

Versuchen Sie immer, Ihren Code zu kommentieren!

#### Aufgabe 1 (Bedingte Anweisung / Verzweigung)

Schreibe ein Programm, das den Benutzer nach seinem Alter fragt. Wenn das Alter größer oder gleich 18 ist, gib „Du bist volljährig“ aus, ansonsten gib „Du bist minderjährig“ aus.

#### Aufgabe 2 (Bedingte Anweisung / Verzweigung)

Schreibe ein Programm, das den Benutzer nach einer Zahl  $n$  fragt. Überprüfe, ob die Zahl gerade ist, und gib entsprechend „Die Zahl ist gerade“ oder „Die Zahl ist ungerade“ aus.

#### Aufgabe 3 (Bedingte Anweisung / Verzweigung)

Schreibe ein Programm, das den Benutzer nach seinem Alter fragt. Basierend auf dem eingegebenen Alter soll das Programm die Altersgruppe bestimmen und ausgeben. Verwende die folgenden Altersgruppen: 0-12 Jahre (Kinder), 13-19 Jahre (Jugendliche), 20-59 Jahre (Erwachsene) und 60+ Jahre (Senioren).

#### Aufgabe 4 (Bedingte Anweisung / Verzweigung)

Schreibe ein Programm, das den Benutzer nach drei Zahlen fragt. Das Programm soll dann die größte Zahl ermitteln und auf der Konsole ausgeben.

#### Aufgabe 5 (Schleife / Iteration)

Schreibe ein Programm, das die Zahlen von 1 bis 10 in einer for-Schleife ausgibt.

#### Aufgabe 6 (Schleife / Iteration)

Schreibe ein Programm, das die Zahlen von 2 bis 20 in einer for-Schleife ausgibt.

#### Aufgabe 7 (Schleife / Iteration)

Schreibe ein Programm, das die Zahlen von 10 bis 1 rückwärts in einer for-Schleife ausgibt.

#### Aufgabe 8 (Schleife / Iteration)

Schreibe ein Programm, das die Quadrate der Zahlen von 1 bis 5 in einer for-Schleife ausgibt.

#### Aufgabe 9 (Schleife / Iteration)

Schreibe ein Programm, das die Summe der Zahlen von 1 bis 100 mit Hilfe einer for-Schleife berechnet und ausgibt.

#### Aufgabe 10 (Schleife / Iteration)

Schreibe ein Programm, das die ungeraden Zahlen von 1 bis 50 in einer for-Schleife ausgibt.

#### Aufgabe 11

Schreibe ein Programm, das überprüft, ob eine gegebene Zahl eine Primzahl ist. Eine Primzahl ist eine Zahl, die nur durch 1 und sich selbst ohne Rest teilbar ist.

### Aufgabe 12

Schreibe ein Programm, das die Multiplikationstabelle für eine bestimmte ganze Zahl anzeigt.

15 X 1 = 15

15 X 2 = 30

...

15x10 = 150

### Aufgabe 13

Schreibe ein Programm, das die Multiplikatorentabelle horizontal von 1 bis n anzeigt.

1x1 = 1, 2x1 = 2, 3x1 = 3, 4x1 = 4, 5x1 = 5, 6x1 = 6, 7x1 = 7, 8x1 = 8

### Aufgabe 14

Schreibe ein Programm, das die ersten n ungeraden natürlichen Zahlen anzeigt und ihre Summe berechnet.

Die ungeraden Zahlen sind :1 3 5 7 9 11 13 15 17 19

Die Summe ist: 100

### Aufgabe 15

Schreibe ein Programm, das ein Muster wie ein rechtwinkliges Dreieck mit einem Sternchen anzeigt.

\*

\*\*

\*\*\*

\*\*\*\*

### Aufgabe 16

Schreibe ein Programm, um ein Muster wie ein rechtwinkliges Dreieck mit einer Zahl anzuzeigen.

1

12

123

1234

### Aufgabe 17

Schreibe ein Programm, das zwei zufällige Zahlen generiert und den Benutzer auffordert, diese zu addieren. Überprüfe die Antwort des Benutzers und gebe eine entsprechende Rückmeldung aus, ob die Antwort korrekt ist oder nicht.

Um eine Zufallszahl im Bereich von 1 bis 100 zu erhalten, können Sie die folgenden Anweisungen verwenden:

```
System.Random random = new System.Random();  
int number = random.Next(1, 101);
```

### Aufgabe 18

Schreibe ein Programm, das eine Währungsumrechnung zwischen festgelegten Währungen durchführt. Der Benutzer gibt einen Betrag in einer Währung ein (z. B. Euro) und wählt eine Zielwährung (z. B. Dollar). Das Programm gibt den umgerechneten Betrag aus, wobei der Wechselkurs fest vorgegeben ist. Bei einer ungültigen Währung wird „Keine gültige Währung“ in der Konsole ausgegeben.

### Aufgabe 19

Schreibe ein Programm, das den BMI (Body Mass Index) für eine Person berechnet. Der Benutzer gibt das Gewicht in Kilogramm und die Größe in Metern ein, und das Programm berechnet und gibt den BMI sowie eine Kategorisierung des Ergebnisses aus (z. B. Untergewicht, Normalgewicht, Übergewicht).

### Aufgabe 20

Schreibe ein Programm, das die ersten 30 Fibonacci-Zahlen berechnet und anzeigt. Die ersten beiden Fibonacci-Zahlen sind 0 und 1. Jede weitere Fibonacci-Zahl ist die Summe der beiden Vorgänger.

Das ergibt die Zahlenfolge: 0, 1, 1, 2, 3, 5, 8, 13, 21, ..., 514229.

