Übung: Foerderband Queue C#

Schreibe ein Programm zur Überwachung eines Förderbandes in der Automobilproduktion. Auf dem Förderband werden Fahrzeuge platziert. Dazu brauchst Du folgendes:

Fahrzeuge **haben** folgende Eigenschaften:

- Eine Farbe (z.B. „Rot“)
- Einen Typ (z.B. „BMW“)
- Ein Gewicht (in kg)
- Eine laufende Nummer (damit sie auf dem Förderband identifiziert werden können)
- Ein Reifenobjekt

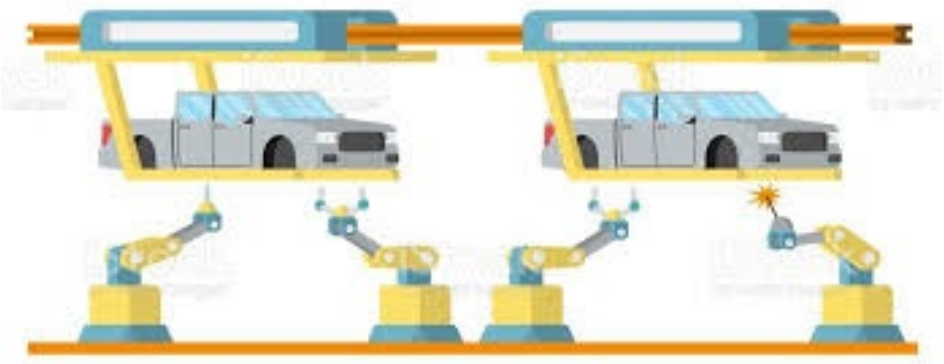
Beim Erzeugen sollen Farbe, Typ und Gewicht mitgegeben werden können. Die laufende Nummer wird beim Platzieren des Fahrzeuges auf das Förderband vergeben (Hinweis: dazu braucht es einen „Setter“ für die Nummer).
Das Reifenobjekt soll mit NULL initialisiert werden (d.h. Fahrzeug ohne Reifen). Der Einfachheit halber verwenden wir nur ein Reifenobjekt (obwohl ein Auto normalerweise 4 Reifen hat, aber diese sind für gewöhnlich ja alle gleich).

Ein solcher Reifen **hat** folgende Eigenschaften:

- Die Reifenbreite in mm (z.B. 225 mm)
- Das Verhältnis der Höhe zur Breite des Reifens in Prozent (z.B. 55 %)
- Die Kennzeichnung der Bauart (z.B. „R“ für Radialreifen)
- Den Felgendurchmesser in Zoll (z.B. 16 Zoll)

(das Beispiel beschreibt damit den Reifen: **225/55 R 16**)

Die Fahrzeuge werden am Anfang auf ein Förderband gehievt und dort weitertransportiert. Am Ende werden sie von dort wieder entnommen.



Dazu wollen wir eine Warteschlange implementieren. Diese Warteschlange (oder Queue) soll als Erweiterung der verketteten Liste programmiert werden, die wir schon gemacht haben. Dazu brauchen nur zwei neue Operationen zur verketteten Liste hinzugefügt werden.

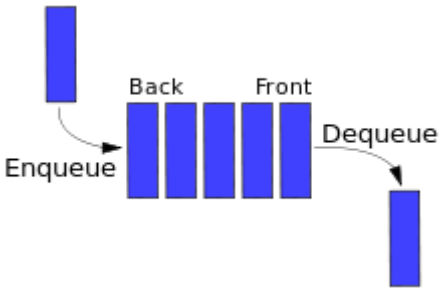
Eine **Warteschlange** (oder Queue) ist eine spezielle Datenstruktur, mit der beliebige Daten verwaltet werden können.

Auf einer Warteschlange sind zwei Operationen definiert:

dequeue
: Einen Wert aus der Warteschlange entfernen.

enqueue
: Einen Wert in die Warteschlange einfügen.

Daten werden immer am Ende eingefügt und am Anfang entnommen, daher ist die Warteschlange ein sogenannter FIFO-Speicher (First In First Out). Dies ist in der nebenstehenden Abbildung dargestellt.



Die Fahrzeuge werden in dieser Queue gespeichert, die unser Förderband darstellt.

Das Foerderband soll folgende Funktionalität bieten:

- Fahrzeuge zum Foerderband hinzufügen (mittels der „enqueue“-Methode). Hier wird die Nummer an das Fahrzeug vergeben.
- Fahrzeuge vom Foerderband entnehmen (mittels der „dequeue“-Methode.
- Eine Fahrzeugliste aller Fahrzeuge auf dem Förderband ausgeben (mittels einer „print“-Methode).
  Die Ausgabe soll z.B. so aussehen (Nummer, Farbe, Typ, Gewicht):

```
1       Rot     Alpha   1500
2       Blau    BMW     2500
3       Gruen   Skoda   2000
```

- Eine Methode „addReifenTo(…)“, mit der man dem Fahrzeug Nummer x ein Reifenobjekt hinzufügen kann. (Hinweis: Das Reifenobjekt muss dabei vorher erzeugt werden).
- Eine Methode „allWithReifenBauart(…)“, der man eine Bauart (z.B. „R“ für Radialreifen) mitgeben kann und die eine Fahrzeugliste ausgibt, wo Reifen dieser Bauart montiert sind.
  Die Ausgabe soll z.B. so aussehen (Nummer, Farbe, Typ, Gewicht, Reifen-Eigenschaften):

```
2       Blau    BMW     2500            Reifen: 225/55 R 16
```

(Die Reifeneigenschaften sind dabei kodiert als:
„Breite/Verhältnis Bauart Felgendurchmesser“)

Allgemeine Anforderungen an das Programm:

- Erzeuge für jede Klasse auch einen Default-Konstruktor.

Hinweise:

- Implementiere eine Basisklasse zum Einlesen der Informationen die zum Erzeugen von Auto-Instanzen benötigt werden. Diese Klasse soll nur eine leere Methoden „ReadAutodaten()“ beinhalten, welche ein leeres Array mit Autos zurückgibt. In einer davon abgeleiteten Klasse soll der Benutzer die Eingaben tätigen können, überschreibe dort die Methode der „leeren Klasse“.
  Hintergrund: wir wollen später gemeinsam eine weitere Klasse zum Einlesen aus einer Datei hinzufügen -> damit wollen wir Exception-Handling lernen.
- Verwende diese in der main()-Methode um die entsprechenden Objekte zum Förderband hinzuzufügen und teste damit dein Programm.

Zusatzaufgabe:

Erweitere die Warteschlange um folgende Funktionen:

NumberElements: Gibt die Anzahl der Elemente die sich in der Warteschlange befinden zurück.

GetObjectAtPosition: Gibt das Objekt das sich an einer bestimmten Position in der Warteschlange befindet zurück. Die Positionsangabe ist ein Wert zwischen 0 und AnzahlElemente-1. WICHTIG!! Es wird nur der Zugriff auf das Objekt in der Queue ermöglicht. Das Objekt selbst wird nicht!! Aus der Queue entfernt.

Hilfestellung zu Warteschlangen in C#:

https://masterdotnet.com/csharp/ds/queueincsharp/