

Programmierung eines Java- Spielprogramms

1. Einleitung

Ziel des Projekts ist die Programmierung eines Java-Spielprogramms für die Firma **TSE GmbH**. Diese hat beschlossen, ein eigenes Call-Center für die telefonische Bestellung, Reklamationen, Beratung und Service-Hotline einzurichten. Für eine entspannte Pausengestaltung der Mitarbeiter im Call-Center soll von Ihnen ein Spielprogramm entwickelt werden. Eine Abstimmung unter den Mitarbeitern favorisiert die Realisierung des sogenannten „Verschiebespiels“.

Bei dem zu programmierenden Spiel handelt es sich um eine Variante des Verschiebespiels, das im Folgenden beschrieben wird. Dieses soll als Java-Anwendung mit einer konsolenbasierten Benutzeroberfläche programmiert werden.

2. Beschreibung des Verschiebespiels

Das Verschiebespiel stammt aus dem Jahr 1874 und wurde von dem Postangestellten Noyes Palmer Chapman erfunden. Es wird auf einem quadratischen Spielbrett mit 3 x 3 Feldern mit nur einem Spieler gespielt. Dabei werden zu Beginn auf dem Spielbrett mit insgesamt neun Feldern, acht (von 1 bis 8) nummerierte Spielsteine beliebig angeordnet und ein Feld freigelassen. Ziel ist es nun durch sukzessives Verschieben der Spielsteine das Spiel in die definierte Endstellung zu bringen, bei der alle Spielsteine zeilenweise aufsteigend sortiert angeordnet sind und das leere Feld rechts unten ist. Dabei dürfen die Spielsteine nur in horizontale und vertikale Richtung, aber nicht diagonal verschoben werden. Vergleichen Sie hierzu Abbildung 1.

Beispielhafte Startstellung

5		2
6	3	1
4	7	8

Mehrere Verschiebungen →

Endstellung

1	2	3
4	5	6
7	8	

Abbildung 1: Start- und Endstellung

In der abgebildeten Startdarstellung ist als erster Spielzug das Verschieben der „2“, der „3“ oder der „5“ möglich. Es sind beliebig viele Spielzüge erlaubt um die Endstellung zu erreichen. Ist die Endstellung erreicht endet das Spiel.

3. Grundlegende Aufgaben

Programmieren Sie das beschriebene Verschiebespiel als Konsolenanwendung in der Sprache Java. Folgende Punkte sind dabei zu beachten.

- Nachdem Starten des Programms wird das Spielbrett mit einer zufälligen Startaufstellung ausgegeben. Recherchieren Sie bei Bedarf im Internet nach dem Generieren von Zufallszahlen in Java.
- Der Spieler kann nun einen Spielzug ausführen, indem er eines der dem leeren Feld benachbarten Felder durch die Eingabe der entsprechenden Zahl über die Tastatur auswählt.
- Das ausgewählte Feld wird anschließend mit dem leeren Feld getauscht und das Spielbrett erneut auf der Konsole ausgegeben.
- Die Anzahl der Spielzüge wird beginnend bei eins mitgezählt und nach jedem Spielzug ausgegeben.

- Nach jedem Spielzug muss geprüft werden, ob die definierte Endstellung erreicht ist (siehe Abbildung 1).
- Ist die Endstellung erreicht, dann ist das Spiel beendet.

4. Projektmanagement

1. Projektplanung:

- Erstellen Sie einen einfachen Projektplan, der den zeitlichen Ablauf und Meilensteine für die Entwicklung des Spiels skizziert.
- Berücksichtigen Sie Phasen wie Analyse, Design, Implementierung, Tests und Dokumentation.
- In der Projektplanung soll ebenfalls dokumentiert sein, welches Teammitglied welche Aufgabe übernimmt.

2. Risikomanagement:

- Identifizieren Sie potenzielle Risiken im Zusammenhang mit der Entwicklung des Java-Spielprogramms.
- Entwickeln Sie Strategien zur Risikominimierung und einen Plan für den Umgang mit auftretenden Problemen.

3. Code- Dokumentation:

- Erstellen Sie eine umfassende Dokumentation, die den Code erklärt, Designentscheidungen beschreibt und den Benutzern eine klare Anleitung bietet.
- Integrieren Sie Kommentare im Code, um die Lesbarkeit zu verbessern.

4. Qualitätssicherung:

- Führen Sie regelmäßige Code-Reviews durch, um die Codequalität zu verbessern.

5. Abgabe:

Die Punkte des Projektmanagements sind in einer PowerPoint-Präsentation zusammenzutragen und bis zum 5.12.24 auf Moodle abzugeben.

