Universität Potsdam
Institut für Computational Science und Informatik
Sommersemester 2024

Praxis der Programmierung

Hausaufgabenprojekt zur Pogrammierung in Java

Abgabe: Laden Sie Ihre Lösungen bis zum 7. Juli 2024 um 23:59 Uhr auf Moodle hoch.

Einführung: Ein elektrisches Netzwerk (kurz: Netz) ist wie folgt definiert:

- Ein Widerstand mit einem Widerstandswert R, gemessen in Ohm, ist ein (sehr einfaches) Netz.
- Zwei Netze mit den Widerstandswerten R_1 und R_2 können in Reihe (seriell), d.h. hintereinander, geschaltet werden. Das Ergebnis ist ein neues Netz mit dem Widerstandswert $R = R_1 + R_2$.
- Zwei Netze mit den Widerstandswerten R_1 und R_2 können parallel, d.h. nebeneinander, geschaltet werden. Es entsteht ein neues Netz mit dem Gesamtwiderstand R, für den gilt:

$$\frac{1}{R} = \frac{1}{R_1} + \frac{1}{R_2}$$

Aufgaben:
In den im Folgenden beschriebenen Klassen sind alle Datenelemente so stark wie möglich zu kapseln (durch Verwendung der Modifier `private` oder `protected`); die Methoden und die Klassen selbst sollen öffentlich sein, solange nichts Gegenteiliges verlangt wird.

1. Definieren Sie eine abstrakte Basisklasse `Net` für elektrische Netze mit einem Datenelement `resistors` vom Typ Array von Strings, das alle im Netz vorkommenden Widerstände auflistet und die eindeutigen Namen (IDs) aller Widerstände als Strings enthält.

   Ferner gibt es

   (a) eine abstrakte Methode `ohm()` mit Rückgabetyp `double`, die den Gesamtwiderstand des Netzes zurückgibt;

   (b) eine implementierte Methode `show()`, die das Array `resistors` durchläuft und alle IDs der Widerstände im Netz zu einem String zusammensetzt und zurückgibt;

   (c) eine implementierte, statische Methode `checkIDs`, die zwei Netze als Argumente entgegennimmt. Sie prüft, ob (mindestens) eine ID eines Widerstands existiert, der in beiden Netzen vorkommt, und gibt einen entsprechenden `boolean`-Wert zurück.

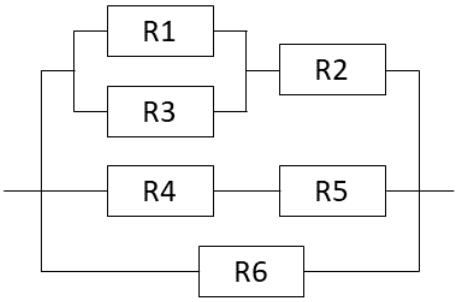
2. Leiten Sie eine Klasse `Resistor` für einfache Widerstände ab. Der Widerstandswert vom Typ `double` wird in einem Datenelement gespeichert und ist unveränderlich.

   Die Klasse enthält einen Konstruktor mit einem `double`-Parameter und einem String-Parameter für den Widerstandswert und für die ID. Beide Datenelemente der Klasse werden initialisiert. Das Array `resistors` hat dabei die Länge 1.

3. Leiten Sie von `Net` die Klassen `Serial` für serielle und `Parallel` für parallele Schaltungen von Netzen ab. Diese Klassen haben zwei zusätzliche Datenelemente vom Typ `Net`, die die seriell bzw. parallel verschalteten Netze speichern. Zum Erzeugen eines neuen Objekts der jeweiligen Klasse werden die statischen Methoden `newParallel` und `newSerial` genutzt. Diese erhalten jeweils zwei Netze als Argumente und prüfen mit `checkIDs`, ob die in den übergebenen Netzen vorkommenden Widerstände konfliktfrei in das neue Netz überführt werden können.
   Ist die konfliktfreie Überführung möglich, rufen die Methoden den Konstruktor der jeweiligen Klasse mit den beiden übergebenen Netzen auf und geben das erzeugte Objekt zurück.
   Ist die konfliktfreie Überführung nicht möglich, wird auf der Konsole eine Fehlermeldung ausgegeben und die Methoden geben kein Objekt (also die Null-Referenz) zurück.

   Die Konstruktoren der beiden Klassen sind `private`. Sie erwarten je zwei Widerstandsnetze als Argumente und speichern diese als Werte der ensprechenden Datenelemente. Das neue Array `resistors` enthält die IDs aller Widerstände aus beiden Netzen. Die Berechnung des Widerstandswerts können Sie der Einführung entnehmen.

4. Schreiben Sie eine Anwendung, die folgendes Netzwerk aufbaut und den Gesamtwiderstand sowie die IDs aller Widerstände auf die Konsole ausgibt. Nutzen Sie, wenn immer es möglich ist, die Methoden Ihrer Klassen! Die Widerstände R_1 bis R_6 haben dabei die Werte 100, 200, 300, 400, 500 bzw. 600 Ohm.



5. Als neue Sorte von Widerständen werden Potentiometer als elektrische Bauteile mit einem regelbaren Widerstandswert eingeführt. Leiten Sie von `Net` eine weitere Klasse `Potentiometer` ab, die zusätzlich einen Setter `setOhm` für den Widerstandswert bietet. Der Konstruktor enthält nur einen String-Parameter für die ID und setzt den Widerstandswert anfänglich auf `0.0` Ohm.

6. Ersetzen Sie in dem oben skizzierten Beispielnetz den Widerstand R_4 durch ein Potentiometer. Schreiben Sie eine neue Anwendung, die die Widerstandswerte der modifizierten Schaltung ausgibt, wenn das Potentiometer von 400 Ohm bis auf 5 Kiloohm in Schritten von 200 Ohm hochgeregelt wird.

Allgemeine Anforderungen an Abgaben:

- *Der Kommentarblock am Anfang jedes Quellcodes muss Ihren Namen und Ihre Matrikelnummer enthalten.*
- *Geben Sie Ihre Programme als .java-Dateien ab.*
- *Alle Programme müssen auf dem Linux-Server* `ts-linux.cs.uni-potsdam.de` *ausführbar sein. Wir empfehlen dies vor der Abgabe mind. einmal zu testen (z.B. im PC-Pool).*