

08 Schleifen: For

1. Zahlenreihe ausgeben: Schreiben Sie eine for-Schleife, die die Zahlen von 1 bis 10 ausgibt.
2. Quadratzahlen berechnen: Nutzen Sie eine for-Schleife, um die Quadratzahlen der Zahlen von 1 bis 5 zu berechnen und auszugeben.
3. Rückwärtszählen: Schreiben Sie eine for-Schleife, die Zahlen von 10 bis 1 rückwärts ausgibt.
4. Schreiben einer For-Schleife, die jede Zahl der 2-er Reihe ausgibt. (2,4,6,8,...).
 - Erweitern Sie dieses Programm, sodass die Schritte (2) als Programmargument übergeben wird. (args auslesen, in int parsen)
5. Schreiben Sie eine for-Schleife, die die Potenz einer Zahl berechnet. Der Benutzer gibt die Basiszahl und den Exponenten ein, und die Schleife berechnet die Potenz (zum Beispiel $2^5 = 32$)
6. Harmonische Reihe: Implementieren Sie ein Programm, das die Summe der harmonischen Reihe bis zu einer vom Benutzer festgelegten Anzahl von Termen berechnet. Die harmonische Reihe ist die unendliche Reihe $1 + 1/2 + 1/3 + 1/4 + \dots$ usw. Verwenden Sie eine for-Schleife, um die Summe der harmonischen Reihe bis zu einer bestimmten Anzahl von Termen zu berechnen, die vom Benutzer eingegeben wird (zB. Eingabe 20 = $1/20$). Geben Sie das Ergebnis aus.

Programme

Einnahmen erfassen

Das Programm soll die Kosten für die Bereiche **Nahrungsmittel**, **Getränke** erfassen.

1. Fragen Sie den Benutzer nach seinen Namen und begrüßen Sie ihn/sie.
2. Fragen Sie den Benutzer in einer Schleife solange um die Eingaben, bis **abbruch** eingegeben wird. Anschließend sollen alle Summen angezeigt werden.
3. Schreiben Sie hierzu ein Menü mit der Abfrage der Aktion mit folgenden Optionen: **nahrung**, **getränk**, **abbruch**
4. Legen Sie zwei double Variablen für Nahrung und Getränke an, in deren die eingegebenen Float-Werte summiert werden.
5. Fragen Sie den Benutzer um den Betrag
6. Implementieren Sie folgende Logik
 - **nahrung**
 - Variable für Nahrung wird um den eingegebenen Betrag erhöht
 - **getränk**
 - Variable für Getränk wird um den eingegebenen Betrag erhöht
 - **abbruch**
 - Ausgabe der beiden summen mit zwei Nachkommastellen.

Beispiel

```
Willkommen bei der Einnahmenerfassung!
Wie ist Ihr Name?
Max
```

```
Hallo Max!
Bitte wählen Sie eine Aktion (`nahrung`, `getränk`, `abbruch`):
`getränk`
Geben Sie eine Summe für die Kategorie getränk ein: 12.00

Bitte wählen Sie eine Aktion (`nahrung`, `getränk`, `abbruch`):
`getränk`
Geben Sie eine Summe für die Kategorie getränk ein: 10.00

Bitte wählen Sie eine Aktion (`nahrung`, `getränk`, `abbruch`):
`nahrung`
Geben Sie eine Summe für die Kategorie getränk ein: 18.00

Bitte wählen Sie eine Aktion (`nahrung`, `getränk`, `abbruch`):
`abbruch`
Abrechnung für Max: 22,00€ für Getränke und 18,00€ für Nahrungsmittel.
```

Pakete einlagern

Erstellen Sie einen String mit dem Inhalt \$\$\$\$\$. Es sollen nun nach Benutzereingaben Pakete eingelagert und entnommen werden können. Der String hat 5 leere Plätze mit den Indizes 0,1,2,3,4. (\$ bedeutet, dass der Platz leer ist.)

1. Legen Sie den String `storage` mit \$\$\$\$ an.
2. Fragen Sie den Benutzer, welche Aktion er ausführen möchte. Geben Sie hierzu folgende Optionen:
 - `einlagern`
 - `auslagern`
 - `beenden`
3. Nach der Wahl der Option, soll eine Paketnummer (0-9) abgefragt werden.
4. Überschreiben des nächsten freien Platzes mit der Paketnummer anhand folgender Logik:
 - `einlagern`
 - das erste \$ wird durch die Paketnummer ersetzt
 - Gibt es keinen freien Platz mehr, so wird eine Meldung ausgegeben.
 - `auslagern`
 - die Paketnummer wird durch \$ ersetzt.
 - `beenden`
 - beendet das Programm.
5. Geben Sie in jedem Schleifendurchlauf die Variable `storage` aus.

