

Aufgabe 14.02.2025 – Lambdas

1. Aufgabe: Temperatur converter:

Lösungsschritte:

1. **Erstelle ein Interface TemperatureConverter**, das eine abstrakte Methode `convert(double temperature)` enthält, die einen `double` zurückgibt.
2. **Erstelle die Klassen CelsiusToFahrenheit, FahrenheitToCelsius, CelsiusToKelvin und KelvinToCelsius**, die das Interface implementieren und die `convert`-Methode überschreiben. Jede `convert` Methode soll die passende Formel anwenden und das Ergebnis dann zurückgeben.
3. **Erstelle eine Test-Klasse**, in der du die Klassen ausprobierst.
4. **Nutze in der Test-Klasse einen Lambda-Ausdruck**, um ein `TemperatureConverter`-Objekt zu erstellen, das Kelvin in Fahrenheit umrechnet. In den Lambda Befehl muss also die Formel zur Umrechnung von Kelvin zu Fahrenheit rein.

2. Aufgabe: Prüfe, ob eine Zahl positiv ist:

Schreibe ein `Predicate<Integer>`, das prüft, ob eine Zahl größer als 0 ist. Teste es mit den Zahlen -5, 0 und 10 in einer **ArrayList**.

Lösungsschritte:

1. Erstelle eine **ArrayList** mit den Zahlen -5, 0, 10.
2. Definiere ein `Predicate<Integer>`, das `true` zurückgibt, wenn eine Zahl größer als 0 ist.
3. Iteriere durch die Liste und teste jede Zahl mit `test()`.

3. Aufgabe: Prüfe, ob ein String leer ist:

Schreibe ein `Predicate<String>`, das prüft, ob ein String leer ("") ist. Gegeben ist folgende **ArrayList** { "Hello", "test", "", "!" }. Überprüfe, ob die Liste einen leeren String enthält.

Lösungsschritte:

1. Erstelle eine **ArrayList** mit den Strings.
2. Definiere ein `Predicate<String>`, das `true` zurückgibt, wenn ein String "" ist.

3. Durchlaufe die Liste mit einer Schleife und wende das Predicate mit test() an.

4. Aufgabe: Prüfe, ob eine Zahl zwischen 20 und 40 liegt:

Schreibe ein Predicate<Integer>, das prüft, ob eine Zahl zwischen 20 und 40 liegt.
Teste es mit den Zahlen 5, 33, 39.

Lösungsschritte:

1. Erstelle eine **ArrayList** mit den Testzahlen.
2. Definiere ein Predicate<Integer>, das true zurückgibt, wenn die Zahl zwischen 20 und 40 liegt.
3. Teste jede Zahl mit einer Schleife.

5. Aufgabe: Prüfe, ob eine Zahl gerade ist:

Schreibe eine Methode getEvenNumbers(ArrayList<Integer> zahlen), die nur gerade Zahlen als ArrayList zurückgibt.

Nutze ein Predicate<Integer> zum Testen.

Lösungsschritte:

1. Erstelle eine ArrayList mit 5 verschiedenen Zahlen
2. Schreibe die Methode und übergebe die ArrayList als Parameter
3. Erstelle ein Predicate, was **true** zurückgibt, wenn eine Zahl gerade ist
4. Gehe die ArrayList durch und prüfe jede Zahl, ob sie gerade oder ungerade ist mit deinem Predicate und test(). Speichere gerade Zahlen in einer neuen ArrayList
5. Gib die ArrayList mit den geraden Zahlen zurück