gds2 Gottlieb-Daimler-Schule 2
Technisches Schulzentrum Sindelfingen
mit Abteilung Akademie für Datenverarbeitung

1 Implementieren und Anwenden eines Standardinterface

1.1. Erstellen Sie ein neues Package. In Moodle finden Sie die Zip-Datei DatenArtikel2SortUI.zip. Laden Sie die Datei herunter und entpacken sie diese. Die enthaltenen Klassen fügen Sie dem neuen Package hinzu.

1.2. Die Klasse `Artikel2` soll das Standard Interface `Comparable<Artikel2>` implementieren, so dass man dann das Array mit Artikeln in der Klasse `Artikel2SortUI` mit Hilfe der Standard-Klasse `Arrays` sortieren kann.

Das Array soll zuerst nach ArtikelNr aufsteigend sortiert werden.

1.3. Ändern Sie die `compareTo()` Methode so ab, dass die Artikel nach dem Verkaufspreis *absteigend* sortiert werden.

1.4. Ändern Sie die `compareTo()` Methode nochmals ab, dass die Artikel nach ihrer Bezeichnung *aufsteigend* sortiert werden.

```
public class Artikel2 implements Comparable<Artikel2>
{
    // Anfang Attribute
    private int iArtikelNr;
    private String sBezeichnung;
    private String sArtikelgruppe;
    private double dVerkaufsPreis;

    // ... weitere Attribute und Methoden ...

    @Override
    public int compareTo(Artikel2 o)
    {
        //return this.iArtikelNr - o.iArtikelNr;
         return (int)(o.dVerkaufsPreis - this.dVerkaufsPreis);
        //return (int)(100 * o.dVerkaufsPreis - 100 * this.dVerkaufsPreis);
        //return this.sBezeichnung.compareTo(o.sBezeichnung);
    }
    // Ende Methoden
}
```

1.5. Erstellen Sie ein neues Package und fügen Sie dem Package die Klasse Bruch hinzu (entweder Ihre selbst erstellte oder von Moodle). Ändern Sie die Klasse so, dass sie das Interface `Comparable<Bruch>` implementiert.
Überlegen Sie, wie man Brüche am besten auf ihre Anordnung vergleicht und implementieren Sie die Methode `compareTo()`

```
public class Bruch implements Comparable<Bruch>
{
    private int zaehler;
    private int nenner;

... // weitere Methoden

    @Override
    public int compareTo(Bruch andererBruch)
    //public int compareTo(Object andererBruch)
    {   // Die Brüche werden über ihren Dezimalwert verglichen
        double dBruch1, dBruch2;
        int iErgebnis = 0;

        // Brüche kürzen um sicher zu gehen, dass Dezimalwert gleich ist
        dBruch1 = this.kuerzen().ToDezimalZahl();
        dBruch2 = ((Bruch)andererBruch).kuerzen().ToDezimalZahl();

        if (dBruch1 > dBruch2)
        {
            iErgebnis = 1;
        }
        else
        {
            if (dBruch1 < dBruch2)
            {
                iErgebnis = -1;
            }
        }
       return iErgebnis;
    }
}
```

1.6. Erstellen Sie im selben Package eine StartUI Klasse zum Testen der Kasse `Bruch`:

Die StartUI Klasse erzeugt eine Array von Brüchen und initialisiert dieses direkt.
Beispiel:

```
Bruch[] aBrueche = {new Bruch(1,2), new Bruch(1,3), new Bruch(2,6),
                    new Bruch(96,124), new Bruch(1,4)};
```

Das Array soll dann sortiert werden und das Ergebnis der Sortierung am Bildschirm angezeigt werden.

gds2 Gottlieb-Daimler-Schule 2
Technisches Schulzentrum Sindelfingen
mit Abteilung Akademie für Datenverarbeitung

```
public class BruchSortUI
{
     * @param args
    public static void main(String[] args)
    {
        Bruch[] aBrueche = {new Bruch(1,22999999), new Bruch(1,23000000),
                            new Bruch(2,6), new Bruch(96,124),
                            new Bruch(1,4)};

        Arrays.sort(aBrueche);

        for (Bruch bruch : aBrueche)
        {
            System.out.println(bruch.ToString());
        }
    }
}
```