Beispiel

```
Willkommen im Speicher
Wählen Sie eine Aktion (einlagern, auslagern, beenden): einlagern
Geben Sie die Paketnummer ein: 2
2$$$$

Wählen Sie eine Aktion (einlagern, auslagern, beenden): einlagern
Geben Sie die Paketnummer ein: 6
```

26\$\$\$

Wählen Sie eine **Aktion** (einlagern, auslagern, beenden): einlagern
Geben Sie die Paketnummer ein: 9
269\$\$

Wählen Sie eine **Aktion** (einlagern, auslagern, beenden): auslagern
Geben Sie die Paketnummer ein: 2
\$69\$\$

Wählen Sie eine **Aktion** (einlagern, auslagern, beenden): einlagern
Geben Sie die Paketnummer ein: 8
869\$\$

Passwort Generator

Schreiben Sie ein Programm, zufällige Passwörter generiert.

1. Fragen Sie hierzu den Benutzer folgende Fragen:
 - Soll das Passwort Großbuchstaben beinhalten?
 - Soll das Passwort Ziffern beinhalten?
 - Soll das Passwort Sonderzeichen beinhalten?
 - Wie lang soll das Passwort sein?
2. Fragen Sie den Benutzer wie viele Passwörter generiert werden sollen
3. Überlegen Sie sich einen passenden Algorithmus um die eingegebene Anzahl an Passwörter zu generieren.
4. Geben Sie die Passwörter aus.

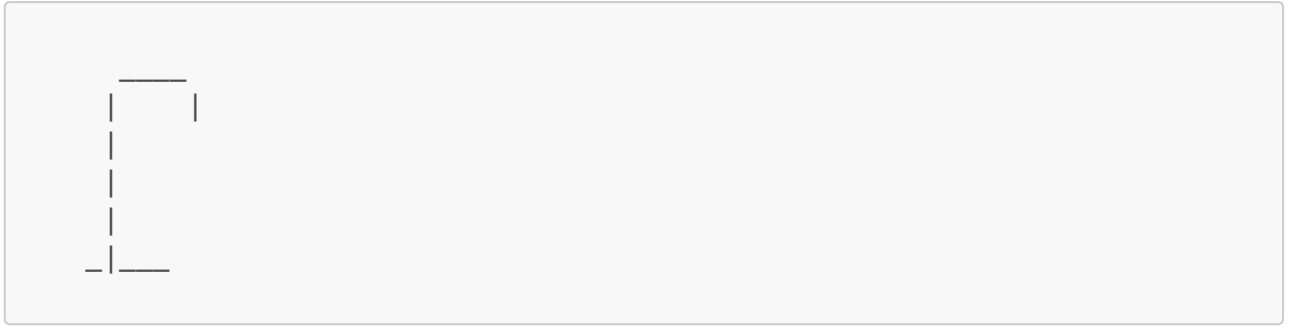
Hangman

Entwickeln Sie das Spiel "Hangman" in Java. Das Spiel sollte folgende Funktionen haben:

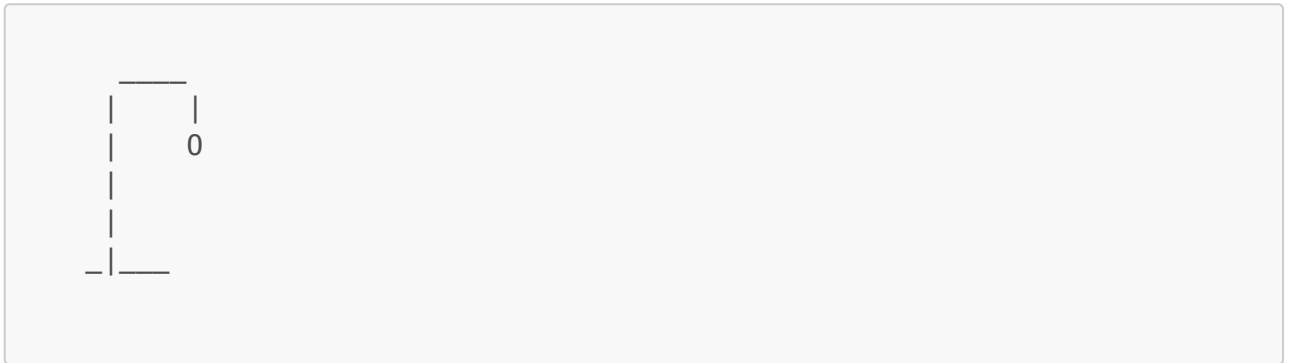
1. Fragen Sie den Benutzer nach einem Wort mit 3 Buchstaben. Die Zahl 3 soll in der Konstante Länge gespeichert werden und somit einfach veränderbar sein.
2. Ist das Wort zu lang oder leer, soll eine Fehlermeldung ausgegeben werden und erneut angefragt werden.
3. Anzeige des Status des Wortes, das der Spieler erraten muss (zum Beispiel als Striche (_), die die Buchstaben repräsentieren).
4. Eine Schleife, die dem Spieler erlaubt, Buchstaben zu raten, bis das Wort vollständig erraten oder der Galgenmann "vollständig gehängt" ist. (Groß-/Kleinschreibung soll egal sein)
5. Begrenzung der Anzahl der Fehlversuche des Spielers (zum Beispiel sechs Fehlversuche, bevor das Spiel endet).
6. Überprüfung der Eingaben des Spielers und Aktualisierung des Spielstands entsprechend der geratenen Buchstaben.
7. Anzeige des Gewinns oder Verlusts des Spiels nach dem Ende der Spielrunde.

