

Programmieren in Java

Einige **Übungen** zum Kapitel **Anweisungen**

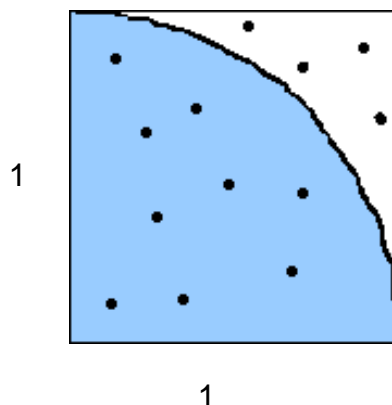
Aufgabe 1: (10 Punkte)

Schreiben Sie ein Programm, das die Summe und den Mittelwert einer Anzahl von positiven ganzen Zahlen berechnet. Lesen Sie die Zahlen in einer Schleife nach und nach von der Tastatur ein und führen Sie dabei gleich die nötige Verarbeitung durch. Brechen Sie die Eingabeschleife ab, sobald eine negative Zahl eingegeben wurde.

Zum Einlesen von ganzen Zahlen können Sie die Klassenmethode `readInt()` der Klasse `TastaturEingabe` verwenden.

Aufgabe 2: (10 Punkte)

In dieser Aufgabe geht es darum, die Zahl π genähert zu berechnen. Um π zu ermitteln, genügt es, die Fläche des Viertels des Einheitskreises zu berechnen. Der Einheitskreis ist ein Kreis mit dem Radius 1. Da die Fläche des Einheitskreises gleich π ist, muss man nur die ermittelte Fläche des Viertelkreises mit 4 multiplizieren um auf π zu kommen. Diese Idee ist in der folgenden Abbildung dargestellt. Man bestimmt zufällige Punkte in einem Einheitsquadrat (d.h., in einem Quadrat mit der Kantenlänge 1). Die Fläche unter dem Viertelkreis ist dann ungefähr gleich der Anzahl der Punkte, die unter der Kreislinie liegen, dividiert durch die Gesamtzahl der Punkte. Dieses Verfahren zur Ermittlung einer Fläche (allgemeiner: eines bestimmten Integrals) heißt wegen seines Zufallscharakters auch Monte-Carlo-Integration.



Schreiben Sie eine Klasse `Pi`, welche eine Methode `main()` enthält, die mittels Monte-Carlo-Integration einen Näherungswert für π berechnet. Lesen Sie dazu eine ganze Zahl ein, die angibt, wie viele Punkte „gewürfelt“ werden sollen. Würfeln Sie entsprechend oft und berechnen Sie den Quotienten aus der Anzahl der Punkte unter der Kreislinie durch die Gesamtzahl der Punkte. Ein Punkt (x, y) liegt unter der Kreislinie, wenn $x^2 + y^2 \leq 1.0$ gilt.

Zum „Würfeln“ eines Punktes (x, y) ermitteln Sie für die x - und y -Koordinate jeweils eine reelle Pseudo-Zufallszahl r mit der Eigenschaft $0.0 \leq r \leq 1.0$. Eine solche Pseudo-Zufallszahl erhalten Sie z.B. durch den folgenden Methodenaufruf:

```
double r = Math.random();
```