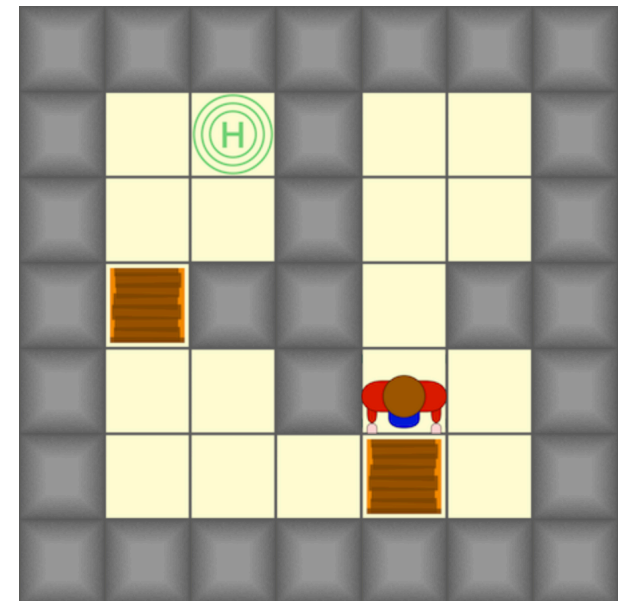
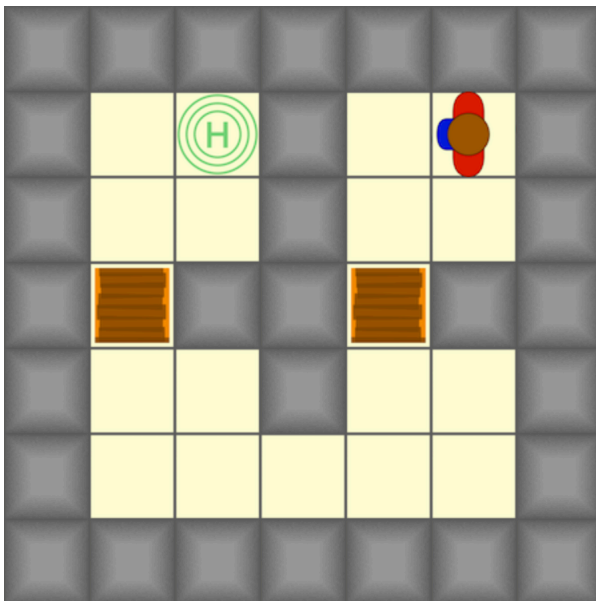


Projekt "DisCaptive" – Vorstudie

Bei DisCaptive handelt es sich um ein Denkspiel bei dem ein Gefangener aus einem Gefängnis flieht. Das Spiel ist natürlich politisch völlig korrekt:

- Der Gefangene wurde von einem Diktator zu Unrecht inhaftiert.
- Die Flucht läuft durchgängig gewaltfrei ab.

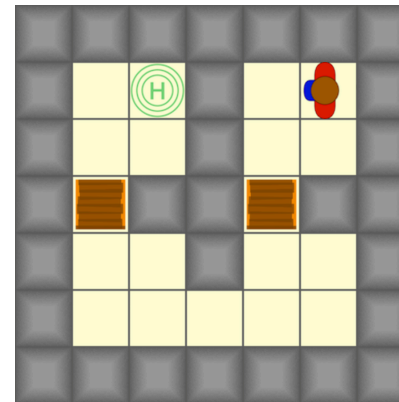
In jeder Spielstufe (*level*) von DisCaptive geht es darum, den Gefangenen Zug für Zug zum Zielfeld zu bewegen, wo er von seinen Fluchthelfern abgeholt wird. In dieser Vorstudie behindern nur Mauern und Kisten die Flucht. Kisten können zwar geschoben, aber nicht gezogen werden. Weil die Kisten schwer sind, kann immer nur eine Kiste geschoben werden.



Projekt "DisCaptive" – Kacheltypen

Technisch gesehen, läuft das Spiel auf einem kachelbasierten, rechteckigen Spielfeld (dem Gefängnis) ab. Die Spielfigur (der Gefangene) kann sich in jedem Schritt eine Kachel weit in die vier Hauptrichtungen bewegen, diagonale Züge sind nicht erlaubt. In der hier betrachteten, vereinfachten Grundversion des Spiels gibt es folgende Typen von Spielfeld-Kacheln:

- Mauer (undurchdringlich)
- leerer Gang
- Anfangsposition einer Kiste
- Startposition der Spielfigur
- Zielposition der Spielfigur

