



# Dungeon Crawl – Tag 4

28. November 2024



# Warm-up

---



# Quiz

---

You are a software developer for Example Corp where you are creating a new Java package. You need to come up with a name for the new package.

Which line of code follows the Java convention for naming packages?

**Choose the correct answer:**

- a) `package ExampleCorp;`
- b) `package exampleCorp;`
- c) `package corpExample;`
- d) `package com.example;`
- e) `package example.com;`
- f) `package www.example.com;`



# Quiz

---

You are a software developer for Example Corp where you are creating a new Java package. You need to come up with a name for the new package.

Which line of code follows the Java convention for naming packages?

**Choose the correct answer:**

- a) `package ExampleCorp;`
- b) `package exampleCorp;`
- c) `package corpExample;`
- d) `package com.example;`
- e) `package example.com;`
- f) `package www.example.com;`



Das `com` in einem Package-Namen steht für "Commercial" und ist Teil der Reverse-Domain-Notation, die verwendet wird, um Java-Packages eindeutig zu kennzeichnen, indem der Domainname des Unternehmens oder der Organisation in umgekehrter Reihenfolge verwendet wird.



# Quiz

---

Was ist der Unterschied zwischen == und equals() in Java?

Wähle die richtige Antwort aus:

- a) == vergleicht den Inhalt der Strings, während equals() die Referenz vergleicht.
- b) == vergleicht die Referenzen von str1 und str2, während equals() den Inhalt der Strings vergleicht.
- c) == und equals() sind in ihrem Verhalten identisch und vergleichen den Inhalt der Strings.
- d) Der Code wird zu einem Fehler führen, weil == und equals() nicht miteinander verglichen werden können.

```
public class VergleichTest {  
    public static void main(String[] args) {  
        String str1 = new String("Hallo");  
        String str2 = new String("Hallo");  
  
        if (str1 == str2) {  
            System.out.println("str1 == str2");  
        } else {  
            System.out.println("str1 != str2");  
        }  
  
        if (str1.equals(str2)) {  
            System.out.println("str1 equals str2");  
        } else {  
            System.out.println("str1 does not equal str2");  
        }  
    }  
}
```



# Quiz

---

Sieh dir den folgenden Code an. Was ist der Unterschied zwischen der Verwendung von Random und Math.random() im Programm, und welche Werte werden ausgegeben?

- a) Random erzeugt Zufallszahlen zwischen 0 und 1, während Math.random() Zahlen im Bereich von 0 bis 10 erzeugt.
- b) Random erzeugt Zufallszahlen als Ganzzahlen, während Math.random() Zufallszahlen als Dezimalzahlen (double) erzeugt.
- c) Math.random() ist die bevorzugte Methode für Zufallszahlen, weil sie schneller ist als Random.
- d) Beide Methoden erzeugen dieselben Zufallszahlen, aber Math.random() ist besser, weil es keine Instanz benötigt.

```
public class ZufallTest {  
    public static void main(String[] args) {  
        // Verwendung der Random-Klasse  
        Random random = new Random();  
        int randomInt = random.nextInt(10); // Zufallszahl zwischen 0 und 9  
  
        // Verwendung der Math.random()-Methode  
        double randomDouble = Math.random(); // Zufallszahl zwischen 0.0 und 1.0  
  
        System.out.println("Zufallszahl mit Random: " + randomInt);  
        System.out.println("Zufallszahl mit Math.random(): " + randomDouble);  
    }  
}
```

# Plan für die Woche

---

Montag

- Entscheidungsbäume
- Packages/Importe
- Abstrakte Klassen

Dienstag

- Logik
- Switch-case

Mittwoch

- Random
- Math Class

Donnerstag

- Scanner
- Rekursion von Methoden
- Codequalität

Freitag

- Datentypen II
- Wrapper: Boolean, Double, Integer
- Casten

# Plan für heute

---

- Scanner
- Rekursion von Methoden
- Codequalität



# Scanner

---

# Scanner

---

- java.util – Package
- seit Java 5
- Lesen und schreiben von Daten



System.in steht dafür, dass Eingaben von der Konsole eingelesen werden sollen.

