
1 class Konto {
2     int nummer;
3     String inhaber = "anonym";
4     double betrag;
5
6     Konto(int nummer, String inhaber) {
7         this.nummer = nummer;
8         this.inhaber = inhaber;
9         this.betrag = 0.0;
10    }
11
12    Konto(int nummer) {
13        this.nummer = nummer;
14    }
15
16    int getNummer() {
17        return this.nummer;
18    }
19
20    void setNummer(int nummer) {
21        this.nummer = nummer;
22    }
23
24    String getInhaber() {
25        return this.inhaber;
26    }
27
28    void setInhaber(String inhaber) {
29        this.inhaber = inhaber;
30    }
31
32    double getBetrag() {
33        return this.betrag;
34    }
35
36    void setBetrag(double betrag) {
37        this.betrag = betrag;
38    }
39
40    void einzahlen(double wert) {
41        this.betrag = this.betrag + wert;
42    }
43
44    void abbuchen(double betrag) {
45        this.betrag = this.betrag - betrag;
46    }
47 }
```

12. Aufgabe (4 Punkte, Klassenmodellierung mit Objekterzeugung)

Gegeben sei das folgende Formular. Mit Formularen werden oft Daten von mehreren gemeinsamen Objekten erhoben. Setzen Sie die Struktur des Formulars als Klasse Formular mit verschiedenen Klassen um, die dann in Objektvariablen der Klasse Formular genutzt werden. Vorschläge für Klassen sind rot umrandet. Ergänzen Sie jeweils einen Konstruktor, mit dem alle Objektvariablen Werte erhalten können.

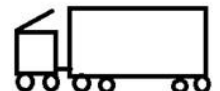
Schreiben Sie eine Klasse FormularTest mit einer Methode

Formular beispieleformularErstellen()

die als Ergebnis möglichst genau die Daten des obigen Formulars enthält. Machen Sie ein Bildschirmfoto des Formular-Objekts und der enthaltenen Objekte in BlueJ mit möglichst vielen Inhalten des obigen Formulars. (Es soll das Objekt mit seinen Inhalten, also mehreren weiteren Objekten, sichtbar sein. Sie sollen das Formular nicht mit dem Interaktionsbrett nachzeichnen, was natürlich mit einer Methode darstellen(Interaktionsbrett ib) durchaus auch interessant wäre.)

Hinweis: Es wurde in der Vorlesung bereits gezeigt, dass es eine Methode add(Objekt) der Klasse ArrayList gibt, mit der ein Objekt in eine ArrayList eingefügt werden kann.

Kostenvoranschlag Umzug



Kund*inn*enname:	Norbert Nerd		
Straße:	Müllerstr. 99		
Ort:	26384 Wilhelmshaven		
von:		nach:	
Stockwerk:	2	Stockwerk:	1
Straße:	Müllerstr. 99	Straße:	Barbarastr.11
Ort:	26384 Wilhelmshaven	Ort:	49071 Osnabrück
zu transportierende Güter:			
	#	Typ	
	1	Bett	
	2	Stuhl	
	2	Regal	
	1	Kleiderschrank	
	1	Tisch	
	3	Lampen	
	4	PC	
Dienstleistungen bei Auszug:		Dienstleistungen bei Einzug:	
#	Art	#	Art
2	Lampen abhängen	3	Lampen aufhängen
1	Schrank demontieren	1	Schrank montieren
Angebot gültig bis:		31.09.2011	
Kosten:		2112 €	