

## Quizzies

Aktuelle Punktzahl: 0 / 13

Beantwortet: 0 / 14

Die Dokumentation beantwortet die Frage des "WAS" und nicht des "WIE", also was gemacht wird und nicht wie.

☒ Trifft zu

☐ Trifft nicht zu

➔ 1 Punkt

## Quizzies

Test beenden

Aktuelle Punktzahl: 1 / 13

Beantwortet: 1 / 14

Dank der Dokumentation ist es somit nicht nötig, den Quelltext zu lesen. Auch dann nicht, wenn Sie wissen wollen, wie eine bestimmte Methode der Klasse eine Aufgabe im Detail erledigt.

☐ Trifft zu

☒ Trifft nicht zu

➔ 1 Punkt

## Quizzies

Aktuelle Punktzahl: 2 / 13

Beantwortet: 2 / 14

Die Dokumentation ist hauptsächlich für Anfänger. Mit der nötigen Erfahrung erkennt man am Namen der Methode, wie sie zu verwenden ist.

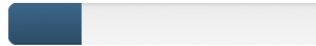
☒ Trifft nicht zu

☐ Trifft zu

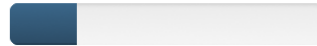
➔ 1 Punkt

## Quizzies

Aktuelle Punktzahl: 3 / 13



Beantwortet: 3 / 14



Unter anderem können folgende Informationen aus der Dokumentation gewonnen werden:

- Name der Klasse
- Beschreibung und Zweck der Klasse
- Liste der Konstruktoren und Methoden der Klasse
- Parameter und Ergebnistyp für jeden Konstruktor und jede Methode
- Beschreibung des Zwecks jedes Konstruktors und jeder Methode

☐ Trifft nicht zu

☒ Trifft zu

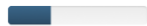
➔ 1 Punkt

Quizzies

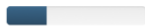
Test beenden

Test unterbrechen

Aktuelle Punktzahl: 4 / 13



Beantwortet: 4 / 14



Schauen Sie sich die Dokumentation der Klasse `java.util.Date` an. Der Konstruktor `Date(int year, int month, int date)` sowie die meisten Methoden haben die Anmerkung "Deprecated". Probieren Sie herauszufinden was es für Sie als Softwareentwickler bedeutet.

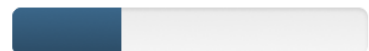
☐ Mit "Deprecated" Gekennzeichnetes sollte nicht mehr verwendet werden. Typischerweise weil es gefährlich ist oder eine bessere Alternative existiert.

☒ "Deprecated" bedeutet übersetzt u.a. "überholt". Es ist somit ein Hinweis, dass die Funktionalität zur Vorgängerversion von Java überarbeitet/verbessert wurde und man ab sofort die "Deprecated" Varianten verwenden soll.

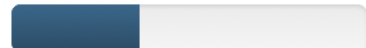
➔ 0 Punkte

## Quizzies

Aktuelle Punktzahl: 4 / 13



Beantwortet: 5 / 14



Der Gleichheitsoperator (`==`) überprüft, ob zwei Referenzen identisch sind.

☐ Trifft nicht zu

☒ Trifft zu

➔ 1 Punkt

Aktuelle Punktzahl: 5 / 13

Beantwortet: 6 / 14

Um zwei Strings zu vergleichen beschreibt das Buch, dass der "korrekte Quelltext" wie folgt aussieht:

```
if (eingabe.equals("ade")) {  
    ...  
}
```

Tatsächlich ist es aber weit sinnvoller den Vergleich wie folgt durchzuführen:

```
if ("ade".equals(eingabe)) {  
    ...  
}
```

☐ Der folgende Vergleich ist in seiner Funktion equivalent zur zweiten Variante.

```
if (eingabe.equals("ade") && eingabe != null) {  
    ...  
}
```

☒ Ist die Variable eingabe nicht initialisiert (also null) so führt der erste Vergleich zu einer NullPointerException. Beim zweiten Vergleich ist das Resultat wie erwartet false.

