Fachhochschul-Bachelorstudiengang
SOFTWARE ENGINEERING
A-4232 Hagenberg, Austria

Titel der Bachelorarbeit

Bachelorarbeit

zur Erlangung des akademischen Grades
Bachelor of Science in Engineering

Eingereicht von

Vorname(n) Nachname

Begutachtet von Titel Vorname(n) Nachname

Hagenberg, Februar 20xx

Erklärung

Ich erkläre eidesstattlich, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen nicht benutzt und die den benutzten Quellen entnommenen Stellen als solche gekennzeichnet habe. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt.

Die vorliegende, gedruckte Bachelorarbeit ist identisch zu dem elektronisch übermittelten Textdokument.

Datum Unterschrift

ACHTUNG: Bei einer in Englisch abgefassten Arbeit, bitte die obige dt. Version durch die engl. unten ersetzen.

Declaration

I hereby declare and confirm that this thesis is entirely the result of my own original work. Where other sources of information have been used, they have been indicated as such and properly acknowledged. I further declare that this or similar work has not been submitted for credit elsewhere.

This printed thesis is identical with the electronic version submitted.

Date Signature

Abstract

This should contain the abstract of your work.

Kurzfassung

Dieses Kapitel sollte die Kurzfassung deiner Arbeit beinhalten.

Inhaltsverzeichnis

Kapitel 1

Einführung

Dies ist ein Beispiel Kapitel, welches zeigt, wie Code mit minted und Syntax highlighting funktioniert.

1.1 Minted

Listing 1: Referenz zu echten file

```
public class Calculator {
    // declare delegate for method to call asynchronously
    private delegate int AddHandler(int a, int b);
    private AddHandler _addHandler = Add;

    public static int Add(int a, int b) {          // synchronous Add
        Console.WriteLine("Adding on thread " + CurrentThread());
        Thread.Sleep(1000); // simulate workload
        return a + b;
    }

    public IAsyncResult BeginAdd(int a, int b) { // asynchronous Add
        Console.WriteLine("Calling Add from thread " + CurrentThread());
        // use delegate to run Add on different thread
        return _addHandler.BeginInvoke(a, b, null, null);
    }

    public int EndAdd(IAsyncResult result) {
        Console.WriteLine("Waiting for Add on thread " + CurrentThread());
        // use the delegate to block until result is available
        return _addHandler.EndInvoke(result);
    }
}

static void Main() {
    Calculator calc = new Calculator();
    IAsyncResult result = calc.BeginAdd(5, 5); // Start async call
    Console.WriteLine("Main thread " + CurrentThread() + " is not blocked.");
    int calcValue = calc.EndAdd(result); // Call EndAdd to retrieve the result
    Console.WriteLine("Main thread is blocked");
    Console.WriteLine("Calculation returned " + calcValue);
    /* output:          Calling Add from thread 1
                        Main thread 1 is not blocked.
                        Waiting for Add on thread 1
                        Adding on thread 3
                        Main thread is blocked
                        Calculation returned 10 */
}
```

Listing 2: Code direkt im .tex file

```
// <int> is the type of the source elements,
// <long> is the type of the partition-local variable
Parallel.ForEach<int, long>(nums, // source collection
    () => 0, // method to initialize the partition variable (subtotal)
    (j, loop, subtotal) => {
        subtotal += j; // j is the value of the current item
        return subtotal; // value to be passed to next iteration
    },
    (subtotal) => Interlocked.Add(ref total, subtotal)
);
```

Listing 3: Andere Sprache

```
coroutineScope {
    try {
        val result = coroutineScope {
            val a = async {
                throw IllegalStateException()
                5 // needed to compile
            }
            val b = async {
                delay(100)
                5
            }

            a.await() + b.await() // return result to the scope
        }
        println(result) // will not be reached
    } catch (e:Exception) {
        println("handled: $e") // => handled: java.lang.IllegalStateException
    }
}
```

Aktuell wird nur Kotlin und C# unterstützt. Um eine neue Sprache hinzuzufügen, muss in CustomListings.sty ein neues Environment zum Schluss angefügt werden.

Quellenverzeichnis