

## Prozedurale Programmierung

### Übungsaufgabe u04a

enum und switch, Zufallszahlengenerator, Funktionen

Aufzählungskonstanten (enum) dienen dazu, den Quellcode lesbarer zu gestalten. Sie ordnen einem ganzzahligen Wert einen Text zu, der an Stelle der Zahl in Zuweisungen oder Ausdrücken verwendet werden kann.

Beispiel:

```
enum Wochentage {Mo=1, Di, Mi, Do, Fr, Sa, So};  
int Feiertag = Do; // Feiertag hat den Wert 4
```

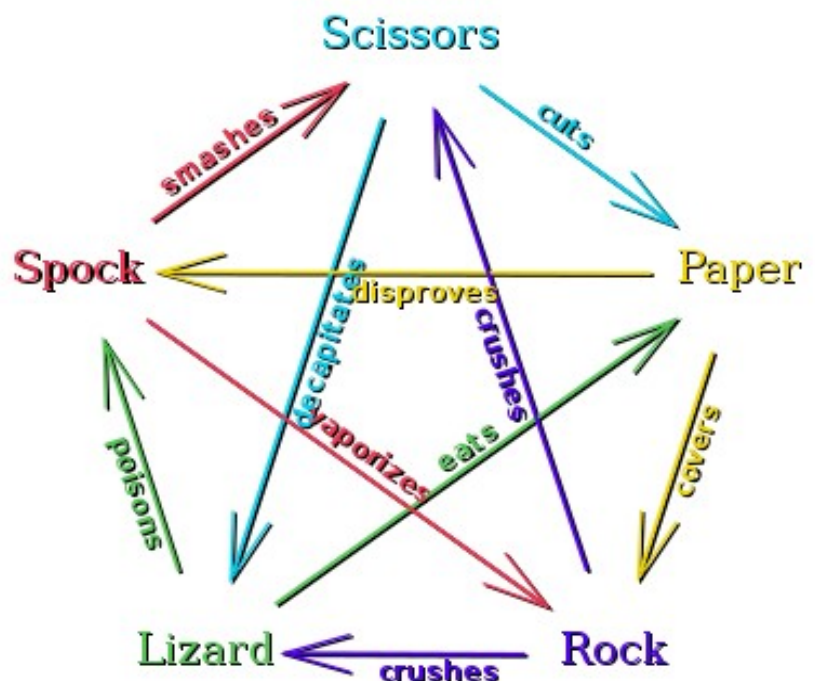
Wenn einer Konstanten kein Wert zugewiesen wird, so wird der vorherige Wert um 1 erhöht. `Di` hat also den Wert 2, `Mi` den Wert 3 usw. Wird der ersten Konstanten kein Wert zugewiesen, so erhält Sie den Wert 0.

Sehen Sie sich das entsprechende Kapitel in Ihrem Lehrbuch an.

Schreiben Sie ein Programm mit mehreren Funktionen (s.u.), welches für zwei Spieler das Spiel **Stein, Schere, Papier, Echse, Spock** spielt

(<http://youtu.be/5cujdrWbc9k>):

- *Schere schneidet Papier*
- *Papier bedeckt Stein*
- *Stein zerquetscht Echse*
- *Echse vergiftet Spock*
- *Spock zertrümmert Schere*
- *Schere köpft Echse*
- *Echse frisst Papier*
- *Papier wiederlegt Spock*
- *Spock verdampft Stein*
- *Stein schleift Schere*



(Quelle: <http://the-big-bang-theory-forum.rockz.com/64818-Schere-Stein-Papier-Echse-Spock>)

- Definieren Sie eine Aufzählungskonstante „symbol“, die den Symbolen stein, schere, papier, echse, spock die Konstanten 0 bis 4 zuordnet.
- Zunächst wird der Zufallszahlengenerator durch einmaligen Aufruf der Funktion `srand(time(NULL))` gestartet. Suchen Sie die notwendigen `include`-Anweisungen für `srand()` und `time()` heraus.

Erstellen Sie eine **Funktion „spiel()“**:

- Definieren Sie Variablen vom Typ „symbol“ für die beiden Spieler und weisen Sie ihnen je eine Zufallszahl zwischen 0 und 4 zu (Funktion `rand()` und Modulo-Operator verwenden). Diese Werte entsprechen einem Symbol im Spiel.
- Geben Sie die Werte für die Spieler in folgender Form aus:  
Spieler1: Spock (wenn die Zufallszahl von Spieler1 die 4 war)  
Spieler2: Echse (wenn die Zufallszahl von Spieler2 die 3 war)
- Geben Sie das zugehörige Symbol als Text aus. Verwenden Sie eine `switch-case`-Anweisung. Verwenden Sie die Aufzählungskonstanten für die `case`-Ausdrücke.  
Bsp.: Wert ist Echse (3) → Ausgabe: „Echse“
- Überlegen Sie, welchen Wert die Funktion zurückgeben soll

Erstellen Sie eine **Funktion „entscheid()“**:

- Entscheiden Sie, welcher Spieler gewonnen hat. Erstellen Sie dazu eine `switch-case`-Anweisung für Spieler1.
- Verwenden Sie die Aufzählungskonstanten für die `case`-Ausdrücke. In den einzelnen `case`-Abschnitten überprüfen Sie mit einem `if-else`-Ausdruck, wer gewonnen hat und geben den Gewinner aus.
- Überlegen Sie, welchen Wert die Funktion zurückgeben soll

Beispiel:

- Spieler1 zeigt Schere: also überprüfe
  - ob Spieler2 Papier oder Echse zeigt
    - dann gewinnt Spieler1
  - sonst
    - gewinnt Spieler2
- Überlegen Sie, wo die richtige Stelle für den Aufruf der Funktion „entscheid()“ ist.

- Ergänzen Sie eine Behandlung für den Gleichstand in einem Spiel. In diesem Fall wird eine weitere Runde gespielt.
- Das Programm soll sich beenden, sobald Spieler1 fünf mal gewonnen hat.

Erweiterung (knifflig!):

Erstellen Sie eine **Funktion „zaehlmit()“**:

- Die Funktion soll Statistik-Daten sammeln:
- mitzählen, wie oft jeder der Werte **Stein, Schere, Papier, Echse, Spock** per Zufallszahl gewählt wurde
- mitzählen, wie oft welcher Spieler (1 oder 2) gewonnen hat
- Tipp: verwenden Sie „static“ Variablen in der Funktion

Erstellen Sie eine **Funktion „auswert()“**:

- Statistik Daten anzeigen:
- Zählergebnisse für die 5 Werte **Stein, Schere, Papier, Echse, Spock** ausgeben
- Anzahlen Gewinne der beiden Spieler ausgeben
- Überlegen Sie sich, wie Sie an die Werte der „static“ Variablen kommen; Tipp: Die Funktion „auswert()“ könnte die Funktion „zaehlmit()“ mit einem besonderen Parameter aufrufen, der veranlasst, dass „zaehlmit()“ die Statistikdaten ausgibt.

Das Programm muss fehlerfrei und ohne Warnungen mit der Compileroption `-Wall` compilieren. Kommentieren Sie den Quellcode hinreichend und tragen Sie Ihren Namen in den Kopfkomentar ein.

**Bewertungsschema:**

**Übersetzungsfehler ==> 0 P !**

sonst:

Codeformatierung	2 P
sinnvolle Kommentare	2 P
Funktion spiele()	4 P
Funktion entscheide()	4 P
Funktion zaehlmit()	2 P
Funktion azuswert()	2 P
Hauptprogramm, Ein-/Ausgaben, Aufrufe	4 P
	-----
Summe	20 P