

Bachelor-Studiengang Informatik

Übungen zur Vorlesung "Grundlagen der Informatik" (GDI), WS 2012/2013

Übungsblatt 9: Interfaces, geschachtelte Arrays

Ausgabe am: 18.12.2012

Abgabe am: 8.1.2013

Aufgabe 1:

Erstellen Sie ein Java-Programm, das von einem gegebenen Eingang ausgehend *alle* Wege durch ein Labyrinth findet.

Das Labyrinth ist von einer Textdatei einzulesen und in einem aus Zeichen bestehenden geschachtelten Array zu speichern, in dem die Wände durch '*' und die Wege durch Leerzeichen dargestellt sind. Der Eingang ist durch ein 'E' gekennzeichnet.

Nach dem Einlesen der Textdatei und dem Erstellen des Labyrinths erfolgt das Aufsuchen der Wege mithilfe der Methode *findAndPrintWaysOut* einer Klasse, die das Interface *Logic* implementiert. Diese Methode bekommt eine (Start-)Position im Labyrinth als Parameter. Falls dieser Standort leer ist, kommt er als möglicher Weg in Betracht und wird durch ein '+'-Zeichen markiert. Ist der Rand des Labyrinths erreicht, so ist ein Weg aus dem Labyrinth gefunden. Andernfalls werden die vier umliegenden Positionen durch rekursive Aufrufe der Methode *findAndPrintWaysOut* auf dieselbe Weise untersucht. Startposition ist der mittels 'E' gekennzeichnete Eingang. Die Methode *findAndPrintWaysOut* kann die Methoden des Interface 'Labyrinth' benutzen, das von einer anderen Klasse implementiert wird.

Da bei jedem erfolgversprechenden Aufruf der Methode *findAndPrintWaysOut* die Position im Feld durch ein '+'-Zeichen markiert wird, kann beim Erreichen des Randes ein Weg durch das Labyrinth ausgegeben werden. Die Wegmarkierungen verhindern auch, dass ein einmal begangener Weg ein weiteres Mal untersucht wird. Wichtig ist aber, dass die Markierung sowohl nach dem Ausgeben eines Weges als auch beim Zurückkehren aus einer Sackgasse wieder gelöscht wird.

Ihr Programm soll mit den beiden Labyrinthen von der Wiki-Seite funktionieren.

Hinweise

- Erstellen Sie eine Klasse *Start* mit lediglich einer Methode *main*, die alle nötigen Objekte erzeugt und den Programmablauf anstößt.
- Diejenige Klasse, die das *Logic*-Interface implementiert, kennt nicht die Klasse, die das *Labyrinth*-Interface implementiert (sondern nur das Interface *Labyrinth*). Die *Labyrinth*-implementierende Klasse muss überhaupt nichts von der *Logic*-implementierenden Klasse wissen.
- Die Klasse, die das *Labyrinth*-Interface implementiert, muss den Namen der Datei kennen; für das Bekanntmachen bietet sich ihr Konstruktor an.
- Die Klasse, die das *Logic*-Interface implementiert, muss das Labyrinth kennen; ...
- Alle Felder aller Klassen müssen *private* sein und dürfen nicht *static* sein.

Allgemeines

- Die Aufgaben sind in Eclipse zu bearbeiten. Legen Sie für die Bearbeitung dieses Übungsblattes ein Paket (engl. Package) namens *uebung09* an.
- Es sind Programmausdrucke (Listings) abzugeben, *keine* Ausdrucke von Testläufen. *Drucken Sie Ihre Listings aus Eclipse heraus, achten Sie auf eine vernünftige Formatierung auch beim Ausdruck!*
- Erlaubt sind *MakeItSimple*-Funktionen (keine nicht besprochene Funktionalität aus der Java Standard Bibliothek) und das bisher erworbene Wissen aus den GDI-Vorlesungen. Fragen Sie, bevor Sie Java-Konstrukte verwenden, die noch nicht behandelt wurden!
- Für die Bearbeitung dieses Übungsblattes gibt es Punkte im Bereich -2 bis +2.