

12 Methoden

Einfache Übungen

Schreiben Sie eine Methode für die folgenden Aufgaben:

1. Berechnen des Mittelwertes dreier ganzer Zahlen.
2. Bestimmen der Anzahl der Tage eines Monats. (Übergabe von Monatsname).
3. Ausgabe des i-ten Zeichens einer Zeichenkette genau n-Mal.
4. Berechnen des Funktionswertes von $\sqrt{x^2 + y^2}$.
5. Schreiben Sie eine Methode, die einen String als Parameter nimmt und die Länge des Strings ausgibt.
6. Schreiben Sie eine Methode, die einen Integer-Array als Parameter nimmt und den Durchschnitt der Werte berechnet und ausgibt.
7. Schreiben Sie eine Methode, die einen String als Parameter nimmt und prüft, ob der String nur aus Zahlen besteht => Rückgabewert bool.
8. Schreiben Sie eine Methode, die einen Integer-Array als Parameter nimmt und den größten Wert im Array zurückgibt.
9. Schreiben Sie eine Methode, die einen Integer-Array als Parameter nimmt und prüft, ob der Array aufsteigend sortiert ist.
10. Schreiben Sie eine Methode, die einen Integer-Array als Parameter nimmt und den Index des ersten Vorkommens einer bestimmten Zahl zurückgibt.
11. Schreiben Sie eine Methode, die einen String als Parameter nimmt und alle Vokale im String zählt und zurückgibt.
12. Schreiben Sie eine Methode, die zwei Strings als Parameter nimmt und prüft, ob sie an der gleichen Stelle den gleichen Buchstaben haben.
13. Schreiben Sie eine Methode, die einen Integer als Parameter nimmt und prüft, ob die Zahl eine Primzahl ist.

Weitere Übungen

1. Schreiben Sie eine Methode, die einen Integer-Array als Parameter nimmt und die Summe aller geraden Zahlen im Array zurückgibt.
2. Schreiben Sie eine Methode, die einen String als Parameter nimmt und den umgekehrten String zurückgibt.
3. Schreiben Sie eine Methode, die einen Double-Array als Parameter nimmt und den Durchschnitt der Werte berechnet und zurückgibt.
4. Schreiben Sie eine Methode, die einen Integer als Parameter nimmt und prüft, ob die Zahl eine perfekte Zahl ist (d.h. die Summe ihrer Teiler ohne die Zahl selbst ergibt die Zahl).
5. Schreiben Sie eine Methode, die einen String als Parameter nimmt und den String in Kleinbuchstaben umwandelt.
6. Schreiben Sie eine Methode, die einen Integer-Array als Parameter nimmt und den Index des letzten Vorkommens einer bestimmten Zahl zurückgibt.
7. Schreiben Sie eine Methode, die einen String als Parameter nimmt und prüft, ob der String ein Palindrom ist, ohne die Leerzeichen zu beachten.
8. Schreiben Sie eine Methode, die einen Double-Array als Parameter nimmt und den größten Wert im Array zurückgibt.

9. Schreiben Sie eine Methode, die einen String als Parameter nimmt und die Anzahl der Wörter im String zählt und zurückgibt.
10. Schreiben Sie eine Methode, die zwei Strings als Parameter nimmt und prüft, ob sie an der gleichen Stelle unterschiedliche Buchstaben haben.

Fortgeschrittene Übungen

1. Schreiben Sie eine Methode, die einen String als Parameter nimmt und eine neue Zeichenkette zurückgibt, in der jeder Buchstabe um eine bestimmte Anzahl von Positionen im Alphabet verschoben wird. Verwenden Sie dazu die ASCII-Werte der Buchstaben.
2. Schreiben Sie eine Methode, die einen Text als Parameter nimmt und die Anzahl der Wörter zählt, die mit einem bestimmten Präfix (zB.: **vor** in "Ein Haus in eimen Vorstadt vor der vorderen Brücke" => 3 x vor) beginnen.
3. Schreiben Sie eine Methode, die einen String als Parameter nimmt und prüft, ob der String ein gültiger Palindrom-Satz ist. Dabei sollen Groß- und Kleinschreibung, Satzzeichen und Leerzeichen ignoriert werden

ISBN

Schreiben Sie eine Methode, die einen String als Parameter nimmt und prüft, ob der String eine gültige ISBN-Nummer (International Standard Book Number) ist.

(https://de.wikipedia.org/wiki/Internationale_Standardbuchnummer#Aufbau_der_ISBN)

Quersumme

Schreiben Sie ein Programm, das eine Methode "berechneQuersumme" implementiert. Die Methode soll die Quersumme einer gegebenen Zahl berechnen und zurückgeben. Die Quersumme einer Zahl ist die Summe der einzelnen Ziffern der Zahl.

Die Methode sollte folgende Signatur haben:

```
public static int berechneQuersumme(int zahl)
```

Die Methode "berechneQuersumme" soll die Quersumme der gegebenen Zahl berechnen und als Ergebnis zurückgeben. Zum Beispiel ist die Quersumme von 1234 gleich $1 + 2 + 3 + 4 = 10$.