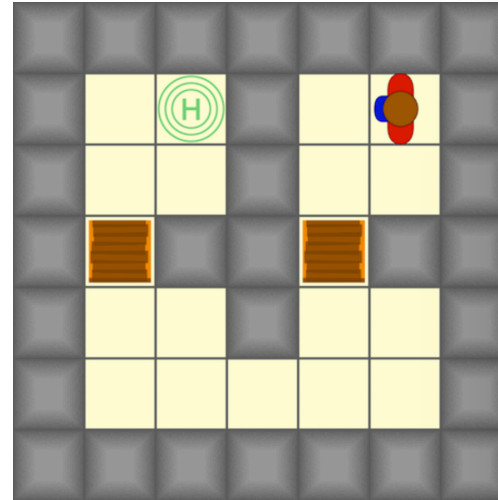


Es gibt in jedem Spielfeld n Kisten, mit $n \in \{1, \dots, 25\}$, sowie genau eine Startposition und genau eine Zielposition für die Spielfigur. Das Spielfeld ist gelöst, sobald die Spielfigur auf der Zielposition steht. Kisten können problemlos über die Startposition oder die Zielposition geschoben werden. Ein Spielfeld ist maximal 30 Kacheln breit und 20 Kacheln hoch.

Projekt "DisCaptive" – Dateiformat

Um die Räume für die einzelnen Spielstufen abzuspeichern wird ein einfaches Textformat verwendet. Im einfachsten Fall wird für jede Spielstufe eine eigene Textdatei verwendet. Die verwendete Darstellung bildet das spätere Erscheinungsbild des Raums nach und benutzt folgende Zeichen für die einzelnen Typen von Spielfeld-Kacheln:

- # Mauer (undurchdringlich)
- (Leerzeichen) leerer Gang
- \$ Anfangsposition einer Kiste
- @ Startposition der Spielfigur
- . Zielposition der Spielfigur



```
Level2.txt
#####
# .# @#
#  # #
#$###$##
#  # #
#      #
#####
```

Die Textdatei für den Beispiel-Raum ist oben abgebildet und besteht aus 7 Zeilen mit jeweils 7 Zeichen. Leerzeichen am Zeilenende können weggelassen werden und Räume müssen nicht quadratisch sein. Die Textdatei `Level4.txt` enthält ein entsprechendes Beispiel.

Vorstudie "DisCaptive" – Aufgabe 1

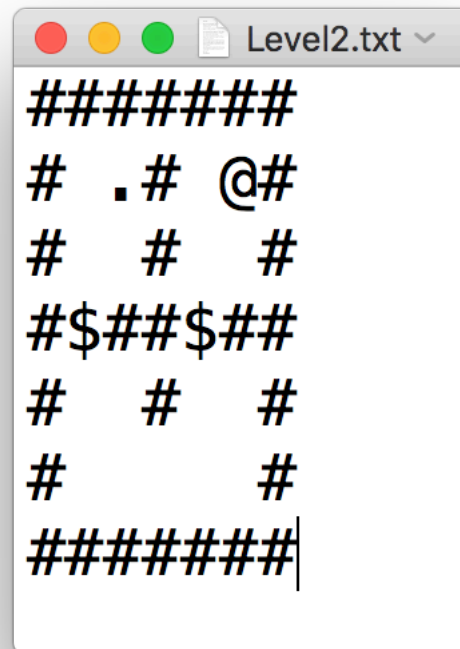
In dieser Vorstudie soll in mehreren Schritten ein einfacher Prototyp für das Basisspiel mit einer einfachen Textausgabe des Spielfelds entwickelt werden. Programmstücke aus dieser Vorstudie können Sie dann in der Meilenstein-Version und der vollständigen Version des Projekts wiederverwenden.

Teil 1: Einlesen und Ausgeben eines Gefängnisplans

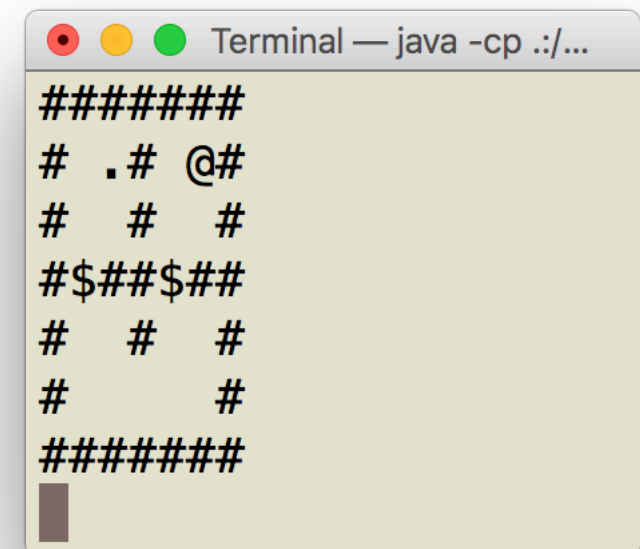
Schreiben Sie eine Funktion, die eines der vorgegebenen Beispiel-Spielfelder `Level1.txt` bis `Level5.txt` in ein zwei-dimensionales `char`-Array einliest und eine weitere Funktion, die das Spielfeld wieder als Text in ähnlicher Form auf den Bildschirm ausgibt.

