

# Übung

## 1 Aufgabe

In dieser Übung soll die Klassen-Bibliothek um ein Benutzersteuerelement zur Definition eines Zeitraums erweitert werden. Das Steuerelement soll über eine neu zu erstellende Klasse „Zeitraum“ bedient werden, die als erstes implementiert werden soll.

## 2 Erstellen einer Klasse

Die Klasse Zeitraum soll die Möglichkeit der Abbildung eines Zeitraums „bis zum“, „ab dem“ oder „von bis“ erlauben. Dabei sollen die Abfrage (und das Setzen) eines Startdatums und des Enddatums als DateTime-Objekt möglich sein. Ist der Typ „bis zum“ gewählt, so wird als Startdatum das kleinste abzubildende Datum des Datentyps DateTime zurückgegeben, bei „ab dem“ als Enddatum das größte abzubildende Datum. Beim Setzen von Start- bzw. Enddatum soll der Typ des Zeitraums ggfs. den gegebenen Werten angepaßt werden. Genauso soll beim Setzen des Typs das entsprechende Start- oder Enddatum dem Typ entsprechend verändert werden. Dabei soll über die Eigenschaft DateOnly einstellbar sein, ob die DateTime-Objekte der Zeitraum-Grenzen immer nur den Date-Anteil oder auch die Zeiten abbilden sollen.

Die Klasse Zeitraum soll eine Funktion „LiegtDatumInZeitraum“ implementieren, die den Übergabeparameter Eingabedatum daraufhin prüft, ob er innerhalb dem durch das Objekt definierten Zeitraum liegt und true (liegt drin) bzw. false zurück geben. Beachten Sie in diesem Fall die Uhrzeit, so dass ein zu prüfendes Datum „02.03. 14:00“ im Zeitraum „01.03.“ bis „02.03.“ liegt.

Die Eigenschaften der Zeitraum-Objekte sollen von außen les- und schreibbar sein.

Darüber hinaus soll das Zeitraum-Objekt über eine toTimeSpan-Funktion verfügen, der das TimeSpan-Objekt als out-Referenz übergeben wird und die true bzw. false zurückgeben soll. False genau dann, wenn der Zeitraum offene Enden („bis zum“, „ab dem“) hat, true dann, wenn „von bis“ als Typ eingestellt wurde. Das Referenz-Objekt wird entweder mit dem korrekten Rückgabe-TimeSpan oder einem leeren TimeSpan gefüllt.

Bevor Sie mit der Implementierung beginnen, zeichnen Sie bitte ein Klassendiagramm, mit dem die beschriebenen Anforderungen abgebildet werden können. Zeichnen Sie bitte auch ein Aktivitäten-Diagramm für den Ablauf beim Setzen des Startdatums, Enddatums und des Typs des Zeitraums, so dass die Daten konsistent gehalten werden.

### 3 Benutzersteuerelement

Wenn die Klasse Zeitraum implementiert wurde und sauber funktioniert (bitte mit einem Testprojekt prüfen), soll ein Benutzersteuerelement hinzugefügt werden, mit dem die Definition des Zeitraums in der GUI möglich wird. Dabei soll für den Designer und/oder ein das Steuerelement nutzendes Programm die Möglichkeit gegeben sein, das Steuerelement auf einen bestimmten Zeitraum voreinzustellen. Hierbei soll die Uhrzeit zunächst unberücksichtigt bleiben und nur der Date-Teil verwendet werden.

Konzipieren Sie ein geeignetes Layout des Steuerelementes und implementieren Sie die oben beschriebene Funktionalität für das Steuerelement.

Stellen Sie sicher, dass die Eigenschaften des Steuerelementes im Designer-Modus einstellbar sind und funktionieren. Testen Sie darüber hinaus die Funktionsfähigkeit des Steuerelementes in einem Testprojekt.

### 4 Mögliche Erweiterungen

Erweitern Sie das Steuerelement um die Möglichkeit, über die Eigenschaft „ZeigeUhrzeit“ auch Zeiten einstellen zu können. Realisieren Sie dies entweder über ein weiteres Steuerelement, das auf dem Erstellten basiert oder machen Sie die Zeit-Erweiterung über eine boole'sche Eigenschaft ein- bzw. ausschaltbar.

Erweitern Sie das Steuerelement um die Möglichkeit, eine Zeitspanne vorgeben zu können. Dabei soll nach Eingabe des ersten Datums das jeweils zweite Datum automatisch berechnet und gesetzt werden. Über die Eigenschaft „FesteZeitspanne“ vom Typ bool soll einstellbar sein, ob die Zeitspanne in jedem Fall die gesetzte Länge einhält oder nach dem ersten Setzen mit Einhalten der Zeitspanne durch den Benutzer durch Setzen anderer Start- und Enddaten überschreibbar ist.