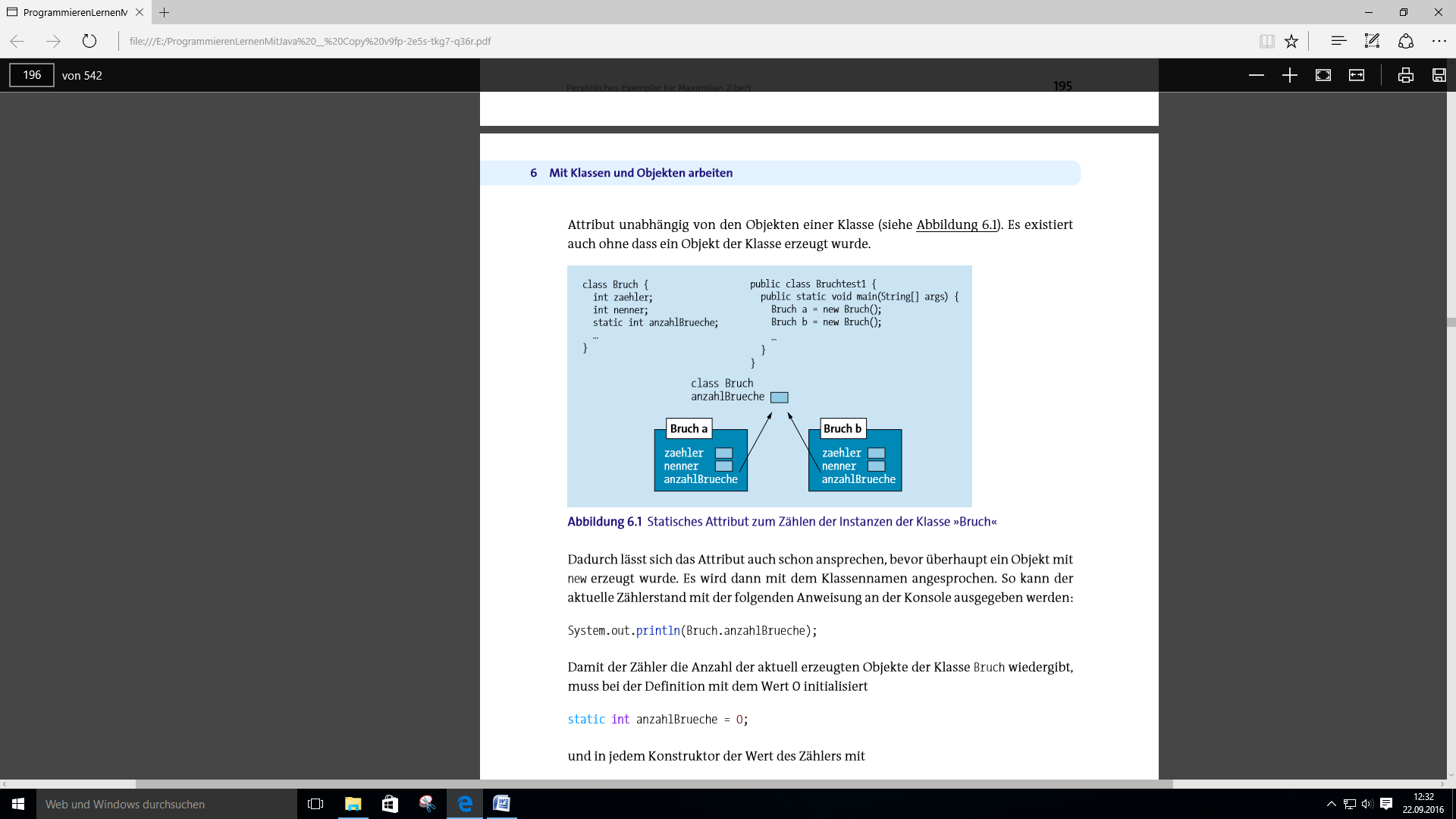
**Statische Attribute**

* Gehören zur Klasse
* Bedeutet: In einer Klasse als statisch definiertes Attribut unabhängig von der Anzahl der existierenden Objekte der Klasse nur ein einziges Mal pro Klasse sichtbar

static int anzahlBrueche;

* *static* = Modifier: Eigenschaften eines Attributs werden verändert
  + Macht ein Attribut unabhängig von den Objekten einer Klasse



* Aktueller Zählerstand kann mit folgenden Anweisungen an die Konsole ausgegeben werden:

*System.out.println(Bruch.anzahlBrueche);*

Damit Zähler die Anzahl der aktuell erzeugten Objekte der Klasse Bruch wiedergibt, muss bei der Definition mit dem Wert 0 initialisiert

*static int anzahlBrueche = 0;*

und in jedem Konstruktor der Wert des Zählers mit

*anzahlBrueche++;*

erhöht werden.

**Statische Methoden**

**Instanzmethoden**

* *static* modifizierte Methoden = Klassenmethoden
* Modifier sorgt dafür, dass betreffende Methode undabhängig von Objekten ist
  + Methode kann verwendet werden, ohne dass vorher ein Objekt mit *new* erzeugt wurde
* Wichtig für *main* Methode!
  + Bildet den Starpunkt eines Java-Programms

**Beispiel**:

* Legt Bruch Objekt an, mit dem der Instanzenzähler auf die Klasse Bruch zugreift

static int getAnzahlbrueche() {

return anzahlbrueche;

}

**Klassenmethoden**

* Können über Klassenbezeichner angesprochen werden, auf den ein Methodenbezeichner folgt
  + Durch Punkt voneinander getrennt
* Methode *getAnzahlbrueche* kann mit *Bruch*.*getAnzahlbrueche()* aufgerufen werden und liefert, bevor ein Objekt der Klasse existiert, den Wert 0 zurück

**Beispiel:**

public class BruchMitSeriennummer {

public static void main (String args []) {

System.out.println( “Anzahl der Brüche: “ + Bruch.getAnzahlBrueche());

Bruch a = new Bruch (1,2);

* Zugriff auf die Anzahl vor Erzeugung einer Instanz

**Zu Beachten:**

* Nur auf statische Datenelemente direkt zugreifen können
* Nur andere statische Methoden aufrufen können und
* *this* nicht verwenden können, weil es kein zugeordnetes Objekt geben kann.