

# Übungsbeispiel

Erstellen Sie ein Visual Studio Projekt für eine **.NET Core 3.1 Konsolen-Applikation**.

Versuchen Sie die folgenden Anforderungen umzusetzen:

Ein Versicherungs-Unternehmen möchte die Polizen zukünftig elektronisch bearbeiten.

## Entitäten

Ein gültiger Versicherungsvertrag wird in Form einer Polize dargestellt. Es werden die folgenden Versicherungsformen unterstützt: **Lebensversicherung, Unfallversicherung, KFZ-Versicherung**.

Jede Polize, egal ob eine Polize für die Lebensversicherung, Unfallversicherung oder KFZ-Versicherung, hat eine **Polizzen-Nummer**, eine **Bezeichnung** und den **Namen (Vor- und Zuname) des Vertragsinhabers**, einen jährlichen Prämienbetrag und einen kumulierten **Schadensbetrag**, der pro Schaden um den Schadenswert erhöht wird.

Für eine Polize **muss** bei der Erstellung eine Polizzen-Nummer, Bezeichnung und einen Vertragsinhaber definiert werden.

**Von einer Polize selbst, kann keine Objekt-Instanz erstellt werden!!!**

**Von den Versicherungsformen können keine weiteren Subklassen mehr abgeleitet werden!!!**

Eine Lebensversicherung **muss** noch zusätzlich der **Namen des Bezugsberechtigten** angegeben werden, die im Falle des Ablebens des Versicherungsnehmers die Versicherungs-Summe ausbezahlt bekommt. Eine Lebensversicherung **ohne** Bezugsberechtigten kann **nicht erstellt** werden.

Bei einer KFZ-Versicherung kann die **Bonusstufe von 0 – 17** erfasst werden. In der Regel ist bei einer Neuversicherung die **Bonusstufe auf 9** eingestellt.

Bei einer Unfallversicherung kann ein **Betrag bei Todesfall** und der **Namen des Bezugsberechtigten** angegeben werden, die Falle des Ablebens des Versicherungsnehmers durch einen Unfall diesen Betrag erhält, erfasst werden.

Alle Versicherungsformen haben ein **Versicherungsobjekt**, das versichert wird. Jedes Versicherungsobjekt hat einen eindeutigen Identifier (**ID, Guid**), eine **Bezeichnung** und einen **Versicherungswert**.

Für ein Versicherungsobjekt **muss** bei der Erstellung eine **ID, Bezeichnung** und einen **Versicherungswert** festgelegt werden.

## Schadensfall

Im Falle eines Schadens wird der Schadensbetrag erhöht. Übersteigt der Schadensbetrag die Versicherungs-Summe, dann muss die benutzerdefinierte **Exception SchadenUebersteigtVersicherungsSummeException** ausgegeben werden.

## Aufgaben des Back-Office

Die Prämie wird vom **Back-Office** abhängig von der Versicherungsform wie folgt berechnet:

Lebensversicherung: **1,3 %** jährlich vom **Versicherungswert - EUR 20,00 Gewinnbeteiligung**

Kfz-Versicherung: **3,3 %** jährlich vom **Versicherungswert + EUR 50,00 Softhilfzuschlag**

Unfall-Versicherung: **0,2 %** jährlich vom **Versicherungswert**

Der Schadensbetrag einer Polizza wird vom **Back-Office** regelmäßig überprüft. Sollte der Schadensbetrag **85%** des Versicherungswertes erreichen, dann wird der Versicherungsvertrag gekündigt.

### Ziel:

Die Übung wurde vollständig umgesetzt, wenn die Methoden der **Vererbung**, **Kapselung**, **Fehlerbehandlung (Exceptions)** und **Interfaces** eingesetzt wurden.

Testen Sie Ihre Programmierung für eine Versicherungsform in der Program.cs.  
Lösen Sie eine Exception aus und fangen Sie die **SchadenUebersteigtVersicherungsSummeException** ab.