

Die Hausaufgaben müssen von den Studierenden einzeln bearbeitet und abgegeben werden. Für die Hausaufgabe sind die aktuellen Informationen vom Blog <https://seblog.cs.uni-kassel.de/ws2223/einfuehrung-in-die-informatik/> zu berücksichtigen.

**Abgabefrist ist der 09.02.2023 - 23:59 Uhr**

## Vorbereitung

Zur Abgabe dieser Hausaufgabe muss zunächst ein neues Respository angelegt werden. Mit dem folgenden Link kannst du das Respository für Hausaufgabe 12 anlegen bzw. einsehen.

[https://classroom.github.com/a/s71N\\_0qx](https://classroom.github.com/a/s71N_0qx)

Achte bei jeder Aufgabe auf das verlangte Dateiformat. Andere Formate werden nicht akzeptiert und folglich mit **0** Punkten gewertet. Handschriftliche Abgaben werden nicht akzeptiert.

Es sollen nur die bis zur jeweiligen Übung vermittelten Konzepte verwendet werden.

**Nicht oder zu spät abgegebene (Teil-)Aufgaben werden mit 0 Punkten bewertet.**

## Code-Formatierung

Formatiere deinen Code vor dem Abgeben wie in Vorlesung und Übung gezeigt. Verstöße gegen Formatierungs- und Namenskonventionen führen zu Punktabzug.

## Lauffähigkeit

**Nicht ausführbare Abgaben werden mit 0 Punkten bewertet.**

## 3-Credit Abschnitt

Diese Hausaufgabe zählt zu dem in Übung 1 erläuterten „3-Credit Abschnitt“ der Veranstaltung. Wenn du dir unsicher bist, welche Prüfung du in dieser Veranstaltung ablegen musst, informiere dich bitte bei deinem Studienservice.

## Aufgabe 1 - Snake (21P)

Lege ein neues Java-Projekt an und binde die benötigte core.jar-Datei ein, um die Processing-Bibliothek verwenden zu können.

In dieser Hausaufgabe soll das Spiel „Snake“ programmiert werden.

### Spielprinzip

Auf dem Spielfeld bewegt sich eine Schlange, deren Richtung der User mit den Pfeiltasten steuern kann. Es erscheint zufällig irgendwo auf dem Spielfeld Futter. Schafft man es, die Schlange so steuern, dass sie einen Futterblock fressen kann, verschwindet er und dafür wird die Schlange um eine Einheit länger. Für jedes gefressene Futter bekommt der User einen Punkt.

Wird die Schlange so gesteuert, dass sie sich selber beißt, ist das Spiel verloren. Kollidiert die Schlange mit dem Rand des Spielfelds ist das Spiel ebenfalls verloren.

Die Schlange soll nicht in die genau entgegengesetzte Richtung zu ihrer aktuellen Bewegungsrichtung gesteuert werden können. D. h. wenn sie sich beispielsweise nach oben bewegt, hat es keinen Effekt, wenn der User den Pfeil nach unten drückt.

### Vorgaben

Das Programmfenster soll 720 x 720 Pixel groß sein. Beim Game-Over soll wie üblich die Punktzahl angezeigt werden und es soll möglich sein, das Spiel erneut zu starten. Farben, Geschwindigkeiten und Größen dürfen selbst festgelegt werden. Achte darauf, dass alle Objekte und Texte erkennbar sind und das Spiel genauso gut spielbar ist wie ein übliches Snake-Spiel.

### Programm-Struktur

Die Strukturierung deines Programms darfst du frei wählen! Als vorgegebene Anforderung wird eine Klasse namens `Pair` verlangt. Diese **generische** Klasse soll zwei **private** Variablen enthalten. Die Variable `x` bekommt als Typ den Platzhalter `T`. Die Variable `y` bekommt als Typ den Platzhalter `U`. Hilfsmethoden sind erlaubt.

`Pairs` sollen dann an anderen Stellen deines Programms verwendet werden, um

- die Koordinaten von Futterblöcken zu verwalten
- die Koordinaten der einzelnen Teile des Schlangenkörpers zu verwalten

Alles Weitere darfst du frei mit allen Mitteln gestalten, die in den Vorlesungen und Übungen behandelt wurden.

### Abgabe

Gib deine Dateien mit der Lösung im GitHub-Repository zur aktuellen Hausaufgabe ab.