Nützliche Funktionen:

- `In.open()`
- `In.done()`
- `In.readLine()`
- `In.close()`



```
#####
# .# @#
#   #
# $$$$ 
#   #
#       #
#####
```



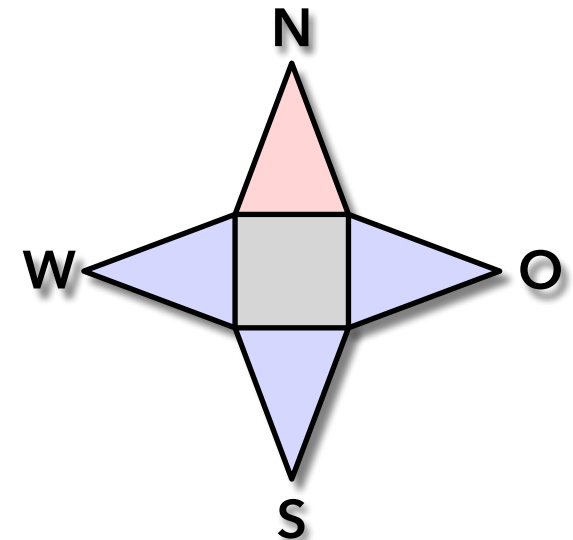
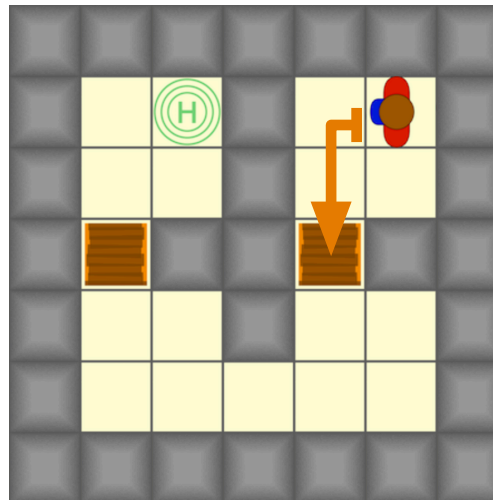
```
#####
# .# @#
#   #
# $$$$ 
#   #
#       #
#####
```

Vorstudie "DisCaptive" – Aufgabe 2

Teil 2: Bewegen der Spielfigur

Ergänzen Sie Ihr Programm um Funktionen zum Bewegen der Spielfigur. Durch Eingabe von zum Beispiel `wss<Return>` bewegt sich die Spielfigur ein Feld nach links und zwei Felder nach unten. Anschließend wird der Raum erneut auf den Bildschirm ausgegeben und die nächste Folge von Bewegungskommandos erwartet. Dabei werden die Anfangsbuchstaben der vier Himmelsrichtungen verwendet, um die Figur in die entsprechende Richtung zu bewegen.

Jedoch darf die Spielfigur nicht durch Mauern laufen und kann im Moment auch noch keine Kisten verschieben. Trifft die Figur auf eine Mauer oder Kiste soll die Meldung "Autsch!" ausgegeben und die Verarbeitung der Bewegungskommandos abgebrochen werden.



Vorstudie "DisCaptive" – Aufgabe 3

Teil 3: Verschieben der Kisten

Erweitern Sie Ihr Programm so, dass die Spielfigur beim Bewegen jetzt auch Kisten verschieben kann. Es kann nur eine Kiste zur Zeit verschoben werden und nur wenn das dahinter liegende Feld frei ist. Testen Sie Ihr Programm auch mit dem großen Beispiel-Spielfeld `Level99.txt`.

Teil 4: (optional) Erkennen eines gelösten Spielfelds

Ihr Programm soll jetzt zusätzlich erkennen, ob das Spielfeld gelöst worden ist, also die Spielfigur ihre Zielposition erreicht hat. Dann soll die Meldung "Flucht geglückt!" ausgegeben und die Verarbeitung weiterer Bewegungskommandos abgebrochen werden.

Das Dateiformat für Spielfelder benutzt das Zeichen `*` für eine Kiste, die bei Spielanfang auf der Zielposition für die Spielfigur steht, also die Kombination aus `$` und `.`. Verwenden Sie die gleiche Konvention in dem `char`-Array des Programms, auch wenn eine Kiste im Verlauf des Spiels über die Zielposition geschoben wird.

