

Musterlösung

# Lösungsblatt zur PiS-Klausur

Sie geben am Ende der Klausur nur das Lösungsblatt ab!

Nachname, Vorname	Matrikelnummer	Note
Unterschrift 21.07.2023		

## 1. Objektvergleich

equals:

```
@Override public boolean equals(Object other) { // 2
    if (other == null) return false; // 2
    if (other == this) return true; // 2
    if (other.getClass() != getClass()) return false; // 2
    Point that = (Point)other; // 2
    return this.x == that.x && this.y == that.y; // 2
}
```

hashCode:

```
@Override public int hashCode() { // 2
    return Objects.hash(x, y); // 2
}
```

hashCodeMin:

```
public int hashCodeMin() {
    return List.
        <Point>of(this, new Point(-x, -y), new Point(x, -y), new Point(-x, y)). // 2
        stream(). // 2
        mapToInt(Point::hashCode). // 2
        min(). // 2
        getAsInt(); // 2
}
```

toString:

```
@Override public String toString() { // 2
    return "(" + x + ", " + y + ")"; // 2
}
```

## 2. Stapelbares

Stackable

```
sealed interface Stackable<T> permits EmptyStack, StackWithElements { // 2
    default boolean isEmpty() {
        return this instanceof EmptyStack; // 2
    }
    default StackWithElements<T> push(T element) { // 2
        return new StackWithElements<T>(this, element); // 2
    }
    Stackable<T> pop(); // 2
    T top(); // 2
}
```

### StackWithElements

```
record StackWithElements<T>(Stackable<T> previous, T element) implements Stackable<T> { // 2
    public Stackable<T> pop() { return previous; } // 2
    public T top() { return element; } // 2
}
```

### EmptyStack

```
record EmptyStack<T>() implements Stackable<T> { // 2
    public Stackable<T> pop() { throw new UnsupportedOperationException(); } // 2
    public T top() { throw new UnsupportedOperationException(); } // 2
}
```

### of-Methode

```
@SafeVarargs static <T> Stackable<T> of(T... elements) { // 2
    Stackable<T> stack = new EmptyStack<T>(); // 2
    for(T element : elements) stack = stack.push(element); // 2
    return stack;
}
```

### toString

```
StackWithElements public String toString() { return pop() + " <- " + top(); }
EmptyStack public String toString() { return "["; }
```

## 3. Zahlensuche

### isDivisible

```
LongPredicate isDivisible(Integer... divisors) { // 2
    return n -> Arrays. // 2
        stream(divisors). // 2
        allMatch(i -> n % i == 0); // 2
}
```

### isValid

```
LongPredicate isValid = isDivisible(2,3,5); // 2
```

### Berechnung

```
LongStream. // 2
    rangeClosed(1, Long.MAX_VALUE). // 2
    filter(n -> isValid.test(n) && isValid.test(quer(n))). // 2
    findFirst(); // 2
```

### Ergebnis

```
$31 ==> OptionalLong[39990] // 2
```

## 4. Fragen

1.	2.	3.	4.	5.	Punkte	1.	2.	3.	4.	5.	Punkte
A	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	E	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
B	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	F	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
C	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	G	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
D	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	H	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Sollten Sie eine Lösung revidieren wollen, streichen Sie die betreffende Reihe durch und notieren Sie hier neben der Tabelle Ihre korrigierte Lösung im Stil von Z13: die hypothetische Frage Z, Antwort 1 und 3.

- 3 Punkte pro vollständig korrekt angekreuzte Antwort
- 2 Punkte, wenn bei mehr als einem Kreuz entweder durch Entfernen oder Hinzufügen eines Kreuzes eine vollständig korrekte Antwort entsteht
- 0 Punkte ansonsten