

2. Skispringen

Diese Aufgabe ist eine Übung zur Festigung der Anwendung der Methoden der Klasse `List`.

Für den Start in umgekehrter Reihenfolge kann auch ein Objekt der Klasse `Stack` verwendet werden.

Aufgabe:

Ein Skispringen hat folgenden Ablauf:

- *Am ersten Durchgang nehmen 50 Skispringer teil.*
- *Nach dem Sprung erhält der Springer eine Punktzahl und wird nach dieser Punktzahl in eine Rangliste eingeordnet.*
- *Die besten 30 Springer qualifizieren sich für den zweiten Durchgang.*
- *Sie starten in umgekehrter Reihenfolge gegenüber der Platzierung auf der Rangliste.*
- *Nach dem Sprung erhält der Springer wiederum eine Punktzahl und wird nach der Gesamtpunktzahl aus beiden Durchgängen in die endgültige Rangliste eingeordnet.*

Implementieren Sie die Simulation eines Skispringens auf Basis eines Prototypen.

Die Punktzahl des ersten Durchgangs soll eine zufällige Zahl zwischen 80 und 120 mit einer Nachkommastelle sein. Die Punktzahl des zweiten Durchgangs soll sich als zufällige Zahl zwischen 80 % und 120 % der Punktzahl des ersten Durchgangs mit einer Nachkommastelle ergeben. So wird simuliert, dass starke und schwache Springer in etwa im selben Leistungsbereich bleiben.

Zusatzaufgabe für besonders leistungsstarke Lernenden:

Es ist auch möglich, dass mehrere Springer mit derselben Punktzahl auf dem 30. Platz des ersten Durchgangs liegen. In diesem Fall qualifizieren sich alle diese Springer für den zweiten Durchgang.

Hier wird alternativ die Realisierung mit Assoziationen zu Listen statt mit einer Unterklasse der Klasse `List` benutzt. Die Unterklasse wird nur verwendet, um die Typkonversionen zu vermeiden und die Liste in eine Zeichenkette zu konvertieren.

Ein Hinweis zum Prototyp:

Leider müssen die Punktzahlen in der Klasse `Skispringer` mit Hilfe einer Methode „zahlMit1Nachkommastelle“ mit einem Trick auf eine Nachkommastelle gerundet werden. Denn ohne diesen Darstellungstrick werden Punktzahlen manchmal mit Rundungsfehlern wie „95.900000001“ dargestellt.

Das fertige Programm zeigt etwa folgendes Ergebnis:

