

TEIL 1

Was wird ausgegeben? Löst die Aufgaben auf Papier, bevor ihr den Code kompiliert.

FEHLERSUCHE

Verbessert, streicht und ergänzt zuerst.

```
Int a == 314;

while (a != 1) {
    if (a % 2 == 0) { a /__ 2;
    } else
        a = 3 * a + 1
    }

System._____.print(a);
```

KONTROLLFLUSS

```
for (int i = 10; i + 3 >= 2; i--) {
    System.out.print(2 * i + " ");
}
```

```
final int[] mysteriousArray = new int[10];

for (int i = 0; i < mysteriousArray.length; i++) {
    if (i < 2) {
        mysteriousArray[i] = 1;
    } else {
        mysteriousArray[i] = mysteriousArray[i - 1] + mysteriousArray[i - 2];
    }
}

for (final int a : mysteriousArray) {
    System.out.print(a + " ");
}
```

```
int x = 0;

for (int i = 10; i < 102; i += 2) {
    x += 2 * i;
}

System.out.print(x);
```

OBJEKTORIENTIERUNG

```
public class Foo {
    protected int x = 14;

    public void next() {
        x++;
    }

    @Override
    public String toString() {
        return Integer.toString(x);
    }
}
```

```
public class Bar extends Foo {
    @Override
    public void next() {
        super.next();
        x--;
    }
}
```

```
public class Baz extends Bar {
    @Override
    public void next() {
        x *= 2;
        super.next();
    }
}
```

```
final Foo foo = new Foo();
final Bar bar = new Bar();
final Bar baz = new Baz();
```

```
foo.next();
((Foo) bar).next();
bar.next();
baz.next();
```

```
System.out.print(foo + ", " + bar + ", " + baz);
```

Vervollständigt jeweils.

Implementiert eine Methode zur Berechnung der Fakultät natürlicher Zahlen rekursiv.

```
public static int factorial(final int n) {
```

(B)

Nähert die Eulersche Zahl $e \approx \sum_{i=0}^{20} \frac{1}{i!}$ an. Nutzt dazu die Methode aus **(A)**.

```
public static double euler() {
```

ATTRIBUTE

Ergänzt um

- eine private Konstante „message“ mit Wert "Hello, World!",
- ein Attribut „identifizier“ vom Typen int, dessen Wert einmalig im Konstruktor festgelegt wird,
- ein Attribut „name“ vom Typen String, das auch von ererbenden Klassen verwendet werden kann.

Die Attribute „identifizier“ und „name“ werden bereits im Konstruktor [...] initialisiert.

```
public classClazz {
```

```
// [...]
```

```
}
```