

2.1 🖥 Essen

- Erstelle eine neue Klasse `Essen`.
- Ergänze die Klasse `Essen` um ein *Attribut* `symbol` vom Typ `Rechteck/`
- Implementiere den *Konstruktor* der Klasse `Essen`:
 - Es soll die Position übergeben werden.
 - Initialisiere `symbol` mit einem neuen roten Rechteck an der richtigen Position mit der Größe 20x20.
- Implementiere eine Methode `entferne()` der Klasse `Essen`, die das `symbol` mittels dessen Methode `Entfernen()` von der Zeichenfläche löscht.
- Teste deine Klasse `Essen`, indem du neue Objekte von ihr anlegst und wieder entfernst.

2.2 🖥 Zeitgesteuertes Essen

Zeitgesteuerte Ereignisse

Die Methode `public void Ausführen()` unserer Spiel-Klasse wird für jedes neue Bild von der Spiel-Engine aufgerufen.

Um Aktionen in Abhängigkeit von der Zeit auszuführen, müssen wir wissen wie viel „Zeit“ bereits vergangen ist.

- Ergänze die Klasse `Snake` um eine *Attribut* `ticks`, das die Anzahl der bereits gezeichneten Bilder speichert.
- Importiere in der Klasse `Snake` die Java Klasse `Random`, indem du außerhalb deiner Klasse `Snake` die Zeile `import java.util.Random;` einfügst.
- Ergänze die Klasse `Snake` um ein *Attribut* `zufall` vom Typ `Random`, initialisiere das *Attribut* im *Konstruktor*.
- Ergänze die Methode `Snake.Ausführen`:
 - Erhöhe zu Beginn die Anzahl der bereits gezeichneten Bilder.
 - Falls das aktuelle Bild ein Vielfaches von 30 ist:
 - * Falls ein `Essen` in dem *Attribut* `essen` der Klasse `Snake` gespeichert ist, entferne es.
 - * Erzeuge ein neues `Essen`-Objekt an einer zufälligen Position (`zufall.nextInt(max)` liefert dir eine Zufallszahl zwischen 0 und `max`).

Modulo

Die **Modulo-Operation**, in Java das Zeichen `%`, liefert als Ergebnis, den Rest der Ganzzahldivision zweier Zahlen.

Beispiele: `7 % 3 == 1`; `64 % 30 == 4`

2.3 Wachstum

Damit unsere Schlange wachsen kann, muss sie aus mehr Körperteilen als nur dem Kopf bestehen.

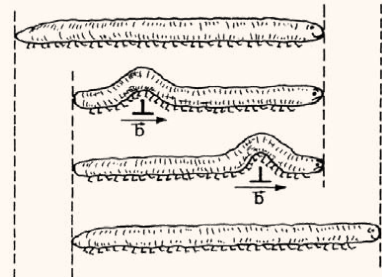
- Ersetze das *Attribut* Kopf der Schlange durch ein Feld namens `koerper` vom Typ `Rechteck`.
- Initialisiere das Feld im *Konstruktor* der Klasse `Schlange` mit Länge 0.
- Ergänze die Klasse `Schlange` um die Methode `wachsen(int x, int y)`:
 - Erzeuge ein neues Feld, das um ein Element länger ist als der aktuelle Körper der Schlange.
 - Erzeuge einen neuen Kopf an der richtigen Position (grün und 20x20 Pixel groß).
 - Speichere den neuen Kopf als erstes Element in dem neu erzeugten Feld.
 - Kopiere alle Elemente des aktuellen Körpers in das neue Feld.
Vergiss nicht: Das neue Feld enthält bereits den Kopf am Index 0.
 - Speichere das neue Feld in dem *Attribut* `koerper`.

2.4 Bewegung der Schlange

Bewegung mehrere Körperteile

Es bietet sich an bei der Bewegung der Schlange die Bewegung der Raupen in der Natur nachzuahmen. Dabei wird jedes Körperteil an die Position des Vorgängers gezogen. Du kannst ein Feld rückwärts durchlaufen, indem du die dir bereits bekannte `for` Schleife leicht modifizierst.

```
for (int i = feld.length - 1; i >= 0; i--) {  
    ...  
}
```



- Beginnend mit dem letzten Körperteil setze die Position auf die des Körperteils davor.
- Verschiebe anschließend den Kopf wie gewohnt in die aktuelle Richtung.
- Teste deine Methoden `bewegen` und `wachsen`, indem du alle 40 Bilder die Schlange in die aktuelle Richtung wachsen lässt.