

Übungsblatt 11

Abgabe: nein

Aufgabe 1 Einmal ein Lambda, bitte

Ersetzt in eurer Hauptklasse die Schleife, die durch alle Akteure läuft, um deren Methode *act* aufzurufen, durch einen Aufruf der Methode *forEach*.

Aufgabe 2 Strömen statt zählen

Ersetzt im Konstruktor der Klasse *Field* die zwei ineinander geschachtelten Schleifen zum Erzeugen der Spielobjekte durch *IntStreams*. Ein geeigneter Ansatz, um eine Zählschleife zu ersetzen, ist z.B. eine Kombination aus *iterate* und *forEach*.

Aufgabe 3 In Strömen rechnen

Nehmt die Methode *getNeighborhood* aus der Klasse *Field* aus der Musterlösung zu Übungsblatt 5 als Vorbild, in der die Nachbarschaftssignatur einer Zelle in einer Schleife berechnet wird. Erhaltet das Array *neighbors*, aber ersetzt den Rest der Methode durch einen einzigen Ausdruck, der mit Hilfe eines *IntStream* dasselbe Ergebnis berechnet. Hier könnt ihr auch euer Wissen über Bitoperationen einfließen lassen. Geeignete Methoden aus *IntStream* sind dabei *range* (oder wieder *iterate*), *filter*, *map* und *reduce* (oder *sum*).

Aufgabe 4 In Strömen sammeln

In der Klasse *Level* wird eine Textdatei in eine Liste von Strings eingelesen. Tatsächlich bietet die Klasse *BufferedReader* auch eine Methode *lines*, die einen *Stream<String>* liefert. Nutzt dessen Methode *toList*, um daraus die Liste zu generieren.

Aufgabe 5 Kreativ strömen

In den Musterlösungen seit Übungsblatt 3 gibt es in der Hauptmethode der Klasse *PI1Game* eine Schleife, die solange ausgeführt wird, bis die Spielfigur verschwunden ist. Könnt ihr die auch durch eine Anweisung ersetzen, die auf Java-Streams basiert?