# Name:

# Quiz-Fragen zu Java:

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

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

int i = 2;

int[] intArray = {3, 4, 5, 6};

int[] anderesIntArray = intArray;

anderesIntArray[3] = 7;

i = intArray[3];

Antwort:

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

int[] intArray = new int[5];

for(int i : intArray)

{

i = 42;

}

j = intArray[0];

Antwort:

1. Welche der folgenden Aussagen sind für Java richtig?

* Wenn eine Klasse die in java.lang.Object für alle Klassen definierte Methode equals redefiniert, dann sollte auch hashCode so redefiniert werden, dass zwei laut equals gleiche Objekte den gleichen hashCode liefern.
* Wenn eine Klasse die in java.lang.Object für alle Klassen definierte Methode hashCode redefiniert, dann muss sie auch equals redefinieren.
* Wenn eine Klasse A eine Methode mit dem Methodenkopf boolean equals(A other) definiert, dann ist equals in der Klasse A überladen.
* Die in java.lang.Object für alle Klassen definierte Methode hashCode hat den Ergebnistyp String.

1. Welche der folgenden Aussagen sind richtig?

* Eine Liste enthält niemals Duplikate.
* Bei einer Liste mit *n* Elementen gibt es *n+1* verschiedene Positionen zum Einfügen eines weiteren Elements.
* Beim Aneinander ketten zweier Listen berechnet sich die Kardinalität der resultierenden Liste aus der Summe der Kardinalitäten der beiden Ursprungslisten.
* Bei einer Menge ist das Einfügen am Anfang teurer als das Einfügen am Ende.
* Beim Einfügen aller Elemente einer Liste in eine leere Menge ist die resultierende Kardinalität der Menge kleiner oder gleich der Kardinalität der Liste.
* Die Elemente einer Menge haben immer eine Ordnung, die vom Klienten definiert werden muss.
* Eine Menge enthält *niemals* Duplikate.
* Bei einer *nicht*-leeren Menge kann direkt auf das letzte Element zugegriffen werden.

1. Für welche Zeile im folgenden Java-Quelltext einer Java-Methode wird der Compiler eine Fehlermeldung ausgeben?
2. private int[] temperaturen = {12,15,18,13,15,1,3,7};
3. int donnerstag = temperaturen[3];
4. temperaturen = new int[temperaturen.length];
5. temperaturen[8] = donnerstag;
6. temperaturen = new int[8];
7. System.out.println(temperaturen[8]);

Antwort:

1. Gegeben sei die Klassendefinition der Klasse „**X**“. Welchen Wert hat die Variable „**b**“ nach Ausführung der folgenden Java-Anweisungen?

X x = new X();

boolean b = x.istLeer();

Antwort:

1. Fügen Sie in der folgenden Java-Methode in den Kopf der for-Schleife geeigneten Quelltext ein, so dass in einer erweiterten for-Schleife alle Elemente der Sammlung bearbeitet werden.

public int berechnePruefsumme(Set<Konto> konten) {

int summe = 0;

for( \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ){

summe += konto.gibNummer();

}

return summe;

}

1. Welche Aussagen zum folgenden Java-Quelltext sind korrekt?

int[] intArray = new int[100];

* **intArray** ist hier der Name einer Variablen.
* Diese Zeile deklariert nur eine Array-Variable, legt aber *kein* Array-Objekt an.
* **intArray** ist der Name des Typs von int[].
* int[] ist der Typ der Variablen **intArray**.

1. Welche Aussagen zum folgenden Java-Quelltext sind korrekt?

int[] intArray = {1,2,3,4};

int len = intArray.length;

int i = intArray[3];

* Der Wert der Variablen **len** nach Ausführung dieser Anweisung ist 4.
* Der Wert der Variablen **i** nach Ausführung dieser Anweisung ist 3.
* In der ersten Zeile wird ein Array-Objekt für 3 int-Werte angelegt.
* Der größte zulässige Index für Zugriffe auf die Elemente von **intArray** ist 3.

# Codeanhang:

## Klasse „X“:

class X {

private String[] \_strings;

public boolean istLeer()

{

return \_strings.length == 0;

}

}