

CSharp/OOP/Ereignis

Lehrziele

- Erstellen von einfachen Klassen
- Wie ein Delegat definiert wird
- Wie ein Ereignis definiert wird
- Wie eine Observable-Klasse erstellt wird
- Wie eine Observer-Klasse erstellt wird
- Wie die Objekte Observable und Observer verbunden werden

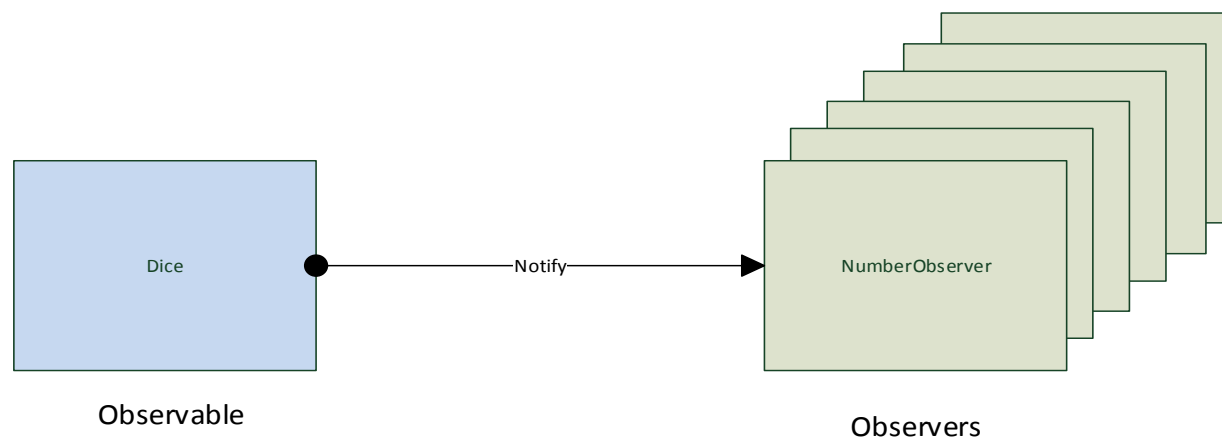
Aufgabenstellung

Dice-Simulation

Es soll eine einfache Würfel-Simulation erstellt werden. Dabei wird ein Würfel (Dice) beliebig oft geworfen und die geworfenen Augenzahlen notiert. Je öfter der Würfel geworfen wird, desto mehr sollte sich die Verteilung der Augenzahlen einer Gleichverteilung nähern ($n \rightarrow \infty$). Wenn sich die Verteilung der Augenzahlen nicht der Gleichverteilung nähert, dann kann davon ausgegangen werden, dass der Würfel manipuliert wurde.

Design der Lösung

Die Struktur der Lösung sollte als Observer-Pattern implementiert werden. Dabei stellt der Würfel das Observable-Objekt dar und verständigt nach jedem Wurf alle registrierten Observer-Objekte. Diese Observer-Objekte notieren die Anzahl der geworfenen Augenzahl und berechnen das Verhältnis zu den gesamt Würfeln. In der folgenden Abbildung soll die Lösungsstruktur skizziert werden:



Klasse *Dice*

Diese Klasse soll einen Würfel darstellen und simulieren.

Eigenschaften:

public	int Count	Ruft die Anzahl von Wuerfen ab. (ReadOnly)	
public	int LastNumber	Ruft den letzten Wurf ab. (ReadOnly)	
public	DiceHandler event Observers	Ereignis zum Registrieren und Abmelden von Beobachtern.	

Methoden/Konstruktor:

public	Dice()	Initialisiert eine neue Instanz der Dice-Klasse.	
--------	--------	--	--

Methoden:

public	void Roll()	Erzeugt einen neuen Wurf und verständigt die registrierten Beobachter.	
public	void Reset()	Setzt alle Werte wieder auf den Anfangszustand zurück.	
protected	void NotifyObservers()	Diese Methode verständigt alle Abonennten.	

Klasse *NumberObserver*

Diese Klasse dient als Beobachter der Klasse Dice.

Eigenschaften:

public	int Count	Ruft die Anzahl von Wuerfen ab. (ReadOnly)	
public	int WatchCount	Ruft die Anzahl der beobachteten Zahl ab. (ReadOnly)	
public	int WatchNumber	Ruft die Zahl ab, welche beobachtet wird. (ReadOnly)	

Methoden/Konstruktor:

public	NumberObserver(int watchNumber)	Initialisiert eine neue Instanz der NumberObserver-Klasse mit der zu beobachteten Zahl.	
--------	--	---	--

Methoden:

public	void Notify(...)	Diese Methode wird vom Observable-Objekt (Dice) aufgerufen.	
public	string ToString()	Konvertiert die Statistikdaten zu einer Zeichenfolge.	

Benutzerdialog

Die Interaktion mit dem Benutzer soll über folgenden Dialog durchgeführt werden:

```

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
* Würfel - Ist der Würfel echt?                                           *
* von Prof. Gerhard Gehrher                                              *
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Zahl 1: 13          15,29%
Zahl 2: 14          16,47%
Zahl 3: 19          22,35%
Zahl 4: 15          17,65%
Zahl 5: 12          14,12%
Zahl 6: 12          14,12%
Anzahl der Würfe:      85
Zuletzt geworfene Zahl: 2

Beenden Sie das Programm mit X

```

Hinweis

- Überprüfen Sie Ihr Klassendiagramm mit den Anforderungen dieser Aufgabenstellung. Sie können das Klassendiagramm von Visual Studio mit (View->Add->Classdiagram) anzeigen lassen.
- Beachten Sie, dass Referenzwerte null sein können.
- In allen Klassen befindet sich ein Header mit
 - Klasse
 - Katalognummer
 - Name
 - Datum

Erweiterte Anforderungen für Spezialisten

- keine

Testdokumentation

- Ausführen der Anwendung.

Abgabe

Termin	1 Woche nach Ausgabe der Übung (wenn nicht anders angegeben)
Projektname	DiceSimulationXX XX...die zweistellige Katalognummer
Format	zip-Format
Ort	Moodle/Kurs - Programmieren

Viel Erfolg!