

☐ Gr. 1, DI (FH) G. Horn, MSc☐ Gr. 2, J.-P. Haslinger, MSc

Name _____ Aufwand in h _____

Punkte _____ Kurzzeichen Tutor / Übungsleiter _____ / _____

1. Zeitspannen reloaded**(2 + 3 Punkte)**

In der letzten Übung haben Sie bereits einen Algorithmus zum Konvertieren von Zeitspannen in Sekunden und umgekehrt entwickelt. Mit Ihrem seit dem neu erworbenen Wissen über Datentypen

- Entwerfen Sie einen eigenen Datentyp `TimeSpan` und implementieren Sie die beiden Algorithmen `TimeSpanToSeconds` und `SecondsToTimeSpan` erneut unter Verwendung dieses Datentyps. Dokumentieren und kommentieren Sie die Änderungen in den Schnittstellen.
- Implementieren Sie einen zusätzlichen Algorithmus `TimeDifference`, der den Zeitunterschied zwischen zwei gegebenen Zeitspannen in Sekunden ermittelt.

2. Zahlen aufsummieren – jetzt mit offenen Feldern**(5 Punkte)**

In der ersten Übung haben Sie einen Algorithmus zur Bildung zweier Summen über eine Zahlenreihe kennengelernt und damals noch "mit beschränkten Mitteln" in Pascal implementiert.

Verbessern Sie nun Ihre damalige Lösung unter Einsatz der von Ihnen in der Zwischenzeit erworbenen Kenntnisse, indem Sie eine Prozedur `ComputeSums` entwickeln, die die zu summierende Zahlenfolge über einen offenen Feldparameter übergeben bekommt. Von der Logik soll sich sonst nichts verändern, d. h. summiert wird bis zur ersten Null im Feld etc.

Testen Sie Ihre Prozedur mit den verschiedensten Feldern (Größe, Indizes...) und zeigen Sie auch, welche (neuen?) Sonderfälle entstehen und dass diese richtig behandelt werden.

3. Römische Zahlen**(4 + 3 + 7 Punkte)**

- Implementieren Sie eine Funktion `IntToRoman`, die eine Dezimalzahl in eine römische Zahl (Zeichenkette) konvertiert. Dabei soll zunächst nur die einfache Umrechnung¹ zum Einsatz kommen.

Beispiel: $49 = 10+10+10+10+5+1+1+1+1 = XXXXVIII$

- Implementieren Sie im Anschluss eine zweite Funktion `IntToRoman2`, die beim Umwandeln auch die Subtraktionsregel¹ beherzigt.

Beispiel: $49 = (50 \text{ weniger } 10) + (10 \text{ weniger } 1) = XLIX$

- Implementieren Sie abschließend auch das Gegenstück `RomanToInt`, das eine als Zeichenkette gegebene römische Zahl in ihre entsprechende Dezimalzahl verwandelt. Es sind dabei beide Zahlenformate (mit und ohne Subtraktionsregel) zu unterstützen. Stellen Sie außerdem sicher, dass Ihre Implementierung auch mit ungültigen Zeichenfolgen ohne Probleme umgehen kann und z. B. auch die Fehlerposition zurückliefert.

Testen Sie alle Ihre Implementierungen ausführlich, besonders auch mit großen Zahlen sowie Sonder- und Grenzfällen.

¹ <http://www.roemische-zahlen.net/>