### Aufgabe 21

Schreibe ein Programm, um das Quadrat der Zahlen von 1 bis 10 berechnen. Die Ausgabe soll wie folgt aussehen:

Das Quadrat von 1 ist 1.

Das Quadrat von 2 ist 4.

Das Quadrat von 3 ist 9.

...

Das Quadrat von 10 ist 100.

### Aufgabe 22

Schreibe ein Programm für ein einfaches Zahlrätselspiel. Zuerst wählt die Anwendung eine Zufallszahl im Bereich von 1 bis 100 (einschließlich) aus. Der Spieler hat dann die Gelegenheit, so lange eine Zahl einzugeben, bis er die zuvor zufällig ausgewählte Zahl errät.

Wenn der Spieler die richtige Zahl errät, endet das Spiel. Falls die Vermutung des Spielers falsch ist, erhält er eine entsprechende Meldung sowie einen Hinweis, ob die gesuchte Zahl kleiner oder größer ist als sein letzter Versuch. Optional können Sie die Anzahl der Rateversuche als Teil des Ergebnisses ausgeben.

Um eine Zufallszahl im Bereich von 1 bis 100 zu erhalten, können Sie die folgenden Anweisungen verwenden:

```
System.Random zufall = new System.Random();  
int gesuchteZahl = zufall.Next(1, 101);
```

### Aufgabe 23

Schreibe ein Programm, das die Zahlen von 1 bis 100 ausgibt. Dabei sollen folgende Regeln gelten:

- Wenn die Zahl durch 3 teilbar ist, gib "Fizz" aus.
- Wenn die Zahl durch 5 teilbar ist, gib "Buzz" aus.
- Wenn die Zahl durch sowohl 3 als auch 5 teilbar ist, gib "FizzBuzz" aus.
- Ansonsten gib die aktuelle Zahl aus.

### Aufgabe 24

Schreibe ein Programm, das die Gültigkeit einer Kreditkartennummer mithilfe des Luhn Algorithmus überprüft. Der Benutzer sollte die Kreditkartennummer als Eingabe eingeben können, und das Programm sollte ausgeben, ob die Nummer gültig ist oder nicht.

Der Luhn Algorithmus, auch als Modulus 10 oder mod 10 Algorithmus bekannt, wird häufig zur Überprüfung der Gültigkeit von Kreditkartennummern verwendet. Hier ist eine vereinfachte Erläuterung des Algorithmus:

- Durchlaufe die Nummer ziffernweise von rechts nach links und bilde die Summe der Ziffern
- Verdoppele dabei jede zweite Ziffer, und wenn dabei ein Wert größer als 9 herauskommt, subtrahiere 9
- Addiere alle Summen der Ziffern der Kreditkartennummer
- Modulo 10 überprüfen: Überprüfe, ob die Summe durch 10 teilbar ist. Die Kreditkartennummer ist gültig, wenn die Summe durch 10 ohne Rest teilbar ist.

Gegeben sei die Beispielidentifikationsnummer 446667651

Ziffer	Verdoppelt	Reduziert	Summe der Ziffern
1		1	1
5	10	10 - 9	1
6		6	6
7	14	14 - 9	5
6		6	6
6	12	12 - 9	3
6		6	6
4	8	8	8
4		4	4
Gesamtsumme:			40

### Aufgabe 25

Schreibe ein Programm, welches zwei übergebene Zahlen mittels der russischen Multiplikation multipliziert und das Ergebnis in der Konsole ausgibt.

Die russische Multiplikation ist ein Verfahren zur schriftlichen Multiplikation, welches auf der Darstellung der beiden Faktoren zur Basis zwei beruht. Das folgende Beispiel erläutert die

Vorgehensweise. Es soll das Produkt der Zahlen 377 und 53 berechnet werden. Dazu werden die beiden Faktoren in zwei Spalten eingetragen:

Halbieren	Verdoppeln
<b>377</b>	<b>53</b>
188	<del>106</del>
94	<del>212</del>
47	424
23	848
11	1696
5	3392
2	<del>6784</del>
1	13568
Ergebnis:	<b>19981</b>

Die Werte der linken Spalte entstehen durch Ganzzahl-Division durch zwei, die Werte der rechten Spalte durch Verdoppeln der Zahlen. Sind die Werte der linken Spalte gerade, so werden die zugehörigen Werte der rechten Spalte gestrichen. Das Produkt errechnet sich aus der Summe der nicht gestrichenen Zahlen der rechten Spalte, in diesem Beispiel aus der folgenden Summe:

$$53 + 424 + 848 + 1696 + 3392 + 13568 = 19981$$

Weiteres Beispiel:

Halbieren	Verdoppeln
<b>22</b>	<b>3</b>
11	6
5	12
2	<del>24</del>
1	48
Ergebnis:	<b>66</b>

$$\text{Ergebnis: } 22 * 3 = 6 + 12 + 48 = 66$$

## Aufgabe 26

Schreibe eine Taschenrechner Konsolenanwendung, die folgende Funktionalitäten bietet:

- Der Benutzer gibt die erste Zahl ein.
- Der Benutzer gibt ein, welche mathematische Operation durchgeführt werden soll (+, -, \*, /).
- Der Benutzer gibt die zweite Zahl ein.
- Das Programm führt die Berechnung durch und gibt das Ergebnis aus.
- Nach der Ausgabe des Ergebnisses wird der Benutzer gefragt, ob er eine weitere Berechnung durchführen möchte.

- Falls ja, kann der Benutzer optional entscheiden, das vorherige Ergebnis als erste Zahl zu verwenden und eine neue Operation und zweite Zahl eingeben.
- Falls nein, endet das Programm.