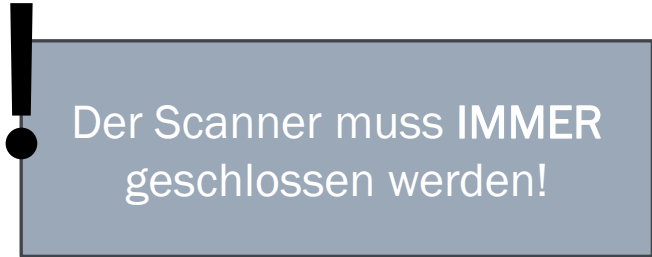
```
Scanner scanner = new Scanner(System.in);
```

Alle Details und Funktionen unter: [Scanner \(Java Platform SE 8 \)](#)

# Scanner-Methoden

---

- `nextLine()` – Liest eine komplette Zeile als String.
- `nextInt()` – Liest eine Ganzzahl (int).
- `nextDouble()` – Liest eine Fließkommazahl (double).
- `next()` – Liest das nächste Wort (bis zum ersten Leerzeichen).
- `hasNext()` – Prüft, ob noch eine Eingabe vorhanden ist.
- `close()` – Schließt den Scanner, um Ressourcen freizugeben.



Der Scanner muss **IMMER** geschlossen werden!

# Umgang mit Scanner

---

- Fehlerbehandlung u.a. falscher Nutzereingaben
- Der Scanner muss immer geschlossen werden

# Rekursion von Methoden

---

# Rekursion

---

- Methoden rufen sich selbst auf
- Aufbau einer rekursiven Methode:
  - Stopp der Rekursion
  - Rekursionsaufruf
- Brauchen häufig mehr Speicher als Iterationen und können langsamer sein
- Wird kein Stopp der Rekursion implementiert, kann es zum StackOverflowError kommen



# Intellij-Aufgabe

---

# Aufgabe

---



1. Implementiere die Berechnung von  $n!$  in einer rekursiven Methode.

**Idee:**  $n! = 1 * 2 * 3 * 4 * 5 * \dots * n$  (Alle natürlichen Zahlen bis zu einer bestimmten Zahl multiplizieren.)

- a) Was passiert bei  $n=0$ ? (**Hinweis:**  $0!=1$  ist mathematisch definiert.)
- b) Wie kann man bei der Berechnung der Funktion vorgehen?
- c) Berechne exemplarisch  $15!$  und  $2!$ .





# Aufgabe

---

2. Implementiere die Berechnung von den Fibonacci-Zahlen in einer rekursiven Methode.

**Idee:** Die Fibonacci-Zahlen sind eine Zahlenfolge, wo sich die nachfolgende Zahl durch die Summe der beiden vorherigen ergibt:  $F(0)=0$ ,  $F(1)=1$ ,  $F(n)=F(n-1)+F(n-2)$  für  $n > 1$ .

- a) Was passiert bei  $n=0$  und bei  $n=1$ ?
- b) Wie kann man bei der Berechnung der Funktion vorgehen?
- c) Berechne exemplarisch  $F(12)$  und  $F(100)$ .

# Codequalität

---



# Was bedeutet für euch Codequalität?

---

[HTTPS://WWW.MENTI.COM/AL4ZAKA3425H](https://www.menti.com/AL4ZAKA3425H)

# Code-Qualitätsstandards

---

- Lesbarkeit
- Wartbarkeit
- Fehlervermeidung
- Effizienz
- Testbarkeit

# Code-Qualitätsstandards

---

## Lesbarkeit:

- Aussagekräftige Benennung von Variablen, Methoden und Klassen
- Kommentare, der Code sollte möglichst selbsterklärend sein
- Einheitliche Formatierung

# Code-Qualitätsstandards

---

## Wartbarkeit:

- Wiederverwenden anstatt Duplikate
- Kleine Funktionen
- Verwendet Konstanten oder Variablen anstatt fester Werte im Code

# Code-Qualitätsstandards

---

## Fehlervermeidung:

- Überprüfe ob alle Fehlerfälle abgedeckt sind
- Null-Überprüfung
- Ressourcen richtig nutzen

# Code-Qualitätsstandards

---

## Effizienz:

- Vermeidung unnötiger Berechnungen
- Verwendung geeigneter Datenstrukturen



# Code-Qualitätsstandards

---

## Testbarkeit:

- Erstellen von Tests
- Code Reviews

# Code-Qualitätsstandards

---

Beispiele für Konventionen:

<https://proglang.informatik.uni-freiburg.de/teaching/java/2017/code-quality-standards.html>

<https://aws.amazon.com/de/what-is/code-quality/>

<https://google.github.io/styleguide/javaguide.html>



# Dungeon Crawl

---