

## Aufgabe 1 (Rennwagen):

### TEAMARBEIT:

#### 1. Die Klassendefinition. Die fett gedruckten Bezeichnungen sind einzuhalten!

Objekte vom Typ **Rennwagen** haben eine Eigenschaft **Geschwindigkeit** (zu speichern als Ganzzahl für die Einheit „km/h“). Zu Beginn hat ein Rennwagen natürlich ein Tempo von 0 km/h. Man kann einen Rennwagen **beschleunigen**, dabei ist mit anzugeben, um wieviel km/h der Rennwagen seine Geschwindigkeit erhöhen soll. Man kann einen Rennwagen **bremsen**, dabei sinkt die Geschwindigkeit automatisch um genau 50 km/h bis auf minimal 0 km/h (Keine negativen Geschwindigkeiten!)

Die Eigenschaft des Rennwagens kann man mit zwei Methoden abfragen:

- **Tempo ( )** liefert die aktuelle Geschwindigkeit als Zahl,
- **coutTempo ( )** zeigt einen ganzen einzeiligen Satz auf dem Bildschirm an, der aussagt, ob der Rennwagen steht oder mit welcher Geschwindigkeit er gerade fährt.

Zeichnen Sie zunächst die UML-Klassendefinition.

Programmieren und speichern Sie Ihre Klassendefinition OHNE die übrigen üblichen Präprozessor-Anweisungen in eine Headerdatei namens **Rennwagen.h**

Binden Sie diese Headerdatei in Ihr nun folgendes Hauptprogramm ein:

#### 2. Das Hauptprogramm (Testprogramm, Szenario)

Erzeugen Sie 4 **Rennwagen** auf unterschiedliche Weise:

- Als statisches Objekt (z.B. „RedBull“)
- Als dynamisches Objekt (z.B. „McLaren“)
- Als Array für 2 Rennwagen (z.B. Formel1[2]). Hier erzeugen Sie sogleich für beide Array-Elemente einen neuen Rennwagen.

**beschleunigen** Sie den RedBull auf 200 km/h, **bremsen** 3 mal ab und lassen sich die Geschwindigkeit als kompletten Satz anzeigen.

Beschleunigen Sie den McLaren auf 230 km/h, bremsen 3 mal ab und fragen das aktuelle Tempo ab. Geben Sie einen selbst gewählten Satz unter Verwendung des Tempowertes am Bildschirm aus. Beschleunigen Sie den McLaren um 60 km/h und wiederholen die Bildschirmausgabe wie eben beschrieben.

Beschleunigen Sie den ersten Formel1-Array-Rennwagen auf 180 km/h, bremsen 4 mal ab und zeigen das Tempo auf zwei Arten an: einmal mit **Tempo ( )** und einmal mit **coutTempo ( )**

Wählen Sie ein beliebiges Szenario für den zweiten Formel1-Array-Rennwagen.

#### 3. Tauschen Sie Ihre .h- und .cpp-Dateien aus und binden sie diese in Ihr eigenes Projekt ein. Testen Sie, ob alles fehlerfrei läuft...

## Aufgabe 2 (Rennwagen-Erweiterung):

Ein Rennwagen bekommt jetzt 2 zusätzliche Eigenschaften:

Einen **Name** (z.B. die Bezeichnungen RedBull, McLaren, Lotus oder Petronas). Dieser **Name** soll beim Erzeugen eines Rennwagens mit übergeben ODER vom Benutzer abgefragt werden. (Damit bleibt das Hauptprogramm kompatibel. Testen Sie es!)  
Der Name wird nur von den geeigneten klasseneigenen Methoden für sinnvolle Bildschirmausgaben verwendet (z.B. „RedBull steht!“ oder „McLaren faehrt mit 120 km/h“).

Einen Zustand „**MotorLaeuft**“. Dazu muss es natürlich zwei Methoden geben:

- **MotorStarten( )** bewirkt, dass der Motor läuft,
- **MotorStoppen( )** bewirkt, dass der Motor abgeschaltet wird.

Korrigieren Sie Ihre bisherigen Methoden dahingehend, dass nur logisch einwandfreie Zustände entstehen:

- Man kann einen Rennwagen erst beschleunigen, wenn der Motor läuft.
- Umgekehrt sollten Sie den Motor nicht ausschalten, solange der Rennwagen noch fährt.

Gestalten Sie Ihre Methoden so, dass das bisherige Test-Szenario kompatibel bleibt:

- Im bisherigen Testablauf wird der Motor nicht gestartet. Wir bekämen immer nur Warnhinweise der Art: „Lotus kann nicht beschleunigt werden, da der Motor aus ist.“ Lassen Sie in diesem Fall das Objekt nachfragen, ob der Motor gestartet werden soll.

Bauen Sie in Ihrem Szenario für den zweiten Formel1-Array-Rennwagen jedoch den Aufruf **MotorStarten( )** ein und versuchen Sie, nach dem ersten Beschleunigen den Motor mit **MotorStoppen( )** abzuschalten.

Verhält sich Ihr Programm korrekt?

## Aufgabe 3 (Kompatibilitätstest):

Tauschen Sie abermals Ihre .h- und .cpp-Dateien aus.

Läuft Ihr neu zusammen gesetztes Projekt immer noch?

## Aufgabe 4 (Zusatz, optional):

Definieren Sie einen Aufzählungstyp **Tempo** mit den Elementen **minimal**, **wenig**, **moderat**, **viel** und **maximal**. Sie können der Methode **beschleunigen(...)** einen dieser Werte übergeben, um den Rennwagen um 30, 75, 110, 150 oder 180 km/h zu beschleunigen. Numerische Werte sind als Parameter aber weiterhin erlaubt. Wird gar nichts übergeben, gilt eine moderate Beschleunigung um 110 km/h.

Das gleiche Spiel können Sie für die Methode **bremsen(...)** vornehmen, z.B. **stark**, **sacht**, **voll** für ein Abbremsen um 100 oder 50 km/h oder eine Vollbremsung auf 0 km/h.

Außerdem könnten Sie die Höchstgeschwindigkeit der Rennwagen festlegen...