

PUC 3CDH	2003/2004/ma Mai 2004	3. Leistungsüberprüfung in C#	Name:
-------------	--------------------------	----------------------------------	-------

Max: 48 Punkte

Punkte: _____ Bewertung: _____

Geschäftsfall: Dienstreisen eines Direktors

In einem Konzern sind genau 5 Direktoren angestellt. Es sind dies:

Direktor: Huber,	Abteilung: EDVO,	Dienstnummer: 124
Direktor: Moser,	Abteilung: Math,	Dienstnummer: 111
Direktor: Mayer,	Abteilung: Motor,	Dienstnummer: 171
Direktor: Leber,	Abteilung: Auto,	Dienstnummer: 172
Direktor: Rucky,	Abteilung: LKW,	Dienstnummer: 133

Jene Direktoren mit gerader Dienstnummer haben einen Dienstwagen mit „Sportausführung“, alle anderen haben einen normalen Wagen.

Zu Beginn sollen alle Namen der Direktoren ausgegeben werden.

Dann soll ein Direktor eine Dienstreise antreten (hier Direktor Huber). Er steigt dabei genau 10 Mal auf das Gas. Dabei wird jedes Mal die Geschwindigkeit ausgegeben. Fährt er schneller als 100 km/h, ist eine Meldung auszugeben: ... (siehe im folgenden Bild).

```

F:\C#\Übungsteil 200304\Tests\BisZuCollections\Einfuehrung01\bin\Debug\Einfu...
Folgende Direktoren sind in dem Konzern?

Direktor: Huber
Direktor: Moser
Direktor: Mayer
Direktor: Leber
Direktor: Rucky

Name des Direktors, der auf die Reise geht: Huber

Herr Direktor fährt: 5,307472
Herr Direktor fährt: 21,30031
Herr Direktor fährt: 55,06642
Herr Direktor fährt: 65,72842
Herr Direktor fährt: 72,90839
Herr Direktor fährt: 83,00461

Eine Ausnahme vom Typ Einfuehrung01.Ausnahme wurde ausgelöst.
Herr Direktor Huber ... Sie fahren zu schnell

Eine Ausnahme vom Typ Einfuehrung01.Ausnahme wurde ausgelöst.
Herr Direktor Huber ... Sie fahren zu schnell

```

Klassen

Class1 (Console): beschreibt die Abfolge der Berechnungen.

- `Main()`: Ausgabe aller Direktoren über die Collectionklasse `Konzern` – Eingabe eines Direktorennamens (hier Huber) – Dieser Direktor geht auf die Reise `reiseDirektor(name)`.

Konzern: ist eine Collectionklasse für die 5 Direktoren. Erstellt 5 Direktoren (s.o).

Eigenschaften: `mdirektoren` in einer `ArrayList`;

- `reiseDirektor(name: string): void::` „vergleicht den eingegebenen Namen mit den Namen der Direktoren und schickt diesen auf die Reise `onGibGas(name)`“.

Direktor: beschreibt den Direktor (siehe Klassendiagramm).

Eigenschaften: siehe Konstruktor

- `Direktor(...):` Konstruktor: setzt die Attribute und weist ihm je nach Dienstnummer ein Auto zu. Gerade Dienstnummer = Dienstwagen mit SportAusführung, ungerade Dienstnummer = (normaler) Wagen. Die `PS` liegen zwischen einem Wert von 100 und 200.
- `onGibGas(name: string): void::` Hier wird genau 10 Mal Gas mit dem Auto gefahren `fahren()`, wobei sich die Parameter `Zeit: dt` und `Beschleunigung: db` zufällig

zwischen den reellen Werten 10 für dt und 5 für db bewegen. Fährt das Auto schneller als 100 km/h, wird eine Ausnahme geworfen (siehe Bild oben).

Wagen: beschreibt das normale Auto.

Eigenschaften: siehe Klassendiagramm. kmH wird zu Beginn = 0 gesetzt.

- `Wagen(ps: int): Konstruktor`
- `fahren(...): float::` Berechnet die Geschwindigkeit $kmH += dt * db$. Testet, ob die Geschwindigkeit nicht über 100 km/h liegt (Ausnahme!).

Dienstwagen: beschreibt das Auto mit Sportausführung. Von `Wagen` abgeleitet! Diese Klasse implementiert das Interface `ISportAusfuehrung!`

Eigenschaften: siehe Klassendiagramm.

- `Dienstwagen(ps: int; dienstNr: int): Konstruktor`
- `fahren(...): float::` Berechnet die Geschwindigkeit $kmH += (dt * db) * kmHFaktor(ps)$. Testet, ob die Geschwindigkeit nicht über 100 km/h liegt (Ausnahme!).

ISportAusfuehrung: Inerface, stellt die Methode `vkMHfaktor(ps)` (s.o.) zur Verfügung.

Ausnahme: Eine Exceptionklasse. Soll folgende Meldung wie oben angegeben aus!

Eigenschaften: `name: string;`

- `Ausnahme(name: string): Konstruktor.`

Abzugeben: Die Listings der cs.-Dateien!

