03-IBVP-REPRO	Reaktive Programmierung
	Reactive Programming

Lehrform *(teaching format)* / **SWS** *(hours per week)*: 2VL + 2UE

Kreditpunkte *(credit points)*: 6

Turnus *(frequency)*: unregelmäßig

Inhaltliche Voraussetzungen *(content-related prior knowledge/skills)*: Praktische Informatik 3

Sprache *(language)*: Deutsch

Lehrende *(teaching staff)*: Prof. Dr. Christoph Lüth

Studiengang *(degree program)*	Module	Semester
Informatik (Bachelor VF, BPO'20)	IBVP	ab 5. Sem.
Informatik (Bachelor VF, BPO'26)	*General Studies*	ab 5. Sem.

Lernergebnisse:

Das Paradigma der reaktiven Programmierung stellt Datenströme und die Verbreitung von Änderungen in einer Anwendung in den Vordergrund. Dabei sollen idealerweise Abhängigkeiten zwischen berechneten und veränderlichen Werten sowie externen Datenquellen deklarativ definiert werden. Reaktive Programmierung kommt heute aber auch häufig bei hochgradig verteilten Systemen zum Einsatz, bei denen sehr kurze Antwortzeiten und hohe Fehlertoleranz wichtig sind.

In der Veranstaltung werden die relevanten Konzepte erarbeitet und dabei die sehr mächtige Sprache Scala gelernt. Außerdem tauchen wir tiefer in fortgeschrittene und nebenläufige Aspekte der funktionalen Programmierung ein.

Learning Outcome:

The paradigm of reactive programming focuses on data streams and the and the propagation of changes in an application. foreground. Ideally, dependencies between calculated and values and external data sources should be defined declaratively. data sources should be defined declaratively. Today, however, reactive programming is also frequently used for highly distributed systems, where very short response where very short response times and high fault tolerance are important.

In the course, the relevant concepts will be presented and we will learn the very powerful programming language Scala. We will also we will delve deeper into advanced and concurrent aspects of functional programming.

Translated with www.DeepL.com/Translator (free version)

Inhalte:

- Grundkonzepte der reaktiven Programmierung;
- Futures, reaktive Ströme, Observables
- Aktoren und Akka;
- Bidirektionale Programmierung;
- Fehlertoleranz und Robustheit;
- Umgang mit der Programmiersprache Scala;
- Fortgeschrittene funktionale Programmierung.

Contents:

- Basic concepts of reactive programming;
- Futures, reactive streams, observables;
- actors and Akka;
- Bidirectional programming;
- fault tolerance and robustness;
- how to use the Scala programming language;
- advanced concepts of functional programming.

Hinweise *(remarks)*: In der Tabelle sind nur die primären/spezifischsten Module aufgelistet, denen diese Veranstaltung zugeordnet ist.