

Matura POSxHIF 2023

Boigner Thomas, Cserich Philipp, Aslan Cemil (5BHIF)

Table of Contents

Ihr Auftrag!	2
smth	3
Enums	4
Stream types :	5
IntStream	5
Stream	5
Philipp Cserich	7
Service	8
Abstract	9
Sources	10

Ihr Auftrag!

Sehr geehrte Damen und Herren,

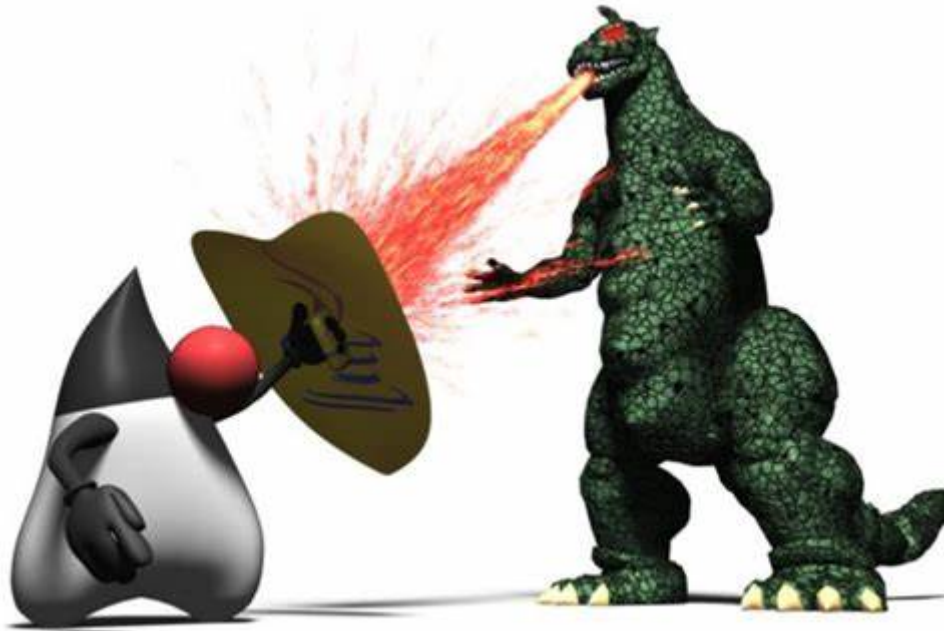
Dieses Dokument dient als Beilage zur Schriftlichen Matura der HTL Spengergasse für das Jahr 2023. Es wurde im Sinne der Vorbereitung für die schriftliche Reifeprüfung von Mitgliedern der 5BHIF erstellt, und dient Ihnen als Hilfestellung für den Antritt.

Ihre Mission startet JETZT!

Viel Erfolg bei der Matura!

(Das folgende Gerät wird in kürze selbstzerstört)

MfG, Aslan Cemil, Boigner Thomas, Cserich Philipp



smth

Enums

```
public enum Test{
    A(1),
    B(2),
    C(3)

    Test(Integer i){
        this.i = i;
    }

    private Integer i;
}
```

While normal enums are mostly used to differentiate predetermined / static values, they can be extended by adding parameters through a constructor.

Stream types :

- IntStream
- Stream

IntStream

- der IntStream ist ein Stream der zur Verarbeitung von Integer Werten dient dies kann zum Beispiel als ersatz für eine for Schleife dienen, wobei die Range der jeweiligen Werte durch folgende Syntax angegeben werden kann :

```
IntStream.range(0, 10).forEach(System.out::println);
```

Dieser Stream gibt die Werte von 0 bis 9 aus. Zur Verwendung des IntStreams für mappings anderer Objektklassen können unterschiedliche Methoden verwendet werden, wie zum Beispiel **mapToObj**. Dadurch kann der gestreamte wert innerhalb eines anderen Objektes verwendet bzw gespeichert werden oder eine methode wiederholt aufgerufen werden.

Stream

Streams können bei jeder Liste verwendet werden, wobei die Liste durch die Methode **stream()** in einen Stream umgewandelt wird. Dieser wird daraufhin sequentiell durchlaufen und kann durch verschiedene Methoden manipuliert werden.

- Filter

```
int minGrade = 80;

List<Student> filteredStudents = students.stream()
    .filter(s -> s.getGrade() >= minGrade)
    .toList();
```

- Collect

```
List<String> names = students.stream()
    .filter(s -> s.getAge() > 18 && s.getState().equals("Bayern"))
    .map(Student::getName)
    .collect(Collectors.toList())
```

Collect is used to convert a stream into a collection. In this case, we are converting the stream into a new list.

- Map

Mapping a new value onto an existing instance

Philipp Cserich

Service

- dto

Data transfer object used to transfer data between layers in your program

Sollte als record umgesetzt sein und einen Großteil der originaldaten des models enthalten. **KEINE** 1 zu 1 abbildung des models!

Anderer Aufbau = WICHTIG und RICHTIG :D

```
public record TeamDto(String name, String description, LocalDate creationDate,
List<Member> memberList) {
    public TeamDto(Team team){

this(team.getName(),team.getDescription(),team.getCreationDate(),team.getMembers());
        log.debug("TeamDto from Team : {} has been built !", team);
    }
}
```

- mutateCommand

Wird verwendet um die Daten vom Client zu erhalten und im Service in ein dto umzuwandeln. Form bzw Platzhalter für die Daten die zukünftig in das Model kommen sollen.

Path : Client → Controller → Service <<<

Abstract

word explanations and examples

Sources

- [1] <https://www.example.com>
- [2] <https://www.example.com>
- [3] <https://www.example.com>