# Name:

# Quiz-Fragen zu Java:

**Hinweis: Wenn auf Klassen oder Methoden mit Namen verwiesen wird, dann sind diese im Codeanhang zu finden.**

1. Beantworten Sie die nachfolgenden Fragen zur Ausführung eines Methodenaufrufs der Methode „**gemuese**“.

Wie viele Zeilen werden insgesamt auf der Konsole ausgegeben?:

Wie lautete die vierte Zeile?:

Wie lautete die achte Zeile?:

1. Welchen Wert hat die Variable **„ch“** nach Ausführung der folgenden Java-Anweisungen?

char ch = ‘4‘;

ch += 2;

Antwort:

1. Welche der folgenden Aussagen zur UML sind richtig?

* Die UML ist eine grafische Sprache für die Beschreibung von Software-Systemen.
* In einem UML-Klassendiagramm werden Benutzt-Beziehungen mit einer geschlossenen Pfeilspitze (leeres Dreieck) notiert.
* Ein Objektdiagramm der UML zeigt Klassen und ihre Beziehungen zueinander.
* In einem Klassendiagramm der UML werden Klassen verkürzt dargestellt, um einen Überblick über Klassen und ihre Beziehungen zu haben.
* In einem UML-Klassendiagramm können Benutzt-Beziehungen (dependency) grafisch gut unterschieden werden von Enthält/-Verwaltet-Beziehungen(association).
* In einem Klassendiagramm der UML sollte im oberen Bereich des Kästchens, das eine einzelne Klasse darstellt, kein Name stehen, denn Klassen haben keine Namen.
* In der UML kann die Implementiert-Beziehung durch geeignete Pfeile dargestellt werden.
* Jedes Java-Programm kann auch in der UML programmiert werden.

1. Welche Ausgabe erscheint auf der Konsole, wenn die Methode „**doSomething**“ ausgeführt wird?

* AABCDEFG
* ABDF
* ACEG
* AACEG

1. Welche Ausgabe erscheint auf der Konsole, wenn die Methode „**doSomethingElse**“ ausgeführt wird?

* abcdefg
* AbCdEfG
* aBcDeFg
* ABCEDFG

1. Welches Ergebnis liefert diese Methode, wenn bei einem Aufruf als aktueller Parameter der Wert 4 übergeben wird?

public int m(int p) {

return (p==0) ? 0 : 2\*p + m(p-1);

}

* 10
* 15
* 20
* 30

1. Welche der folgenden Aussagen zu JUnit sind richtig?

* Für jeden Testfall in einer JUnit -Testklasse wird bei der Ausführung durch JUnit ein eigenes Exemplar der Testklasse erzeugt und an diesem die jeweilige Methode aufgerufen.
* Der Balken bei der Ausführung einer JUnit -Testklasse wird bei erst dann rot, wenn mehr als ein Test fehlschlägt.
* Bei der Ausführung einer JUnit -Testklasse durch JUnit wird immer nach dem ersten fehlgeschlagenen Testfall abgebrochen.
* Ein JUnit -Testfall überprüft, ob das beobachtete Verhalten eines Testobjektes mit dem erwarteten Verhalten übereinstimmt.
* Ein Testfall schlägt fehl, wenn bei der Ausführung einer der Prüfmethoden(assertEquals u.ä.) die Prüfbedingung nicht zutrifft.
* JUnit dient dem Testen von Java-Software, ist aber selbst in C++ geschrieben.

1. Welche Werte haben die Variablen nach Ausführung des folgenden Java-Quelltextes?

List<String> meineListe = new ArrayList<String>();

meineListe.add(0, “Zebra”);

meineListe.add(0, “Nashorn”);

meineListe.add(0, “Antilope”);

int laenge1 = meineListe.size();

meineListe.add(1, “Elefant”);

meineListe.remove(0);

int laenge2 = meineListe.size();

String s1 = meineListe.get(0);

String s2 = meineListe.get(1);

**laenge1**:

**laenge2**:

**s1**:

**s2**:

# Codeanhang:

## Methode „doSomething“:

public void doSomething() {

String wort = “ABCDEFG”;

String result = “A”;

for(int i = 0; i < wort.length; i += 2)

{

result += wort.charAt(i);

}

System.out.println(result);

}

## Methode „doSomethingElse“:

public void doSomethingElse() {

String wort = “ABCDEFG”;

String result = “”;

for(int i = 0; i < wort.length; ++i)

{

result = result + wort.charAt(i);

if(i % 2 == 0)

{

result = result.toUpperCase();

}

else

{

result = result.toLowerCase();

}

}

System.out.println(result);

}

## Methode „gemuese“:

public void gemuese() {

List<String> gemuesemix = new ArrayList<>();

gemuesemix.add(“Gurke”);

gemuesemix.add(“Paprika”);

gemuesemix.add(“Tomate”);

gemuesemix.add(“Zucchini”)

List<String> farbmix = new ArrayList<>();

farbmix.add(“gelb”);

farbmix.add(“rot”);

farbmix.add(“gruen”);

for(String gemuese : gemuesemix) {

for(String farbe : farbmix) {

System.out.println(farbe + “e “ + gemuese);

}

}

}