



Erste eigene Klassen

inkl. Anwendung (über Zugriffs-Klasse)

Auto - Klassendeklaration

```
public class auto {
    //Attribute
    private String Brand;
    private String Model;
    private String Color;

    //Konstruktor
    public auto(String brand, String model, String color) {
        Brand=brand;
        Model=model;
        Color=color;
    }

    //getter/ setter
    public void setBrand(String brand) { Brand = brand; }

    public String getBrand() { return Brand; }

    public void setModel(String model) { Model = model; }

    public String getModel() { return Model; }

    public void setColor(String color) { Color = color; }

    public String getColor() { return Color; }

    //weitere Methoden
    public void fahren() {
        System.out.println("Der " + Color + "e " + Brand + " fährt los");
    }

    public void radioAnmachen() {
        System.out.println("Das Radio ist an... \n Wühlmaus, Knopfaugen,
                           lange Nase wer könnte das sein ...");
    }

    public void tanken() {
        System.out.println("Das Auto tankt...");
    }
}
```



Auto - Zugriffs-Klasse

```
import java.util.Scanner; import java.util.Random;
```

```
public class main {  
    public static void main(String[] args) {  
        Scanner AutoErstellen = new Scanner(System.in);  
        System.out.println("Welches Auto möchtest du fahren (Marke, Model, Farbe);");  
        String brand = AutoErstellen.next();  
        String model = AutoErstellen.next();  
        String color = AutoErstellen.next();  
        auto userCar = new auto(brand, model, color);  
        System.out.println("Dein Auto ist ein " + userCar.getColor() +  
            "er " + userCar.getBrand() + " " + userCar.getModel());  
        Scanner Ziel = new Scanner(System.in);  
        System.out.println("Wo möchtest du hinfahren?");  
        String ReiseZiel = Ziel.next();  
        Scanner tanken = new Scanner(System.in);  
        System.out.println("Möchtest du davor tanken?");  
        String Tanken = tanken.next();  
        if (Tanken.equals("ja")) userCar.tanken();  
        userCar.fahren();  
        Scanner radio = new Scanner(System.in);  
        System.out.println("Möchtest du das Radio anschalten");  
        String Radio = radio.nextLine();  
        if (Radio.equals("ja")) userCar.radioAnmachen();  
        Random number1 = new Random();  
        Random number2 = new Random();  
        if (number1.nextInt(3) == 0 && Tanken.equals("nein")) {  
            System.out.println("Dein Tank ist leer. Du kannst nicht weiterfahren.");  
        }  
        else if (number2.nextInt(5) == 0) {  
            System.out.println("Du hattest einen Unfall. Du bist gestorben.");  
        }  
        else{  
            System.out.println("Du bist angekommen.");  
        }  
    }  
}
```

```
Welches Auto möchtest du fahren (Marke, Model, Farbe)  
VW Touran schwarz  
Dein Auto ist ein schwarzer VW Touran  
Wo möchtest du hinfahren?  
Portugal  
Möchtest du davor tanken?  
ja  
Das Auto tankt...  
Der schwarze VW fährt los  
Möchtest du das Radio anschalten  
nein  
Du bist angekommen.
```

Dreieck - Klassendeklaration (innermathematische Klasse - teils gekürzt - vollständig auf GitHub!)

```
public class ClassDreieck {
```

```
    private int posAX; private int posAY; private int posBX; private int posBY;  
    private int posCX; private int posCY;
```

Attribute

```
    public ClassDreieck(int posAX, int posAY, int posBX, int posBY, int posCX, int posCY) {  
        this.posAX = posAX; this.posAY = posAY; this.posBX = posBX;  
        this.posBY = posBY; this.posCX = posCX; this.posCY = posCY;  
    }
```

Konstruktor

```
    public int getPosAX() { return posAX; }
```

getter/ setter

```
    ...
```

```
    public void setPosAXAY(int posAX, int posAY) { this.posAX = posAX; this.posAY = posAY; }
```

```
    public double SeitenlängeA() {
```

weitere Methoden

```
        return Math.sqrt(Math.pow((posCX - posBX), 2) + Math.pow((posCY - posBY), 2));  
    }
```

```
    ...
```

```
    public double Umfang() { return SeitenlängeA() + SeitenlängeB() + SeitenlängeC(); }
```

```
    public double Flächeninhalt() {
```

```
        double s = Umfang() / 2;
```

```
        return Math.sqrt(s * (s - SeitenlängeC())  
            * (s - SeitenlängeA()) * (s -SeitenlängeB()));
```

```
    }
```

```
} //Ende ClassDreieck
```

