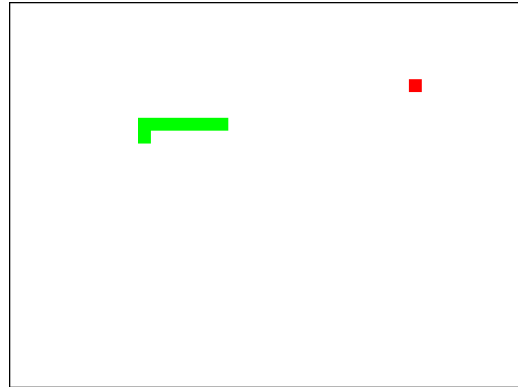
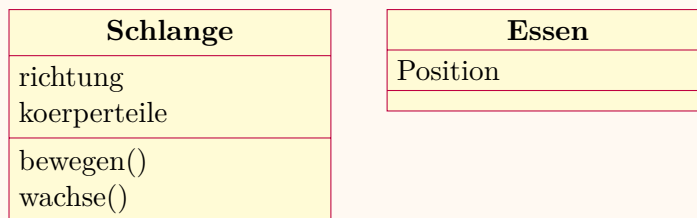


1.1 Modellierung

- Erstelle ein Klassendiagramm mit allen **Klassen**, die du im linken Screenshot ausmachen kannst.
- Notiere die **Attribute**, die du sehen kannst und die, die Klassen haben müssen.
- Notiere die **Methoden**, die die Klassen vermutlich haben.
- Zeichne und beschrifte die Beziehungen zwischen den Klassen.



Minimal Version der "offensichtliche" Klassen



1.2 Erste Klassen

- Kopiere die **Snake** BlueJ-Projekt Vorlage aus dem Vorlagen-Verzeichnis.
- Erzeuge Objekte der bereits vorhandenen Klassen und untersuche ihre Methoden.
- Erstelle eine neue Klasse **Schlange**:
 - Ergänze die Klasse Schlange um das Attribut `String richtung`.
 - Ergänze die Klasse Schlange um ein Attribut `kopf` vom Typ `Rechteck`.
 - Implementiere den *Konstruktor* der Klasse Schlange:
 - * Es soll die Position übergeben werden.
 - * Die Startrichtung soll **"runter"** sein.
 - * Erzeuge einen neuen grünen Kopf an der richtigen Position mit der Größe 20x20.

1.3 Bewegung der Schlange

- Implementiere die Methode `bewegen` der Schlange.
- Je nach ihrer Richtung soll sich der Kopf der Schlange um 20 Pixel bewegen.
 Tipp: Verwende die Methode `Verschieben(int deltaX, int deltaY)` der Klasse `Rechteck`.
- Erzeuge ein `Schlangen`-Objekt und verschiebe es über den Bildschirm.

1.4 Wer ruft die Methoden der Klasse Schlange auf?

Eine Klasse verwaltet das gesamte Spiel, indem sie auf Nutzereingaben reagiert und die richtigen Methoden der anderen Objekte ausführt.

Welche Attribute hat diese Klasse?

1.5 🖥 Die Spiele Klasse

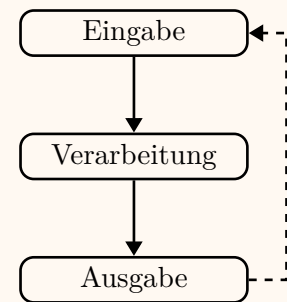
- Erstelle eine neue Klasse **Snake**:
 - Ergänze die Klasse **Snake** um ein **Referenzattribut** `schlange`.
 - Initialisiere das Attribut `schlange` im **Snake** Konstruktor an der Position (380, 280).

1.6 🖥 Nutzereingabe

Nutzereingabe

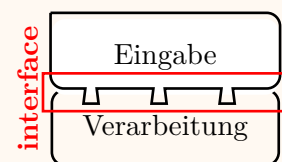
Unser Spiel übernimmt die Verarbeitung und teilt der Spiel-Engine die Veränderungen auf dem Bildschirm mit damit eine neue Ausgabe erzeugt werden kann.

Eine **Spiel-Engine** nimmt uns die meiste Arbeit ab, indem sie in einer Schleife die Nutzereingabe entgegen nimmt und an unser Spiel zur Verarbeitung weiterleitet.



Schnittstelle

Um in die Schleife integriert zu werden, muss unsere Spiel-Klasse alle vier durch das **Interface** `Zeichenfenster.AktionsEmpfaenger` vorgegebenen Methoden implementieren.



- Ergänze die Klasse **Snake** um die 4 Methoden des `AktionsEmpfaenger` Interfaces.

```
public void Ausführen()
public void Taste(char taste)
public void SonderTaste(int taste)
public void Geklickt(int x, int y, int anzahl)
```
- Definiere, dass die Klasse **Snake** das Interface `AktionsEmpfaenger` erfüllt.

```
class Snake implements Zeichenfenster.AktionsEmpfaenger
```
- Registriere neue **Snake**-Objekte als `AktionsEmpfänger`, indem du den Konstruktor um den Aufruf `Zeichenfenster.AktionsEmpfängerEintragen(this)`; erweiterst.

1.7 🖥 Steuerung mit der Tastatur

- Implementiere die Methode `Taste(char taste)` sodass die Richtung der Schlange angepasst wird.
 - 'W': rauf
 - 'S': runter
 - 'A': links
 - 'D': rechts
- Rufe `schlange.bewegen()` in der `Ausführen`-Methode auf.
- Teste dein Spiel, indem du ein neues **Snake**-Objekt erzeugst.
- **Für Schnelle:** Implementiert die Schlangensteuerung mit den Pfeiltasten und der Methode `Sondertaste` (Tipp: die Klasse `java.awt.event.KeyEvent` hilft).