Um die Aufgabe zu lösen, können Sie die gegebene Zahl in ihre einzelnen Ziffern aufteilen und diese dann addieren. Eine Möglichkeit, die Ziffern einer Zahl zu extrahieren, ist die Verwendung des Modulo-Operators (%) und der Division (/).

Beachten Sie, dass die Methode auch negative Zahlen akzeptieren und die Quersumme korrekt berechnen sollte.

Spiel: Zahl erraten

Schreiben Sie ein Programm, das eine Zufallszahl zwischen 1 und 100 generiert. Der Spieler soll dann versuchen, diese Zahl zu erraten, indem er Eingaben über die Konsole macht. Das Spiel sollte die folgenden

Methoden enthalten:

- **generateRandomNumber():** Diese Methode generiert eine Zufallszahl zwischen 1 und 100 und gibt sie zurück.
- **getUserGuess():** Diese Methode fordert den Benutzer auf, eine Zahl einzugeben, und liest diese von der Konsole ein. Die eingegebene Zahl wird zurückgegeben.
- **checkGuess(int randomNumber, int userGuess):** Diese Methode vergleicht die vom Benutzer eingegebene Zahl mit der generierten Zufallszahl. Sie gibt eine Nachricht aus, ob die Zahl zu hoch, zu niedrig oder richtig geraten wurde.
- **playGame():** Diese Methode ist der Hauptteil des Spiels. Sie ruft die anderen Methoden in geeigneter Reihenfolge auf, um das Spiel ablaufen zu lassen. Das Spiel sollte so lange laufen, bis der Benutzer die richtige Zahl geraten hat.

Spiel: Hangman

Schreiben Sie ein Programm, das das Spiel "Hangman" umsetzt. Das Spiel sollte die folgenden Methoden enthalten:

- **chooseWord():** Diese Methode wählt ein zufälliges Wort * aus einer vordefinierten Liste von Wörtern und gibt es * zurück.
- **initializeGuess(word):** Diese Methode nimmt das * ausgewählte Wort als Parameter und gibt einen initialen * Zustand des geratenen Wortes zurück. Zum Beispiel könnte der * initiale Zustand "_ _ _ _" sein, wobei jedes * Unterstrichzeichen für einen Buchstaben steht, der noch * nicht erraten wurde.
- **getGuess():** Diese Methode fordert den Benutzer auf, * einen Buchstaben als Tipp einzugeben, und liest diesen von * der Konsole ein. Der eingegebene Buchstabe wird * zurückgegeben.
- **updateGuess(word, guess, guessedWord):** Diese Methode * nimmt das ausgewählte Wort, den geratenen Buchstaben und den * aktuellen Zustand des geratenen Wortes als Parameter. Sie * überprüft, ob der geratene Buchstabe im Wort enthalten ist, * und aktualisiert den Zustand des geratenen Wortes * entsprechend. Die Methode gibt den aktualisierten Zustand * des geratenen Wortes zurück.
- **isGameOver(word, guessedWord):** Diese Methode überprüft, * ob das Spiel vorbei ist. Das Spiel ist vorbei, wenn entweder * das geratene Wort komplett erraten wurde oder der Spieler * keine Versuche mehr übrig hat.
- **playGame():** Diese Methode ist der Hauptteil des Spiels. Sie ruft die anderen Methoden in geeigneter Reihenfolge auf, um das Spiel ablaufen zu lassen. Das Spiel sollte den aktuellen Zustand des geratenen Wortes und die Anzahl der verbleibenden Versuche nach jeder Runde ausgeben. Das Spiel sollte so lange laufen, bis das Spiel vorbei ist.

Spiel Schere-Stein-Papier

Schreiben Sie ein Programm, das das Spiel "Schere, Stein, Papier" gegen den Computer ermöglicht. Das Spiel sollte die folgenden Methoden enthalten:

- **generateComputerChoice():** Diese Methode generiert die Wahl des Computers, entweder "Schere", "Stein" oder "Papier", und gibt sie zurück.
- **getUserChoice():** Diese Methode fordert den Benutzer auf, seine Wahl ("Schere", "Stein" oder "Papier") einzugeben, und liest sie von der Konsole ein. Die eingegebene Wahl wird zurückgegeben.
- **determineWinner(computerChoice, userChoice):** Diese Methode nimmt die Wahl des Computers und die Wahl des Benutzers als Parameter und bestimmt den Gewinner des Spiels. Sie gibt eine Nachricht aus, wer gewonnen hat, oder ob es unentschieden ist.
- **playGame():** Diese Methode ist der Hauptteil des Spiels. Sie ruft die anderen Methoden in geeigneter Reihenfolge auf, um das Spiel ablaufen zu lassen. Das Spiel sollte den Gewinner nach jeder Runde ausgeben. Das Spiel sollte so lange laufen, bis der Benutzer entscheidet, es zu beenden.