6. Erweiterte Aufgaben

- Selbst gewählte Startstellung: Die Startstellung soll anstatt durch einen Zufallsgenerator durch den Spieler selbst gewählt werden können. Der Spieler soll dazu nach dem Programmstart auswählen können, auf welche Art die Startstellung erzeugt wird.
- Weiterspielen nach Spielende: Nach einem Spiel soll der Spieler gefragt werden, ob er das Programm beenden oder eine weitere Runde spielen möchte.
- Installieren Sie Java auf dem Pi damit das Spiel über SSH auf dem Pi gespielt werden kann.
- Sinnvoller Einsatz von OOP
- Nutzen Sie git für die Kollaboration mit Ihrem Teamkollegen/Teamkollegin.
- Implementieren Sie das Spiel in Processing mit einer grafischen Oberfläche.
- Beliebige Spielbrettgröße: Der Spieler soll am Anfang nach der Seitenlänge des quadratischen Spielbretts gefragt werden. Somit sind auch größere Spielbretter als 3 x 3 möglich. Das Spielbrett soll jedoch immer in quadratischer Form erstellt werden.
- Speichern des Spielstandes: Das Spiel soll nach jedem Spielzug unterbrochen werden können, z.B. wenn der Spieler nach einem Zug eine bestimmte Taste drückt. Daraufhin soll der Spielstand in eine Datei

gespeichert werden. Wird das Programm neugestartet, soll zu Beginn der gespeicherte Spielstand aus der Datei wieder eingelesen werden können.

- **Computerspieler:** Programmieren Sie einen Computerspieler, der versucht das Spiel zu lösen. Die Funktion kann nach jedem Spielzug aktiviert werden. Der Computer versucht dann mit der aktuellen Brettstellung zu spielen.
- **Spiel über Netzwerk beobachten:** Das Spiel kann von anderen Computern aus über das Netzwerk beobachtet werden. Als Protokoll können Sie entweder TCP (Client-Server) oder MQTT (Publish-Subscribe) verwenden. Die Spielzüge werden dann über das Netzwerk übertragen und auf den verbundenen Computern angezeigt.

7. Hinweise zur Programmierung

Verwenden Sie Methoden um ihren Programmcode zu gliedern. Für die Abbildung des Spielbretts können Sie ein Array, ein zweidimensionales Array oder eine Liste verwenden. Überlegen Sie sich eine geeignete Darstellung für das leere Feld.

Eine Möglichkeit wäre das Programm nach folgenden Methoden zu gliedern. Diese Gliederung ist nur ein Vorschlag und muss nicht so übernommen werden. Überlegen Sie sich für jede Methode die benötigten Parameter sowie Rückgabetypen.

Methoden	Beschreibung
printBoard	Gibt das Spielbrett mit aktueller Stellung am Bildschirm aus.
initBoardRandom	Erzeugt auf dem Spielbrett eine zufällige Stellung. Kann zum Erzeugen der Startstellung genutzt werden.
swapFields	Tauscht zwei Felder auf dem Spielbrett.
isAdjacentFields	Prüft, ob zwei Felder horizontal oder vertikal benachbart sind.
getAdjacentFields	Ermittelt alle Felder, die zum angegebenen Feld horizontal oder vertikal benachbart sind.
isGameOver	Prüft, ob die aktuelle Stellung die Endstellung ist.
getFieldIndex	Ermittelt die Position (Index) zu einem Feld.
getEmptyFieldIndex	Ermittelt die Position (Index) des leeren Feldes.
main	Enthält die Hauptschleife und verwendet die anderen Methoden.

8. Organisatorisches und Bewertung

- Der Projektzeitraum beginnt um 7:40 Uhr und endet um 15:00 Uhr.
- Die Aufgabe ist in zweier Gruppen zu bearbeiten.
- KI ist lediglich für die Lösung der Zusatzaufgaben gestattet. Die Lehrkraft behält sich das Recht vor, die Note bei Verdacht auf unzulässige Nutzung zu ändern. Normale Internetrecherche ist zulässig.
- Ab 13:30 Uhr beginnt die Lehrkraft mit der Abnahme. Dabei stellen Sie ihr Programm der Lehrkraft vor und erklären Ihre Vorgehensweise und Ihren Programmcode.
- Nach der Abnahme bewertet die Lehrkraft Ihr Programm mit Hilfe des vorgesehenen Abnahmeprotokolls. Sie können maximal 100 Punkte erreichen.
- Durch die Bearbeitung aller grundlegenden Aufgaben erreichen Sie 80 von 100 Punkten.
- Mit den erweiterten Aufgaben können zusätzliche Punkte erzielt werden.
- Die Präsentation ist zum 5.12.24 in Moodle abzugeben. Die Projektkompetenznote bildet sich aus der technischen Abnahme und der Bewertung der Präsentation, wobei beide Teile gleich gewichtet werden.
- Neben dem Inhalt wird auch der Aufbau und die Gestaltung für die Bewertung betrachtet.