

Aufgabe

Es soll ein Interpreter für einfache arithmetische Ausdrücke geschrieben werden. Übergeben wird zunächst der Ausdruck, der mehrere Variablen enthalten kann. Danach soll der Interpreter den korrekten Wert für den Ausdruck berechnen, wenn für jede Variable ein Wert übergeben wird.

Implementiere die Klasse *ExpressionInterpreter* (siehe Codesnippet unten) so, dass nach Aufruf von *ExpressionInterpreter(string expressionAsText)* und *int CalculateWith(IDictionary valuesOfVariables)* der in 'expressionAsText' beschriebene Ausdruck mit den in 'valuesOfVariables' gegebenen Werten korrekt berechnet wird.

Es dürfen nach Belieben weitere Klassen, Methoden usw. ergänzt werden.

Der resultierende Code soll Produktionsqualität haben.

Für die Präsentation lassen wir freie Hand. Im persönlichen Termin erhältst Du ein Zeitfenster von ca. 15min

Codesnippet

```
using System.Collections.Generic;

namespace worksample
{
    class ExpressionInterpreter
    {
        public class Variable
        {
            public Variable(char name)
            {
                this.name = name;
            }

            private char name;
        }

        public ExpressionInterpreter(string expressionAsText)
        {
            // ...
        }

        public int CalculateWith(Dictionary<Variable, int> valuesOfVariables)
        {
            // ...
        }
    }
}
```

Beispiel

Die folgende Methode

```
// ...  
int example()  
{  
    var interpreter = new ExpressionInterpreter("3*x + 20- y *(z + 17)");  
    var values = new Dictionary<Variable, int>();  
  
    values.Add(new Variable('x'), 1);  
    values.Add(new Variable('y'), 2);  
    values.Add(new Variable('z'), 3);  
  
    int result = interpreter.CalculateWith(values);  
    return result;  
}
```

sollte den Wert -17 zurückliefern.

Randbedingungen

- Die übergebenen Ausdrücke bestehen ausschließlich aus folgenden Bestandteilen:
 - Variablennamen, bestehend aus genau einem kleinen Buchstaben innerhalb a-z,
 - den binären Operatoren '+', '-' und '*' mit ihren üblichen Bedeutungen und Regeln,
 - Natürlichen Zahlen, inklusive 0, es gibt keine Vorzeichen,
 - den Klammern '(' und ')',
 - Leerzeichen, also ASCII 32.
- Die Werte der Variablen können beliebige ganze Zahlen sein (aber siehe nächste Bedingung).
- Das Ergebnis des Ausdrucks oder einzelner Terme wird den Wertebereich von int nicht überschreiten.
- Es darf davon ausgegangen werden, dass die an *CalculateWith()* übergebenen Parameter korrekt sind. Insbesondere werden die 'valuesOfVariables' genau zu den in der Formel verwendeten Variablen passen.

Viel Erfolg!