

Programmierung 2 - Sommersemester 2019

Prof. Dr. Markus Esch

Übungsblatt Nr. 16 Abgabe KW 19

1. Aufgabe

Erweitern Sie die Klassen `PersonQueue` aus Übungsblatt 10 um einen internen Iterator. Der Iterator soll nur innerhalb der Klasse nutzbar sein und folgendes Interface implementieren:

```
interface PersonIterator extends java.util.Iterator<Person> { }
```

Erweitern Sie die Klasse `PersonQueue` um eine Methode `void print()`. Diese soll die gesamte Queue mithilfe des Iterators ausgeben. Erweitern Sie die Klasse `PersonQueue` um eine weitere Methode `String smallest()`. Diese soll die Person in der Warteschlange mit dem lexikalisch kleinsten Vornamen zurückgeben. Nutzen Sie hier ebenfalls den Iterator. Der Iterator sollte als innere Klasse realisiert werden. Überlegen Sie, welchen Typ einer inneren Klasse Sie am besten verwenden.

2. Aufgabe

Implementieren Sie eine Klasse `NumberCruncher`. Diese Klasse soll ein float Array besitzen. Über den Konstruktor wird die Größe des Arrays übergeben. Initialisiert wird das Array mit zufälligen Werten. Die Klasse soll eine Methode `crunch(String[] operations)` besitzen. Das übergebene String-Array beinhaltet Operationen, welche der Reihe nach auf dem float-Array der Klasse angewendet werden sollen.

Folgende Operationen sollen möglich sein:

- **sum**: Summiert die Elemente des Arrays paarweise von links nach rechts auf und speichert den neuen Wert in dem jeweils rechten Datenfeld. D.h.: $a[1] = a[0] + a[1]$; $a[2] = a[1] + a[2]$ usw.
- **swirl**: Führt n zufällige Vertauschungen der Datenfelder durch; n ist durch die Länge des float-Arrays gegeben.
- **divide**: Teilt die $n/2$ größten Werte im Array durch die $n/2$ Kleinsten und speichert den neuen Wert im Datenfeld des jeweils größeren Wertes. D.h. der größte Wert wird durch den Kleinsten geteilt. Der Zweitgrößte durch den Zweitkleinsten usw.
- **subtract**: Analog zu sum nur mit Substruktion.
- **average**: Bestimmt den Durchschnitt aller Werte im Array und speichert den Durchschnittswert im Datenfeld mit dem größten Wert.

- (a) Implementieren Sie die `NumberCruncher`-Klasse mithilfe von anonymen Klassen. D.h. die einzelnen Operationen sollen in Form von anonymen Klassen realisiert werden.

- (b) Implementieren Sie eine zweite Variante bei der die einzelnen Operationen in Form von Top-Level-Klassen realisiert werden. Vergleichen Sie Ihre beiden Lösungen und identifizieren Sie die wesentlichen Unterschiede hinsichtlich der Struktur.
- (c) Implementieren sie eine Testklasse.