Dreieck - Zugriffs-Klasse

```
import java.util.Scanner;
```

```
public class MainDreieck {
```

```
    public static void main(String[] args) {
```

```
        Scanner scan = new Scanner(System.in);
```

```
        System.out.println("Koordinaten Punkt A: ");
```

```
        System.out.println("X: ");
```

```
        int A = scan.nextInt();
```

```
        ...
```

```
        int F = scan.nextInt();
```

```
        ClassDreieck Michael = new ClassDreieck(A, B, C, D, E, F);
```

```
        System.out.println("Punkt A: (" + A + "/" + B + ")");
```

```
        ...
```

```
        System.out.println("SeitenlängeC: " + Michael.SeitenlängeC());
```

```
        ...
```

```
        System.out.println("Umfang: " + Michael.Umfang());
```

```
        System.out.println("Flächeninhalt: " + Michael.Flächeninhalt());
```

```
    }
```

```
}
```

```
Koordinaten Punkt A:  
X:  
1  
Y:  
0  
Koordinaten Punkt B:  
X:  
9  
Y:  
4  
Koordinaten Punkt C:  
X:  
6  
Y:  
7  
Punkt A: (1/0)  
Punkt B: (9/4)  
Punkt C: (6/7)  
SeitenlängeC: 8.94427190999916  
SeitenlängeA: 4.242640687119285  
SeitenlängeC: 8.602325267042627  
Umfang: 21.78923786416107  
Flächeninhalt: 17.99999999999996
```

Bruch - Klassendeklaration (Zugriffsklasse nur auf GitHub!)

```
public class Bruch {
```

```
    // Anfang Attribute
```

```
    private int Zaehler;
```

```
    private int Nenner;
```

```
    // Ende Attribute
```

```
    // Konstruktor
```

```
    public Bruch(int z, int n) {
```

```
        Zaehler = z;
```

```
        Nenner = n;
```

```
    }
```

```
    // Anfang Methoden
```

```
    public void setZaehler(int z) {
```

```
        Zaehler = z;
```

```
    }
```

```
    public int getZaehler() {
```

```
        return Zaehler;
```

```
    }
```

```
    public void setNenner(int n) {
```

```
        Nenner = n;
```

```
    }
```

```
    public int getNenner() {
```

```
        return Nenner;
```

```
    }
```

```
    public void erweitere(int x) {
```

```
        Zaehler *= x;
```

```
        Nenner *= x;
```

```
    }
```

```
    public void kuerze(int x) {
```

```
        if ((Zaehler % x == 0)
```

```
            && (Nenner % x == 0)) {
```

```
            Zaehler /= x;
```

```
            Nenner /= x;
```

```
        }
```

```
    }
```

```
    public void addiere(Bruch b) {
```

```
        Zaehler = Zaehler * b.getNenner()
```

```
            + Nenner * b.getZaehler();
```

```
        Nenner = Nenner * b.getNenner();
```

```
    }
```

```
    public void subtrahiere(Bruch b) {
```

```
        Zaehler = Zaehler * b.getNenner()
```

```
            + Nenner * b.getZaehler();
```

```
        Nenner = Nenner * b.getNenner();
```

```
    }
```

```
    public void multipliziere(Bruch b) {
```

```
        Zaehler = Zaehler * b.getZaehler();
```

```
        Nenner = Nenner * b.getNenner();
```

```
    }
```

```
    public void dividiere(Bruch b) {
```

```
        Zaehler = Zaehler * b.getNenner();
```

```
        Nenner = Nenner * b.getZaehler();
```

```
    }
```

```
    public int ggt() {
```

```
        int z = Zaehler;
```

```
        int n = Nenner;
```

```
        while (n != 0) {
```

```
            if (z > n) { z = z - n;
```

```
            } else { n = n - z;
```

```
            }
```

```
        }
```

```
        return z;
```

```
    }
```

```
    public int kgv() {
```

```
        int z = Zaehler;
```

```
        int n = Nenner;
```

```
        while (n != z) {
```

```
            if (z < n) {
```

```
                z = z + Zaehler;
```

```
            } else {
```

```
                n = n + Nenner;
```

```
            }
```

```
        }
```

```
        return z;
```

```
    }
```

```
    public void kuerzeVoll () {kuerze(ggt());}
```

```
    public double inDezimal() {
```

```
        return (double) Zaehler / (double) Nenner;
```

```
    }
```

```
}
```

Bruch
<div><div></div><div>Zaehler: int</div></div>
<div><div></div><div>Nenner: int</div></div>
<div><div></div><div>Bruch(int, int)</div></div>
<div><div></div><div>setZaehler(int)</div></div>
<div><div></div><div>getZaehler(): int</div></div>
<div><div></div><div>setNenner(int)</div></div>
<div><div></div><div>getNenner(): int</div></div>
<div><div></div><div>erweitere(int)</div></div>
<div><div></div><div>kuerze(int)</div></div>
<div><div></div><div>addiere(Bruch)</div></div>
<div><div></div><div>subtrahiere(Bruch)</div></div>
<div><div></div><div>multipliziere(Bruch)</div></div>
<div><div></div><div>dividiere(Bruch)</div></div>
<div><div></div><div>ggt(): int</div></div>
<div><div></div><div>kgv(): int</div></div>
<div><div></div><div>kuerzeVoll()</div></div>
<div><div></div><div>inDezimal(): double</div></div>