
1 String-Addition

1.1 Aufgabe

Implementiere eine Funktion `greetPlayerAskChoiceHelper`, die einen Namen als String übergeben bekommt und einen String zurückgibt, in dem die Person gebeten wird, sich für Schere, Stein oder Papier zu entscheiden.

```
159 greetPlayerAskChoiceHelper("Anna")
```

Hey Anna! Please choose rock, paper or scissors!

```
160 greetPlayerAskChoiceHelper("Timo")
```

Hey Timo! Please choose rock, paper or scissors!

2 String-Addition und Typkonversion

2.1 Aufgabe

Implementiere eine Funktion `showRoundNumber`, die zwei Integer als Argumente übergeben bekommt. Dabei handelt es sich um die Runde die gerade gespielt wird und die Anzahl der Runden, die insgesamt gespielt werden. Die Funktion gibt einen String zurück, in dem beide Informationen enthalten sind.

```
161 showRoundNumber(1, 3)
```

Round 1 of 3

```
162 showRoundNumber(4, 5)
```

Round 4 of 5

2.2 Aufgabe

Implementiere eine Funktion `greetPlayerAskNameHelper`. Dieser Funktion wird die Nummer eines Spielers als Integer übergeben. Sie gibt einen String zurück, indem der Spieler begrüßt und nach seinem Namen gefragt wird.

```
163 greetPlayerAskNameHelper(1)
```

Hello Player 1! What's your name?

```
164 greetPlayerAskNameHelper(2)
```

Hello Player 2! What's your name?

3 Ausgabe

3.1 Aufgabe

Implementiere eine Funktion `printRoundNumber`. Dieser Funktion werden die aktuelle Rundenzahl und die Anzahl der Runden die insgesamt gespielt werden als Integer übergeben. Sie gibt beide Informationen an der Konsole aus.

```
165 printRoundNumber(1, 5)
```

Round 1 of 5

```
166 printRoundNumber(1, 1)
```

Round 1 of 1

Nutze `showRoundNumber`!

4 Bedingungen

4.1 Aufgabe

Implementiere eine Funktion `showPlayerPoints`. Dieser Funktion wird ein Name als String und eine Punktezahl als Integer übergeben. Sie gibt einen String zurück, in dem steht wie viele Punkte die Person hat.

```
167 showPlayerPoints("Kevin", 1)
```

Kevin has 1 point

```
168 showPlayerPoints("Barbara", 3)
```

Barbara has 3 points

4.2 Aufgabe

Implementiere eine Funktion `computeRoundWinner`. Dieser werden die Eingaben der beiden Spieler als Strings übergeben. Sie gibt zurück, wer die Runde gewonnen hat. Die einzigen möglichen Argumente sind `"rock"`, `"paper"` und `"scissors"`. Die einzigen möglichen Rückgabewerte sind `"1"`, `"2"` und `"Nobody"`.

```
169 computeRoundWinner("rock", "scissors")
```

1

```
170 computeRoundWinner("rock", "paper")
```

2

```
171 computeRoundWinner("rock", "rock")
```

Nobody

4.3 Aufgabe

Implementiere eine Funktion `showWinnerWithNameHelper`. Dieser wird übergeben, wer das Spiel/die Runde gewonnen hat. Die einzigen Möglichkeiten sind `"1"`, `"2"` und `"Nobody"`. Außerdem werden die Namen der Spieler und ein Boolean übergeben. Wenn dieses `true` ist soll als String zurückgegeben werden welcher Spieler, das Spiel gewonnen hat. Ansonsten wird zurückgegeben welcher Spieler die Runde gewonnen hat. In dem String werden die übergebenen Namen verwendet.

```
172 showWinnerWithNameHelper("1", "Grace", "Alan", true)
```

Grace wins this game!

```
173 showWinnerWithNameHelper("2", "Grace", "Alan", false)
```

Alan wins this round!

```
174 showWinnerWithNameHelper("Nobody", "Grace", "Alan", false)
```

Nobody wins this round!

4.4 Aufgabe

eine Funktion `showRoundWinnerWithName`. Dieser wird übergeben wer das Spiel gewonnen hat. Die einzigen Möglichkeiten sind `"1"`, `"2"` und `"Nobody"`. Außerdem werden die Namen der Spieler übergeben. Die Funktion gibt einen String zurück in dem steht, wer die Runde gewonnen hat. In dem String wird der Name des Spielers verwendet.

```
175 showRoundWinnerWithName("1", "Grace", "Alan")
```

Grace wins this round!

```
176 showRoundWinnerWithName("2", "Grace", "Alan")
```

Alan wins this round!

```
177 showRoundWinnerWithName("Nobody", "Grace", "Alan")
```

Nobody wins this round!

Nutze `showWinnerWithNameHelper`!

4.5 Aufgabe

eine Funktion `showGameWinnerWithName`. Dieser werden Namen der Spieler als Strings und ihre die Punktezahlen als Integer übergeben. Die Funktion gibt einen String zurück in dem steht, wer das Spiel gewonnen hat.

```
178 showGameWinnerWithName("Grace", "Alan", 1, 2)
```

Alan wins this game!

```
179 showGameWinnerWithName("Grace", "Alan", 5, 0)
```

Grace wins this game!

```
180 showGameWinnerWithName("Grace", "Alan", 1, 1)
```

Nobody wins this game!

Nutze `showWinnerWithNameHelper`!

