

## Workshop Einführung in die Robotik

### Indoor Robot „Stan“

Betreut von Ina Nowak und Angelika Korinth

#### Einführung

Hier habt Ihr die Möglichkeit, selbst einen mobilen Roboter mit der Programmiersprache C++ zu programmieren. Dazu haben wir Euch einige Dinge vorbereitet. Um Euch das Programmieren zu erleichtern, findet Ihr bereits vorgefertigte Klassen und Funktionen. Auf dem Bildschirm sind bereits für jede Gruppe die Klasse 'Girls.cpp' und 'Main.cpp' geöffnet. In der Klasse 'Girls.cpp' dürft Ihr ein paar Bewegungsabläufe für unseren Roboter Stan programmieren. Dazu existieren bereits vorhandene Funktionsköpfe, die Ihr nur ausfüllen müsst. Um unten stehende Aufgaben zu lösen, könnt Ihr folgende Funktionen verwenden:

- **bewegung->fahreVorwaerts(double strecke)**  
Diese Funktion veranlasst Stan eine in Millimetern angegebene Strecke geradeaus zu fahren.
- **bewegung->rotate(double grad)**  
Diese Funktion veranlasst Stan sich um einen angegebene Gradzahl nach links zu drehen.
- **bewegung->bleibStehenSolangeEtwasImWegIst(double abstandZumHindernis)**  
Diese Funktion veranlasst Stan erst loszufahren, wenn der Weg frei ist, d.h. wenn sich vor ihm kein Hindernis in einem von Euch angegeben Abstand befindet.
- **bewegung->weicheAus(double abstandZumHindernis, int anzahlHindernisse)**  
Diese Funktion veranlasst Stan, sich nach Erkennen eines Hindernisses um 90° zu drehen und anschließend weiter zu fahren. Stan weicht einer festgelegten Anzahl von Hindernissen aus, anschließend fährt er einfach geradeaus weiter, wenn er nicht durch Aufruf einer passenden Funktion gestoppt wird.
- **bewegung->stop():** Diese Funktion veranlasst Stan anzuhalten.
- **bewegung->drive():** Diese Funktion veranlasst Stan geradeaus zu fahren. Er hält aber nicht mehr von alleine an.



Roboter Stan

# Aufgaben

## 1. void Girls::fahreGeradeAus()

Bei dieser Aufgabe soll Stan eine von Euch angegebene Strecke geradeaus fahren. Die Strecke muss in Millimetern angegeben werden (z.Bsp.: 1m = 1000mm).

## 2. void Girls::fahreNachVorneUndDreheDichUmUndFahreWiederZurück()

Bei dieser Aufgabe soll Stan eine von Euch angegebene Strecke nach vorne fahren, sich umdrehen und wieder zurückfahren.

## 3. void Girls::quadrat()

Bei dieser Aufgabe soll Stan die Kanten eines Quadrats abfahren. Dabei kann z.B. eine Kantenlänge von 1m angegeben werden. Beachte dabei den zur Verfügung stehenden Raum.

## 4. void Girls::fluechte()

Bei dieser Aufgabe soll Stan erst losfahren, wenn sich kein Hindernis vor ihm befindet. Sobald Stan fährt und ein Hindernis vor ihm auftaucht, soll er ausweichen. Achtet darauf, dass Stan irgendwann auch wieder anhält und die Ultraschallsensoren erst Hindernisse ab einer bestimmten Höhe erkennen.

Die oben programmierten Funktionen werden in der Klasse Main.cpp aufgerufen. Allerdings darf immer nur eine Funktion aktiviert werden, die dann von Stan ausgeführt wird. Andere Funktionen müssen auskommentiert sein. Sämtliche Funktionen sind in der Main-Klasse bereits angegeben.

Um das Program in Stans Sprache zu übersetzen, muss es compiliert werden. Dazu gibt man auf der Konsole den Begriff „make“ ein. Sobald make fehlerfrei zurückgekommen ist, kann das Programm ausgeführt werden. Dazu gibt man auf der Konsole den Begriff „./Robi - mnc stan“ ein und Stan führt Eure Anweisung aus. Achtet **vorher** darauf, dass Ihr dran seid und Stan genügend Platz hat.

### Ein paar Hinweise:

- *double* ist eine Dezimalzahl
- *int* ist eine ganze Zahl
- Am Ende jeder Anweisung muss ein Strichpunkt stehen.
- Mit *cout<< „ das was hier steht, wird dann ausgegeben “<<endl;* kann man eine Ausgabe auf den Bildschirm machen.

# Workshop Einführung in die Robotik

## Mini Roboter

Betreut von **Bahram Kord** und **Markus Kächele**

### Einführung

Wir bieten euch hier die Möglichkeit, ein paar erste Schritte mit mobilen Tisch-Robotern zu unternehmen. Diese ungefähr faustgroßen Roboter haben den Vorteil, dass sie wenig Platz einnehmen, aber dennoch über nahezu alle gängigen Sensoren verfügen. Selbst eine Kamera ist mit an Bord. Zudem ist die Programmierung denkbar einfach. Für die Aufgaben haben wir euch ein paar Funktionen vorbereitet:



- **bewegen(*Distanz*)**: Lässt den Roboter um die angegebene *Distanz* in **Zentimetern** nach vorne fahren. Wenn eine negative Zahl eingegeben wird, fährt er rückwärts.
- **drehen(*Gradzahl*)**: Veranlasst den Roboter, sich um eine angegebene *Gradzahl* zu drehen.

Nachdem die Verbindung zum Roboter hergestellt wurde, können die Befehle einfach eingegeben werden.

## Aufgaben

1. **Vor und zurück fahren**: Zu Beginn etwas ganz einfaches. Fahrt den Roboter eine bestimmte Strecke vor, dreht um und fährt dieselbe Strecke zurück.
2. **Ein Quadrat fahren**: Versucht das auf dem Blatt aufgemalte Quadrat nachzufahren.
3. **Ein Dreieck fahren**: Versucht das auf dem Blatt aufgemalte Dreieck nachzufahren. Hinweis: Ein Geodreieck könnte hilfreich sein...