

Aufgabenblatt A07: Streams,

1 Java Denksportaufgaben

1. In der Mathematik definiert „=“ eine Äquivalenzrelation auf den reellen Zahlen. Aber hier geht es um Java! Definiert „==“ eine Äquivalenzrelation auf den Werten numerischer primitiver Typen in Java? Wenn Sie meinen „ja“, beweisen Sie dies bitte! Wenn Sie meinen „nein“, konstruieren Sie bitte Beispiele für die Eigenschaften, die verletzt sind, um dies zu belegen!
2. Im folgenden Code heißt ein Instanz-Attribut (field) einer Unterklasse wie ein Instanz-Attribut einer Oberklasse.
 - 2.1. Ist der Code legal?
 - 2.2. Welche Schwächen hat der Code ggfs?
 - 2.3. Wie machen Sie das ggfs. besser, wenn Sie in einer derartigen Situation sind und so etwas benötigen?

```
class Base {
    public String className = "Base";
}

class Derived extends Base {
    private String className = "Derived";
}

public class PrivateMatter {
    public static void main(String[] args) {
        System.out.println(new Derived().className);
    }
}
```

2 Generics und Streams

Implementieren Sie bitte das Interface *Deque* in einer generischen Klasse aus Aufgabenblatt a05 mittels eines Arrays mit den folgenden Vorgaben!

1. Die Elemente des Deques sollen in einem Array gespeichert werden (Array, **nicht ArrayList**)!
2. Die implementierende Klasse soll eine default Kapazität haben, so dass ein Objekt mit einem default constructor erzeugt werden kann.
3. Sie müssen darauf achten, dass das Array zur Speicherung der Elemente bei Bedarf vergrößert wird. Um eine möglicherweise sinnvolle Verkleinerung müssen Sie sich nicht kümmern, das ist eher ein Thema für das 3. Semester.
4. In Ergänzung zu den im Interface vorgegebenen Methoden soll die Klasse je eine Methode haben, die die Elemente des Deques in der Stack- bzw. der Queue-Reihenfolge als Stream liefern. Ergänzen Sie diese Methoden bitte in dem Interface! Wenn Sie es für sinnvoll halten, können Sie dem Interface auch weitere Elemente hinzufügen.

Für diese beiden Aufgaben ist der Abgabetermin für alle:

Mittwoch, 20.05.2020, 08:00

Fangen Sie bitte schon an, sich mit der folgenden Aufgabe zu beschäftigen. Zum Einlesen können Sie Tutorials (z.B. von Oracle, Gluon und vielen anderen) verwenden. Ich habe viel aus [9, 8] (beides in der elektronischen Bibliothek der HAW) u. a. gelernt.

3 Fraktale

Oracle liefert ein Beispiel für Fraktale (Mandelbrot-Menge) aus. Der im wesentlichen ursprüngliche Code steht im Lernraum zur Verfügung. Ursprünglich war dies ein Netbeans Projekts. Den Code sollten Sie leicht in IntelliJ oder Eclipse verwenden können.

Das ist nach bestem Wissen und Gewissen der unveränderte Source Code. Ich habe die Original-Quelle noch nicht wiedergefunden. Hier habe ich zwei Probleme, die aber die Nutzung nicht (ernsthaft) beeinträchtigen:

1. Die css-Datei kann in meinem Setup nicht geladen werden.
2. Die Laufzeitangaben werden nicht korrekt im Label positioniert.

Wenn Sie mit dem module-path arbeiten, müssen Sie das Paket, in dem Sie den Code bearbeiten in der module-info.java Datei exportieren (Siehe Kapitel Modules im Skript). Verwenden Sie den class path, sollte es direkt laufen. Ihre Aufgaben:

1. Mit dem Projekt kommt eine Klasse *Complex*. Diese ersetzen Sie bitte durch Ihre Version der Klasse *Complex* aus dem zweiten Aufgabenblatt.
2. Notieren Sie bitte vom Eindruck des Bildschirmaufbaus und der Zeitangaben Laufzeiten auf Ihrem Rechner, woei die Leistungsdaten Ihres Rechners:

Folgende Lernziele verbinde ich mit dieser Aufgabe:

- Klassen in ein bestehendes System eingliedern lernen, ggfs. unter Nutzung von Refactoring.
- Verwendung von Streams und Lambdas um an einem sinnvollen Beispiel sehen, wo die Verwendung etwas bringt.
- Generell: Mit fremdem Code umgehen können und ihn ggf. auch verbessern können.

Diese Aufgabe möchte ich auf jeden Fall im Praktikum besprechen!

Abgabe (gerne werfe ich auch einen Blick auf frühe Versionen!) bzw. Besprechung in Ihrem letzten Praktikumstermin.

Literatur

- [1] Hans Lauwerier. *Fraktale verstehen und selbst programmieren. Band 1: Einführung*. Wittich Fachbuchverlag, Hückelhoven, 2., verbesserte Auflage, 1992, 197 Seiten. ISBN-10 3-88984-060-4. Timmkoppel, Schinkenzimmer.
- [2] Hans Lauwerier. *Fraktale verstehen und selbst programmieren. Band 1: Vertiefung*. Wittich Fachbuchverlag, Hückelhoven, 2., verbesserte Auflage, 1992, 270 Seiten. ISBN-10 3-88984-061-2. Timmkoppel, Schinkenzimmer.
- [3] Benoît B. Mandelbrot. *Die fraktale Geometrie der Natur*. Birkhäuser, Basel, Boston, Berlin, Einmalige Sonderausgabe Auflage, 1991, 491 Seiten. ISBN-10 3-7643-2646-8. Timmkoppel, Schinkenzimmer.
- [4] Heinz-Otto Peitgen, Hartmut Jürgens und Dietmar Saupe. *Fractals for the Classroom – Part One*, Band 1. Springer-Verlag, Berlin, Heidelberg, New York, NY, 1992. Es gibt eine deutsche Übersetzung: [5].
- [5] Heinz-Otto Peitgen, Hartmut Jürgens und Dietmar Saupe. *Bausteine des Chaos – Fraktale*. Springer, Klett-Cotta, Berlin, Heidelberg, New York, Stuttgart, 1992. Timmkoppel, Bibliothekskeller, Faszinierend! Die Deutsche Übersetzung von [4].
- [6] Heinz-Otto Peitgen, Hartmut Jürgens und Dietmar Saupe. *Fractals for the Classroom – Part Two: Complex Systems and the Mandelbrot Set*, Band 2. Springer-Verlag, Berlin, Heidelberg, New York, NY, 1992. Timmkoppel, Bibliothekskeller, Wie [5] ein faszinierendes Buch.

- [7] Heinz-Otto Peitgen und Peter H. Richter. *The Beauty of Fractals*. Springer-Verlag, Berlin, Heidelberg, New York, NY, 1986. Timmkoppel, Ein wundervolles Buch!
- [8] Kishori Sharan. *Learn JavaFX: Building User Experience and Interfaces with Java 8*. Apres Media, New York, NY, 2015, 1210 Seiten. ISBN-10 978-1-4842-1142-7. [vorlesungen/pr/matrial/java/learnjavafx8.pdf](#), Electronic Edition.
- [9] Johan Vos, Weiqi Gao, Stephen Chin, Dean Iverson und James Weaver. *Pro JavaFX 8. A Definite Guide to Building Desktop, Mobile ,and Embedded Java Clients*. Apress, New York, NY, 2014, 604 Seiten. ISBN-10 978-1-4302-6575-7. [vorlesungen/pr/matrial/java/projavafx8.pdf](#), Electronic Edition.