Junioraufgabe 1: Auf und Ab

Team-ID: 00130

Team-Name: ThinkPadHacker

Bearbeiter dieser Aufgabe:
Yannik Schiebelhut

23. November 2018

Inhaltsverzeichnis

Lösungsidee

Für alle Ziffern, die sich auf einem herkömmlichen Würfel befinden, wird der Spielplan durchlaufen. Dabei wird gezählt, wie oft gewürfelt werden muss und festgestellt, ob das Ziel überhaupt erreicht wird.

Umsetzung

In einer Schleife wird für die Werte 1-6 der folgende Prozess durchlaufen:

- Erstelle ein Array aus Booleans (für jedes Feld ein Index). Da die Schrittgröße je Durchlauf konstant ist, wird der Versuch das Ziel zu erreichen abgebrochen, wenn das selbe Feld zweimal betreten wird.
- Bis im Ziel oder Abbruchbedingung erreicht:
  - Gehe den aktuellen Wert vorwärts.
  - Prüfe, ob eine Leiter an das Feld angrenzt. Falls ja, betrete das Feld, auf das diese Leiter zeigt.
  - Prüfe, ob das Feld schon einmal betreten wurde. Ist dies der Fall, beende den Durchlauf mit einer Fehlernachricht, sonst vermerke, dass das Feld betreten wurde.
  - Erhöhe einen Zähler für die erfolgten Durchläufe um eins.

Beispielhafter Schritt:

Schrittgröße = 1;

aktuelles Feld = 5;

// gehe einen Schritt vorwärts

aktuelles Feld = 6;

prüfe auf Leitern; // findet Leiter von Feld 6 zu Feld 27

aktuelles Feld = 27;

falls (war auf Feld [27] == false)

war auf Feld [27] = true;

gegangene Schritte += 1;

sonst

// gebe aus, dass das vollenden mit **Schrittgröße** unmöglich ist

Quellcode

```
public class J1 {


	/**
	 * The main method of the program. All essential work is done here.
	 *
	 * @param args This program doesn't require arguments.
	 */
	public static void main(String[] args) {
		// repeat procedure for every value of the dice
		for (int stepDistance = 1; stepDistance <= 6; stepDistance++) {
			int field = 0; // pointer to the current field
			int numTry = 0; // counts how often the dice has been rolled to finish the game


			// prepare prevention of endless run
			boolean[] wasOnField = new boolean[101];
			for (int i = 0; i < wasOnField.length; i++) {
				wasOnField[i] = false;
			}
```

```
                        // actual try
                        while (field != 100) {
                                // go step
                                field += stepDistance;
                                numTry++;

                                // if greater than hundred, go backwards
                                if (field > 100) {
                                        int tmp = field - 100;
                                        field = 100 - tmp;
                                }

                                // endless loop prevention / detection
                                if (wasOnField[field] == true) {
                                        // message on endless loop
                                        System.out.printf("Finishing the game with step distance %d is
impossible!\n", stepDistance);
                                        break;
                                } else {
                                        wasOnField[field] = true;
                                }

                                field = handleLadders(field);

                                // message on success
                                if (field == 100) {
                                        System.out.printf("Game finished with stepDistance %d!\tNeeded to
roll the dice %d times!\n", stepDistance, numTry);
                                }
                        }
                }
        }
```

```
	/**
	 * This method is used to keep the overview in the main method. It simply
	 * resolves the given filed into the one, the ladder is pointing to, if one
	 * exists. If no ladder exists, pField is returned.
	 *
	 * @param pField the field to be resolved
	 * @return field after transformation by a ladder
	 */
	public static int handleLadders(int pField) {
		// switch für Leitern
		// siehe J1.java
		}
	}
}
```