

## Praktikum „Informatik 2 für ET/MT/RES“

### 3. Praktikumsaufgabe

In dieser Praktikumsaufgabe soll das Spielfeld erstellt werden, auf dem sich später der Roboter bewegen wird.

In Navigationssystemen werden Wege durch die Angabe der abzufahrenden Wegpunkte, der POI (Points of Interest) festgelegt. Der Benutzer soll daher die Möglichkeit erhalten, Wegpunkte über die Konsole einzugeben. Anschließend soll der Robotersimulator einen kurzen Weg finden, auf dem alle Wegpunkte erreicht werden. In einer späteren Übungsaufgabe werden Sie die graphische Oberfläche erstellen und dann den Roboter diesen Weg fahren lassen.

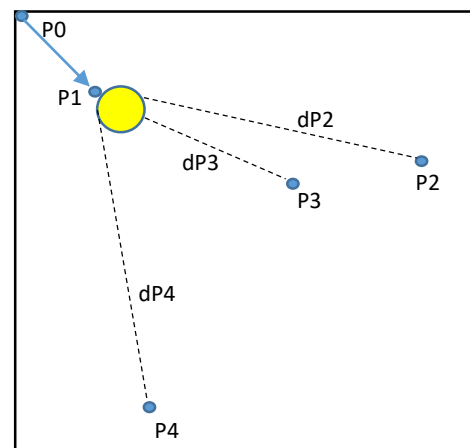
1. Erstellen Sie die Klasse `Spielfeld`. Der Konstruktor bleibt leer, er wird in einer späteren Übung programmiert. Definieren Sie Integer-Konstanten für die Breite und die Länge des Spielfeldes so, dass Sie das Spielfeld gut auf Ihrem Bildschirm sehen können (z.B. 1000 x 1000 Pixel).
2. Erstellen Sie die Methode `Punkt[] punkteEingeben()`. Der Benutzer soll zunächst über die Konsole eingeben, wie viele Punkte er haben möchte. Anschließend soll er die Koordinaten der Punkte eingeben. Die Punkte werden in einem Array gespeichert, das als Rückgabewert der Methode zurückgegeben wird.

Zu Beginn soll sich der Roboter in der linken oberen Ecke des Spielfelds befinden, d.h. der Punkt (0,0) ist der erste Punkt des Arrays.

Verwenden Sie zur Konsoleneingabe die Klasse `Scanner`, die Sie mit `import java.util.*` einbinden müssen. Mit der Methode `nextInt()` können Sie die Koordinaten eines Punktes als einzelne Ganzzahlwerte einlesen. Informieren Sie sich im Internet über die Verwendung der Klasse `Scanner`.

Alle eingegebenen Punkte müssen im Spielfeld liegen. Machen Sie den Benutzer darauf aufmerksam, wenn er einen ungültigen Wert eingibt. Gehen Sie in dieser Übung davon aus, dass der Benutzer nur ganze Zahlen eingibt, aber keine Buchstaben oder Sonderzeichen. In einer späteren Vorlesung werden Sie Exceptions kennen lernen, mit denen solche Fehleingaben erkannt werden können.

3. Erstellen Sie die Methode `poiSortieren(Punkt[] poi)`, in der die Punkte aus dem Array `poi` in der abzufahrenden Reihenfolge sortiert werden. Der Roboter soll sich am Anfang links oben im Spielfeld befinden. Dann soll er für jeden Punkt herausfinden, welcher noch nicht angefahrne Punkt am nächsten ist. In dem Beispiel ist der Roboter bereits von P0 zu P1 gefahren. Als nächstes fährt er zu P3, weil die Strecke dP3 kürzer ist als dP2 und dP4. In dem Beispiel werden also die Punkte in dieser Reihenfolge sortiert: P0 - P1 - P3 - P2 - P4.



Abschließend sollen die Punkte und die jeweiligen euklidischen Abstände auf der Konsole ausgegeben werden.

4. Erstellen Sie die Methode `poiAbfahren()`, in der der Benutzer zunächst die Punkte eingeben kann und anschließend der kürzeste Weg berechnet wird.

Sie können den Debugger verwenden, um Fehler im Code zu finden. Setzen Sie zunächst einen Haltepunkt in der Funktion, die Sie testen möchten. Klicken Sie dazu links auf den weißen Streifen, Sie erkennen den Haltpunkt am Stop-Zeichen. Starten Sie Ihr Programm. Sobald der Haltepunkt erreicht ist, wird der Debugger von BlueJ angezeigt, mit dem Sie die Befehle Ihres Programms Schritt für Schritt durchgehen können. Überprüfen Sie nun, ob die Variablen korrekte Werte haben.