



Dungeon Crawl – Tag 2

26. November 2024

Plan für die Woche

Montag

- Entscheidungsbäume
- Packages/Importe
- Abstrakte Klassen

Dienstag

- Logik
- Switch-case

Mittwoch

- Random
- Math Class

Donnerstag

- Rekursion von Methoden

Freitag

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

Plan für heute

- Logik
- switch-case

Logik und Operatoren

== oder equals()

==

- Vergleich von Referenzen von Objekten
- **true**: Beide Objekte auf dasselbe Speicherobjekt verweisen
- Bei primitiven Datentypen erfolgt ein direkter Vergleich

equals()

- Vergleicht der Inhalt von Objekten
- In der *Object-Klasse* definiert, kann aber überschrieben werden



Intellij-Aufgabe

Logische Operatoren

Operator	Beschreibung	Kurzbeispiel
!	Negation, invertiert den Ausdruck	<code>boolean lügnerSpricht = !wahrheit;</code>
&&	Und, <code>true</code> , genau dann wenn alle Argumente <code>true</code> sind	<code>boolean krümelmonster = istBlau && magKekse;</code>
	or <code>true</code> , wenn <i>mindestens</i> ein Operand <code>true</code> ist	<code>boolean machePause = hungrig durstig;</code>
^	Xor <code>true</code> wenn genau ein Operand <code>true</code> ist	<code>boolean zustandPhilosoph = denkt ^ ist;</code>



Intellij-Aufgabe

Negation (!)

Bedingung	Ergebnis (!)
true	false
false	true

AND-Operator (&&)

Bedingung 1	Bedingung 2	Ergebnis
true	true	true
true	false	false
false	true	false
false	false	false

OR-Operator (||)

Bedingung 1	Bedingung 2	Ergebnis
true	true	true
true	false	true
false	true	true
false	false	false

XOR-Operator (^)

Bedingung 1	Bedingung 2	Ergebnis
true	true	false
true	false	true
false	true	true
false	false	false



Quiz

Was gibt der folgende Code aus, wenn der Benutzer 20 Jahre alt und Mitglied ist?

- a) "Zugang gewährt"
- b) "Zugang verweigert"
- c) Es gibt einen Fehler im Code
- d) Der Code läuft unendlich

```
public class AndOperator {  
    public static void main(String[] args) {  
        int alter = 20;  
        boolean istMitglied = true;  
  
        if (alter >= 18 && istMitglied) {  
            System.out.println("Zugang gewährt");  
        } else {  
            System.out.println("Zugang verweigert");  
        }  
    }  
}
```



Quiz

Was gibt der folgende Code aus, wenn der Benutzer kein Admin ist, aber Zugang hat?

- a) "Zugang gewährt"
- b) "Zugang verweigert"
- c) Es gibt einen Fehler im Code
- d) Der Code läuft unendlich

```
public class OrOperator {  
    public static void main(String[] args) {  
        boolean istAdmin = false;  
        boolean hatZugang = true;  
  
        if (istAdmin || hatZugang) {  
            System.out.println("Zugang gewährt");  
        } else {  
            System.out.println("Zugang verweigert");  
        }  
    }  
}
```



Quiz

Was gibt der folgende Code aus, wenn der Benutzer 30 Jahre alt, ein Passwort und eine Benutzer-ID hat und kein Admin ist?

- a) "Zugang gewährt"
- b) "Zugang verweigert"
- c) Der Code gibt einen Fehler zurück
- d) Der Code läuft unendlich

```
public class KomplexeLogik {  
    public static void main(String[] args) {  
        int alter = 30;  
        boolean hatPasswort = true;  
        boolean hatBenutzerID = true;  
        boolean istAdmin = false;  
  
        if ((alter > 18 && (hatPasswort ^ hatBenutzerID)) || istAdmin) {  
            System.out.println("Zugang gewährt");  
        } else {  
            System.out.println("Zugang verweigert");  
        }  
    }  
}
```

Switch-case

Switch-Case

- Prüfung verschiedener Bedingungen

```
int tag = 2;

switch (tag) {
    case 1:
        System.out.println("Montag");
        break;
    case 2:
        System.out.println("Dienstag");
        break;
    case 3:
        System.out.println("Mittwoch");
        break;
    default:
        System.out.println("Ungültiger Tag");
}
```



Info: Bis Java 7 nur primitive Datentypen und enum-Werte als switch-Bedingung möglich! Und keine Strings für die Bedingungen. Eine **break-Anweisung** ist nach jedem Fall notwendig!

Switch-Case

- Prüfung verschiedener Bedingungen mit Strings ab Java 8 möglich

```
String tag = "Dienstag";

switch (tag) {
    case "Montag":
        System.out.println("Erster Tag der Woche");
        break;
    case "Dienstag":
        System.out.println("Zweiter Tag der Woche");
        break;
    case "Mittwoch":
        System.out.println("Dritter Tag der Woche");
        break;
    default:
        System.out.println("Ungültiger Tag");
}
```

Switch-Case

– ab Java 8: „Arrow-Case“ – Lambda-Ausdrücke („Methodenverkürzungen“)

```
int zahl = 2;

// Switch mit Lambda-Ausdrücken
switch (zahl) {
    case 1 -> System.out.println("Eins");
    case 2 -> System.out.println("Zwei");
    case 3 -> System.out.println("Drei");
    default -> System.out.println("Ungültig");
}
```

Switch-Case

– ab Java 12: „Arrow“-Syntax

```
int zahl = 2;

// Switch-Expression mit Arrow-Syntax (ab Java 12)
String ergebnis = switch (zahl) {
    case 1 -> "Eins";
    case 2 -> "Zwei";
    case 3 -> "Drei";
    default -> "Ungültig";
};

System.out.println("Das Ergebnis ist: " + ergebnis);
```

Switch-Case

- mehrere Fälle in einem Case

```
String monat = "Januar"; // Beispiel: "Januar"

String saison = switch (monat) {
    case "Dezember", "Januar", "Februar" -> "Winter";
    case "März", "April", "Mai" -> "Frühling";
    case "Juni", "Juli", "August" -> "Sommer";
    case "September", "Oktober", "November" -> "Herbst";
    default -> "Ungültiger Monat";
};

System.out.println("Die Saison im " + monat + " ist: " + saison);
```



Aufgabe

1. Schreibe ein Java-Programm, das basierend auf einer Zahl von 1 bis 7 den entsprechenden Wochentag ausgibt. Verwende die **switch-case**-Anweisung. Wenn die Zahl außerhalb des Bereichs von 1 bis 7 liegt, soll das Programm „Ungültige Zahl“ ausgeben.
2. Schreibe ein Java-Programm, das den Monat als Zahl (1 bis 12) entgegennimmt und die entsprechende Jahreszeit ausgibt.
3. Schreibe ein Java-Programm, das basierend auf dem Wochentag als String (z. B. "Montag", "Dienstag") die Anzahl der Stunden bis zum Wochenende anzeigt. Verwende den switch-case-Befehl und setze die richtige Anzahl der Stunden für jeden Wochentag.

Quellen

https://de.wikibooks.org/wiki/Java_Standard:_Operatoren