Nach jeder Runde soll ein Galgenmann angezeigt werden:

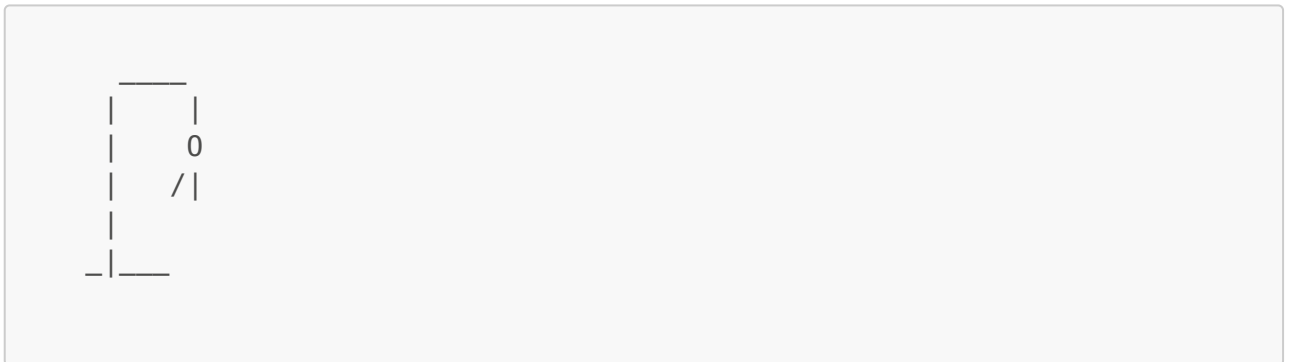
- 1 Fehler:



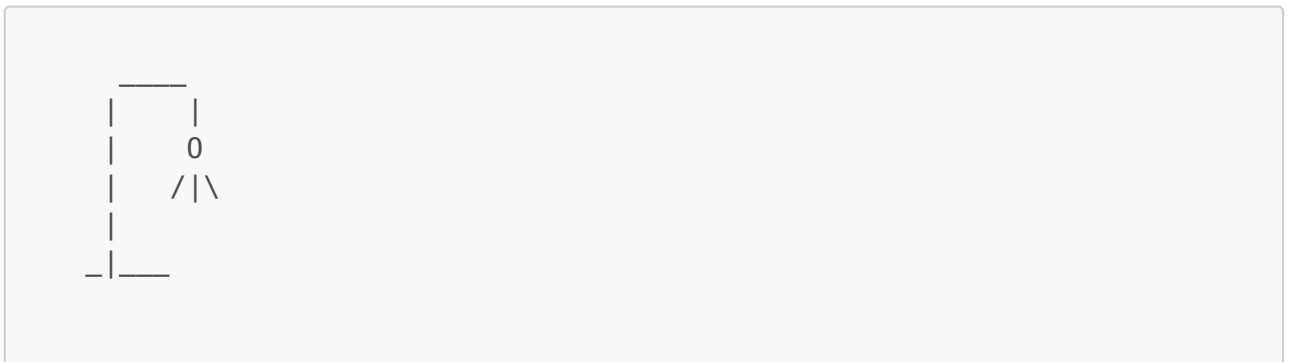
- 2 Fehler:



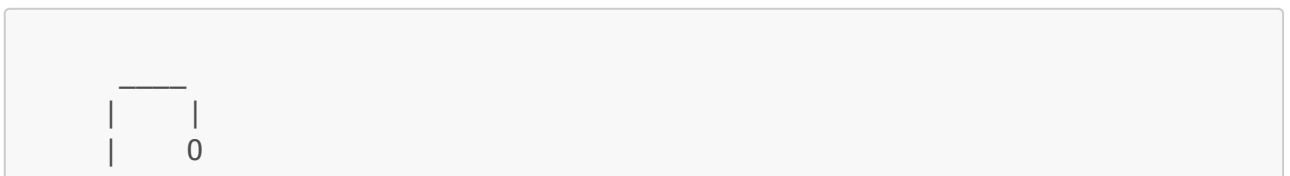
- 3 Fehler:

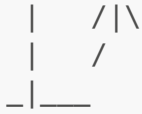


- 4 Fehler:



- 5 Fehler:





- 6 Fehler:



- Beispielwörter mit 3 Buchstaben: **B**au, **H**ut, **A**rm, **T**ag, **E**is, **Z**oo, **S**ee
- Beispielwörter mit 4 Buchstaben: **H**aus, **B**ahn, **T**ier, **B**uch, **F**est