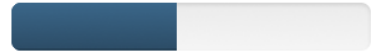
☒ Der folgende Vergleich ist in seiner Funktion equivalent zur zweiten Variante.

```
if (eingabe != null && eingabe.equals("ade")) {  
    ...  
}
```

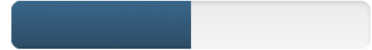
➔ 1 Punkt

## Quizzies

Aktuelle Punktzahl: 6 / 13



Beantwortet: 7 / 14



Welche Antwort stimmt im Zusammenhang mit dem folgende Codeausschnitt?

```
String abc = "abc";  
String abcMitLeerzeichen = "abc ";  
String abcTrim = abcMitLeerzeichen.trim();  
  
if (abc.equals(abcTrim)) {  
    System.out.println("true");  
} else {  
    System.out.println("false");  
}
```

- ☐ Die Ausgabe des obigen Codes lautet false
- ☐ Beim Ausführen tritt eine Exception auf bevor es zu einer Ausgabe kommt.
- ☒ Die Ausgabe des obigen Codes lautet true

➔ 1 Punkt

## Quizzies

Aktuelle Punktzahl: 7 / 13



Beantwortet: 8 / 14



Zahlen aus welchem Wertebereich generiert der folgende Code?

```
Random zufallszahlenGenerator = new Random();  
int zufallszahl = zufallszahlenGenerator.nextInt(6);
```

- ☒ { 0, 1, 2, 3, 4, 5 }
- ☐ { 0, 1, 2, 3, 4, 5, 6 }
- ☐ { 1, 2, 3, 4, 5, 6 }
- ☐ { 1, 2, 3, 4, 5 }

➔ 1 Punkt

## Quizzies

Aktuelle Punktzahl: 8 / 13



Beantwortet: 9 / 14



Welche Zufallszahlen werden mit folgendem Code generiert?

```
Random zufallszahlenGenerator = new Random();  
double zufallszahl = zufallszahlenGenerator.nextDouble() * 10;
```

Hinweis: Schauen Sie sich die ausführliche Dokumentation der nextDouble() Methode an.

- ☐  $0 < X < 10$
- ☐  $0 < X \leq 10$
- ☒  $0 \leq X < 10$
- ☐  $0 \leq X \leq 10$

➔ 1 Punkt

## Quizzies

Aktuelle Punktzahl: 9 / 13



Beantwortet: 10 / 14



Eine Klassendefinition beginnt wie folgt:

```
public class SchluesselWertPaare<S,W>
{
    public void hinzufuegen(S schluessel, W wert) {
        ...
    }
    ...
}
```

Objekte welchen Typs kann die Methode hinzufuegen entgegennehmen? Markieren Sie die korrekte(n) Antwort(en)!

- ☐ Nur Objekte der Klassen S und W, die genau so im Programm vorhanden sein müssen: public class S {...} und public class W {...}.
- ☐ Nur vom Typ String
- ☐ Keine. Solche Klassendefinitionen gibt es nicht.
- ☒ Objekte der Typen, die beim Erzeugen des SchluesselWertPaare-Objekts für S und W festgelegt wurden.

➔ 1 Punkt

- ☐ Schlüsselwerte unklar

## Quizzies

Aktuelle Punktzahl: 10 / 13



Beantwortet: 11 / 14



Der folgende Code importiere alle Klassen im Paket java.util. Trifft diese Aussage zu?

```
import java.util.*;
```

- ☒ Ja
- ☐ Nein

➔ 1 Punkt

## Quizzies

Aktuelle Punktzahl: 11 / 13



Beantwortet: 12 / 14



Welche Imports sind korrekt, wenn man die Klasse `java.util.ArrayList` verwenden möchte?

- ☐ `import ALL;`
- ☐ `import ArrayList;`
- ☒ `import java.util.*;`
- ☒ `import java.util.ArrayList;`

➔ 1 Punkt