Serialisierung von Objekten und Persistierung im Dateisystem

Ziel: Objektdaten serialisiert im Dateisystem speichern und wieder einlesen

3.1. Gegeben ist ein Programm zur Erfassung und Anzeige von Artikeldaten. Der Quellcode dieses Programms findet sich in Moodle (SerializeArtikelContainer.zip → herunterladen → entpacken → in Eclipse in ein neues Package reinziehen). Das Programm ist lauffähig, aber noch nicht vollständig. Die Menüpunkte (5) und (6) haben keine Wirkung, da die entsprechenden Methoden zum Speichern der Artikel in einer Datei bzw. zum Lesen noch nicht ausprogrammiert sind

Diese Programm soll also dahin gehend ergänzt werden, dass man die erfassten Artikeldaten in einer Datei persistieren kann, um sie später wieder lesen zu können.

Wie man im Quellcode der Main-Methode bei `case 5` sieht, ist die Methode `speichereArtikelDaten()` eine Objektmethode der Klasse `ArtikelContainer` und muss also dort implementiert werden. Diese Methode ist dafür zuständig, die im `ArtikelContainer` gespeicherten Artikelobjekte zu persistieren.
Implementieren Sie die Methode und analysieren Sie die erzeugte Datei mit einem Hex-Editor (siehe Moodle).

Den Klassenkopf um `implements Serializable` ergänzen:

```
public class Artikel implements Serializable
{
    //...
}
```

In der Klasse ArtikelContainer, folgende Methode ergänzen:

```
public boolean speichereArtikelDaten(Path artikelDatei)
{
    boolean status = true;

    try (ObjectOutputStream oos =
            new ObjectOutputStream (
               new BufferedOutputStream(
                 Files.newOutputStream(artikelDatei,StandardOpenOption.CREATE,
                                                    StandardOpenOption.TRUNCATE_EXISTING))))
    {
        for (Artikel artikel : alleArtikel)
        {
            oos.writeObject(artikel);
        }

    }
    catch (Exception e)
    {
        status = false;
    }
    return status;
}
```

3.2. Ergänzen Sie nun das Programm noch um den notwendigen Code, um die Artikelobjekte wieder aus der Datei zu lesen.

Zum Testen beenden Sie das Programm und starten es neu, um dann die zuvor geschriebene Datei zu lesen.

```
public boolean leseArtikelDaten(Path artikelDatei)
{
    boolean status = true;
    Artikel neuerArtikel;
    int iAnzahl = 0;

    try (ObjectInputStream ois =
            new ObjectInputStream (
               new BufferedInputStream(
                 Files.newInputStream(artikelDatei))))
    {
        do
        {
            neuerArtikel = (Artikel)ois.readObject();
            this.alleArtikel.add(neuerArtikel);
            iAnzahl++;
        } while (true); // Endet durch EOF

    }
    catch (Exception e)
    {
        if (iAnzahl == 0)
        {
            status = false;
        }
    }
    return status;
}
```

3.3. Es gibt prinzipiell zwei Möglichkeiten den `ArtikelContainer` zu persistieren:

1. Die Klasse `ArrayList<Artikel>` komplett persistieren.
2. Die Klasse `Artikel` in einer Schleife einzeln persistieren.

Realisieren nun die Variante die Sie in Aufgabe 3.1/3.2 *nicht* gewählt hatten.

```java
public boolean speichereArtikelListe(Path artikelDatei)
{
    boolean status = true;

    try (ObjectOutputStream oos =
            new ObjectOutputStream (
               new BufferedOutputStream(
                 Files.newOutputStream(artikelDatei,StandardOpenOption.CREATE,
                                                    StandardOpenOption.TRUNCATE_EXISTING))))
    {
            oos.writeObject(alleArtikel);

    }
    catch (Exception e)
    {
        status = false;
    }
    return status;
}

@SuppressWarnings("unchecked")
public boolean leseArtikelListe(Path artikelDatei)
{
    boolean status = true;

    try (ObjectInputStream ois =
            new ObjectInputStream (
               new BufferedInputStream(
                 Files.newInputStream(artikelDatei))))
    {
        alleArtikel = null;
        alleArtikel = (ArrayList<Artikel>)ois.readObject();
    }
    catch (Exception e)
    {
        if (alleArtikel == null)
        {
            status = false;
        }
    }
    return status;
}
```