4.6 Aufgabe

Implementiere eine Funktion `computeNewPoints`. Dieser wird der Gewinner der letzten Runde, die Nummer eines Spielers und die Punktzahl des Spielers mit dieser Nummer übergeben. Die einzigen Möglichkeiten für das erste Argument sind `"1"`, `"2"` und `"Nobody"`. Die Funktion gibt die neue Punktzahl des Spielers zurück.

```
181 computeNewPoints("1", 1, 3)
```

4

```
182 computeNewPoints("2", 1, 3)
```

3

```
183 computeNewPoints("Nobody", 1, 1)
```

1

5 Ausgabe

5.1 Aufgabe

Schreibe eine Funktion `printRoundWinnerWithName`. Diese funktioniert genau wie `showRoundWinnerWithName`. Sie gibt den Strings aber an der Konsole aus und nicht zurück.

```
184 printRoundWinnerWithName("1", "Grace", "Alan")
```

Grace wins this round!

```
185 printRoundWinnerWithName("2", "Grace", "Alan")
```

Alan wins this round!

```
186 printRoundWinnerWithName("Nobody", "Grace", "Alan")
```

Nobody wins this round!

Nutze `showRoundWinnerWithName`!

5.2 Aufgabe

Implementiere eine Funktion `printPlayersPoints`. Dieser Funktion werden zwei Spielernamen als Strings und deren Punktezahl als Integer übergeben. Sie gibt an der Konsole aus welcher Spieler wie viele Punkte hat.

```
187 printPlayersPoints("Kevin", "Barbara", 1, 5)
```

```
Kevin has 1 point
Barbara has 5 points
```

```
188 printPlayersPoints("Ali", "Maria", 1, 5)
```

```
Ali has 1 point
Maria has 5 points
```

Nutze `showPlayersPoints`!

5.3 Aufgabe

Implementiere eine Funktion `printGameWinnerWithName`. Dieser Funktion werden zwei Spielernamen als Strings und deren Punktezahl als Integer übergeben. Sie gibt das Endergebnis an der Konsole aus.

```
189 printGameWinnerWithName("Kevin", "Barbara", 1, 5)
```

```
Barbara wins this game!
```

```
190 printGameWinnerWithName("Ali", "Maria", 6, 5)
```

```
Ali wins this game!
```

```
191 printGameWinnerWithName("Ali", "Maria", 6, 6)
```

```
Nobody wins this game!
```

Nutze `showGameWinnerWithName`!

6 Eingabe

6.1 Aufgabe

Implementiere eine Funktion `greetPlayerAskName`. Der Funktion wird eine Spielernummer als Integer übergeben. Sie fordert den Spieler dazu auf seinen Namen an der Konsole einzugeben. Der Name wird anschließend zurückgegeben.

Nutze `greetPlayerAskNameHelper`!

6.2 Aufgabe

Schreibe eine Funktion `ask_rounds`, die fragt wie viele Runden gespielt werden sollen. Diese Eingabe wird als Integer zurückgegeben.

6.3 Aufgabe

Implementiere eine Funktion `greetPlayerAskChoice`. Der Funktion wird eine Spielername als String übergeben. Sie fordert die Spielerin dazu auf Schere, Stein oder Papier einzugeben. Die Auswahl wird als String zurückgegeben.

Nutze `greetPlayerAskChoiceHelper`

6.4 Aufgabe

Implementiere eine Funktion `playOneRound`. Dieser Funktion werden die aktuelle Rundenzahl und die Zahl der Runden, die gespielt werden, übergeben. Sie gibt diese beiden Informationen in der Konsole aus. Der Funktion werden auch die Namen der Spieler übergeben. Sie spricht die Spieler mit ihren Namen an und bittet sie Schere, Stein oder Papier einzugeben. Das Ergebnis dieser Runde wird anschließend aus- und gegeben. Hierbei wird der Name des Gewinners genutzt.

Es wird auch zurückgegeben, wer gewonnen hat. Die einzig möglichen Rückgabewerte sind `"1"`, `"2"` und `"Nobody"`.

Nutze `printRoundNumber`, `greetPlayerAskChoice`, `computeRoundWinner` und `printRoundWinnerWithName`!

6.5 Aufgabe

Implementiere eine Funktion `playRPSHelper`. Diese Funktion werden die Anzahl der Runden, die gespielt werden und die Namen der Spieler übergeben. Sie spielt das Spiel unter diesen Bedingungen. Nachdem alle Runden gespielt wurden, wird das Endergebnis ausgegeben.

Beispiel:

```
Round 1 of 3
Hey Alan! Please choose rock, paper or scissors!
...
...
```

Nutze `playOneRound`, `computeNewPoints` und `printPlayersPoints`!

6.6 Aufgabe

Implementiere eine Funktion `playRPS`. Diese Funktion fragt zwei Spieler nach ihren Namen. Anschließend fragt sie wie viele Runden gespielt werden sollen. Danach wird das Spiel mit diesen Einstellungen durchgeführt.

Beispiel:

Hello Player 1! What's your name?

Alan

Hello Player 2! What's your name?

Grace

How many rounds do you want to play?

3

Round 1 of 3

Hey Alan! Please choose rock, paper or scissors!

Nutze greetPlayerAskName, aksRounds und playRPSHelper!

6.7 Aufgabe

Nutze das Kontrollzeichen `"\n"` im Code um bei der Ausgabe Zeilen zu überspringen.