Objektorientierte Programmierung

HS-Bochum

WS 19/20

Übung 3

Aufgabe 1)

Erklären Sie was das nachfolgende Listing tut.

```
Stream.of("a", "c", "d", "c", "c")
        .map(s -> { return s.toUpperCase(); })
        .filter(s -> { return s.startsWith("C"); })
                .forEach(System.out::println);
```

Formulieren Sie, falls möglich eine äquivalente jedoch effizientere Variante.

Aufgabe 2)

Gegeben ist der folgende Code:

```
public class Expression {

    public static <T extends String> List<T> doSomething(List<T> a)
    {
        List<T> l = a.size() != 0 ? new ArrayList<>() : null;

        a.stream().forEach( ii ->
                l.add( (T) a.stream().map(Integer::parseInt)
                .map(e -> {return e;})
                .flatMap( e -> Arrays.stream(new Integer[]{e,1}))
                .reduce( (i,j) -> i>j ? i+j : j )
                .get().toString() ) );

        return l;
    }

}


public static void main(String[] args) {

    ArrayList<String> a = new ArrayList<>() ;
    a.add("61");
    a.add("-4");
    a.add("1");
    final List<String> strings = Expression.doSomething(a);
    strings.stream().forEach(System.out::println);

}
```

Hierbei erstellt ‚Arrays.stream(X)' einen neuen Stream aus dem Array X.
Was ist die Ausgabe des Programms? Diskutieren Sie Möglichkeiten zur Verbesserung.

Aufgabe 3)
Beantworten Sie folgende Fragen:

I. Was ist eine weiterführende Operation bei Streams?
II. Was ist eine terminierende Operation bei Streams?
III. Erläutern Sie den Unterschied zwischen generischen und spezialisierten Streams.
IV. Kann ein Stream mehrfach ‚durchlaufen' werden?
V. Erklären Sie die Operationen map, filter, collect, flatMap, forEach und reduce.