Entscheiden und begründen Sie, ob mit der angegebenen Signatur die Methode gültig ist.

|  |  |
| --- | --- |
| Methode | Begründung |
| public void static muh(int futter){..} | Gültig |
| public static oink(boolean suhle, double temp) {..} | Ungültig; Kein Rückgabedatentyp |
| public static String putput(int koerner;booelan fuchs) {..} | Ungültig; Die Parametern sind von eine semikolon getrent |
| public static double Blubblub(boolean genugWasser) {..} | Gültig |
| public static int kikeriki {..} | Gültig |

Schreiben Sie die **Fachklasse** *RundeForme*n mit den drei statischen Methoden für

* Die Kreisfläche (A = r² \* π),
* Die Oberfläche einer Kugel (A = 4 \* r² \* π),
* Das Volumen einer Kugel (V = 4/3 \* r³ \* π).

Legen Sie sinnvolle Methodennamen, gemäss Konvention fest.   
Kommentieren sie die Methodenköpfe gemäss JavaDoc Syntax.

Schreiben Sie dazu eine Testklasse mit Namen *RundeFormenTest*, mit der Sie verschiedene Werte ausprobieren können, ohne das Programm neustarten zu müssen.

Schreiben Sie eine **Fachklasse** *MyMath* mit folgenden statischen Methoden

* pow - soll die **n**-te Potenz von **x** als Rückgabewert liefern (xn),
* fak – soll die Fakultät einer Zahl berechnen, also 5! = 1\*2\*3\*4\*5 = 120 ,
* root – soll die n-te Wurzel aus einer Zahl ziehen, siehe Infoblatt,
* pythagoras – soll mit den Parametern für die Seiten a und b als Ergebnis die Hypotenuse (c) liefern (a² + b² =c²).

Sie dürfen die (echte) Math Klasse nicht benutzen. Schreiben Sie eine Testklasse mit Namen *MyMathTest*, in der Benutzer mit einem Menü (**Tip**: switch case verwenden) die Korrektheit der implementierten Methoden testen kann.

Die trigonometrische Funktion sin(x) kann durch die folgende unendliche Potenzreihe dargestellt werden:

Wird diese Reihe nur bis zu einem bestimmten Glied auswertet, erhält man einen Näherungswert für sin *x*. Man kann beispielsweise abbrechen, wenn nicht mehr als  des aufgelaufenen Näherungswerts ausmacht. Entwickeln Sie ein entsprechendes Programmfragment und testen Sie es mit verschiedenen Werten von *x* (im Bereich 0 – ) und .