

Lokalisieren Sie bitte in der folgenden Abbildung mit einer Kurzform der Klasse Bruch

```

public class Bruch {
    private int zaehler;
    private int nenner = 1;
    private String etikett = "";
    static private int anzahl;

    public Bruch(int z, int n, String eti) {
        setzeZaehler(z);
        setzeNenner(n);
        setzeEtikett(eti);
        anzahl++;
    }
    public Bruch() {anzahl++;}

    public void setzeZaehler(int z) {zaehler = z;}
    public boolean setzeNenner(int n) {
        if (n != 0) {
            nenner = n;
            return true;
        } else
            return false;
    }
    public void setzeEtikett(String eti) {
        int rind = eti.length();
        if (rind > 40)
            rind = 40;
        etikett = eti.substring(0, rind);
    }
    public int gibZaehler() {return zaehler;}
    public int gibNenner() {return nenner;}
    public String gibEtikett() {return etikett;}

    public void kuerze() {
        . . .
    }
    public void addiere(Bruch b) {
        zaehler = zaehler*b.nenner + b.zaehler*nenner;
        nenner = nenner*b.nenner;
        kuerze();
    }
    public boolean frage() {
        . . .
    }

    public void zeige() {
        . . .
    }
    public void dupliziere(Bruch bc) {
        bc.zaehler = zaehler;
        bc.nenner = nenner;
        bc.etikett = etikett;
    }
    public Bruch kclone() {
        return new Bruch(zaehler, nenner, etikett);
    }

    static public int hanz() {return anzahl;}
}

class Bruchrechnung {
    public static void main(String[] args) {
        Bruch b1 = new Bruch(890, 25, "");
        b1.zeige(); b1.kuerze(); b1.zeige();
    }
}

```

Diagram showing 9 numbered callouts pointing to specific code elements:

- 1: `private int zaehler;`
- 2: `static private int anzahl;`
- 3: `public Bruch(int z, int n, String eti) {`
- 4: `setzeEtikett(eti) {`
- 5: `addiere(Bruch b) {`
- 6: `bc.zaehler = zaehler;`
- 7: `return new Bruch(zaehler, nenner, etikett);`
- 8: `static public int hanz() {return anzahl;}`
- 9: `new Bruch(890, 25, "");`

neun Begriffe der objektorientierten Programmierung, und tragen Sie die Positionen in die folgende Tabelle ein:

Begriff	Pos.
Definition einer Instanzmethode mit Referenzrückgabe	
Deklaration einer lokalen Variablen	
Definition einer Instanzmethode mit Referenzparameter	
Deklaration einer Instanzvariablen	
Methodenaufruf	

Begriff	Pos.
Konstruktordefinition	
Deklaration einer Klassenvariablen	
Objekterzeugung	
Definition einer Klassenmethode	