

### Aufgaben 7 Vererbung



1. Sie entwerfen eine Software für selbst fahrende Roboterplattformen. Damit der Code für verschiedene Arten von Robotern universell einsetzbar ist, soll es zunächst eine allgemeine Basisklasse **Roboter** geben mit den folgenden **protected** Attributen:

- eine *ID* Nummer zur Identifizierung
- *fehlerstatus* als boolescher Wert (gestoppt / arbeitet normal)
- *position* als reelle Zahl (gefahrte Strecke in Metern)

Ergänzen Sie die Klasse auch um einige Methoden:

- Konstruktor zur Initialisierung der Attribute (ID als Parameter übergeben)
- Get-Methoden für die Attribute
- Eine *status*-Methode (Ausgabe der ID, Position des Roboters und ob alles normal funktioniert oder er gestoppt wurde)

Für unsere Roboter benötigen wir zusätzlich **Laserscanner**, die als eigene Klasse definiert werden:

- privates Attribut *abstand* (Abstand zum nächsten Hindernis)
- Konstruktor und get-Methode
- Eine Methode *scan*, die einen zufälligen Wert zwischen 0.0 und 25.0 in *abstand* speichert
- **Tipp:** `rand() / double(RAND_MAX)` liefert einen Wert von 0.0 bis 1.0

Leiten Sie von der Basisklasse **Roboter** eine spezialisierte Klasse **Husky** ab. Husky ist eine Roboterplattform mit 4 Rädern (nicht zu implementieren) und erhält außerdem einen Laserscanner. Eine Methode *fahren* bewegt den Roboter um einen angegebenen Streckenabschnitt vorwärts falls der Abstand zum nächsten Hindernis groß genug ist. Anderenfalls ändert sich der Fehlerstatus. Der Streckenabschnitt ist als Parameter an die Funktion zu übergeben.

In der main-Funktion erstellen Sie ein Objekt der Klasse **Husky** und lassen dieses solange 0,5 m vorwärtsfahren, bis ein Hindernis erreicht wurde. Zeigen Sie regelmäßig den Status auf dem Bildschirm an.

2. (Zusatz) Zu erstellen ist ein C++-Programm, das den Wissenschaftlern ermöglicht, die Merkmale einer infolge der Technoevolution entstandenen Spezi „Helmtiger“ im Vergleich zum bekannten Tiger zu untersuchen.

Erstellen Sie dazu die Klasse **Tiger**, die die Angaben zu

- Registriernummer,
- Schädelfestigkeit (ganze Zahl)

enthält.

Erstellen Sie für die Klasse den geeigneten Konstruktor mit zwei Parameter, get-Funktionen und die Funktion für die Ausgabe des Inhaltes der Objekte.

Die ebenfalls zu erstellende Klasse **Helm** erhält

- die Farbe (std::string) und ebenfalls
- die Festigkeit.

Erstellen Sie für die Klasse einen Konstruktor. Ermöglichen Sie außerdem den lesenden Zugriff auf die Datenfelder mit Hilfe der get-Methoden.

Die Klasse **Helmtiger** ist von der Klasse Tiger abzuleiten.

In der Klasse ist

- ein weiteres Datenfeld als Zeiger auf Objekt der Klasse Helm

zu definieren.

Zum Zugriff auf das Datenfeld „helm“ von außerhalb der Klasse wird eine set-Funktion benötigt, die ein nachträgliches Zuweisen des Helms ermöglicht.

Eine weitere Methode der Klasse **Helmtiger** soll die Schädelfestigkeit des „behelmteten“ Tigers als die Summe der Festigkeiten des Tigers und des Helms berechnen und zurückgeben.

In der main-Funktion ist zum Testen der Klassen jeweils ein Objekt der Klassen Tiger, Helm und Helmtiger zu erstellen. Geben Sie die Daten der Objekte aus und, wo es möglich ist, die Gesamtfestigkeit.