|  |  |  |
| --- | --- | --- |
| **Kurs:** Distributed & Mobile Systems | | **Code:** W2PROG20 |
| **Studiengang** | BSc Wirtschaftsinformatik | |
| **Studiengruppe** | BWI-A19 | |
| **Dozent/in** | Walter Rothlin / Frank-Stefan Heinz | |
| **Leistungsnachweis(e)** | Einzelarbeit mit Fachgespräch | |
| **Präsenz** | **Selbststudium** | **Workload** |
| 40 h | 30 h | 70 h |

|  |
| --- |
| **Lernergebnisse** |

Die Studierenden

* Sie können eine Client-Server Applikation auf der Basis von TCP/IP Sockets realisieren.
* Sie können verteilte Applikationen konzipieren und entwickeln.
* Sie kennen die grundlegenden Konzepte, Techniken und Methoden, um verteilte und mobile Softwaresysteme zu designen und in Python zu implementieren.
* Sie sind in der Lage, bei Software-Architekturfragen einen entscheidenden Beitrag zur Lösung beitragen.
* Sie kennen die wichtigsten SOA – Konzepte (SOAP / REST) und können einfache Webservices selbst in Python entwickeln

|  |
| --- |
| **Lehrinhalte** |

1. TCP/IP Socket communication
2. XML-Processing in Python
3. Services via TCP/IP (csv und xml)
4. 2-Tier Application (Python Frontend / DB as Backend)
5. Bestehender Web-Services (SOAP / REST) in Python nutzen
6. DAO dazu implementieren und Application refactoring
7. Message queues in Python
8. Mobile Entwicklung und Marketing Rundgang (Frank-Stefan Heinz)
9. Eigener Web-Service in Python implementieren
10. IoT Anwendung entwickeln (Compose services to a new service)

|  |
| --- |
| **Lehr- und Lernmethoden** |

* Stoffvermittlung Frontalunterricht, Übungen lösen im Präsenzunterricht und Selbststudium

|  |
| --- |
| **Leistungsnachweise** |

Leistungsnachweise:

* 1 Python Programm mit mindestens einer Klasse anhand von Requirements nach den Clean-Code Regeln in Einzelarbeit entwickeln (Gewicht: 50%)
* Ein Fachgespräch zur Einzelarbeit und den Übungen (Gewicht: 50%)

|  |
| --- |
| **Lehrmittel** |

Verpflichtend

Inhalt vom Module Web- & N-tier Applications verstanden. Python-Kurse bestanden und beherrscht seine Entwicklungsumgebung.

|  |
| --- |
| **Semesterprogramm** |

|  |  |
| --- | --- |
| 1. **TCP/IP Socket communication** | |
| Datum | Di 21.09.2021 |
| Zeit | 17:30 – 20:45 |
| Dozent | Walter Rothlin |

Vorbereitung

* GUI applications in Python (Buch Kapitel 10)

Inhalte

* Ein eigener Client und Server in Python implementieren, welcher über eine Socket-Communication Daten austauscht.

|  |  |
| --- | --- |
| 1. **XML-Processing in Python** | |
| Datum | Di 05.10.2021 |
| Zeit | 17:30 – 20:45 |
| Dozent | Walter Rothlin |

Inhalte

* Die drei Grundoperationen (Validierung, Transformation und Data-Access) mit einem XML Dokument in Python implementieren

|  |  |
| --- | --- |
| 1. **Services via TCP/IP (csv und xml)** | |
| Datum | Do 14.10.2021 |
| Zeit | 09:00 – 12:15 |
| Dozent | Walter Rothlin |

Vorbereitung

* TCP-IP Socket Communication Beispiel und XML Operationen in Python verstehen
* String-Operatione (split, join,…) anwenden können

Inhalte

* Erweitern der TCP/IP Socket Applikation als Service Call

Nachbereitung

* Testen und Fertigstellen der Applikation

|  |  |
| --- | --- |
| 1. **2-Tier Application (Python Frontend / DB as Backend)** | |
| Datum | Do 21.10.2021 |
| Zeit | 13:15 – 16:30 |
| Dozent | Walter Rothlin |

Vorbereitung

* DB-Anbindung in Python

Inhalte

* GUI Application für Adress-DB entwickeln

|  |  |
| --- | --- |
| 1. **Bestehender Web-Services (SOAP / REST) in Python nutzen** | |
| Datum | Di 02.11.2021 |
| Zeit | 17:30 – 20:45 |
| Dozent | Walter Rothlin |

Vorbereitung

* Web-Services (Unterschied SOAP / REST) konzeptionell verstehen

Inhalte

* API eines bestehender Web-Service (Google, Open-Weather, Map.geo.admin,…) verstehen
* Applikation in Python implementieren, welche einen *open* WEB-Service nutzt

Nachbereitung

* Applikation testen und fertig implementieren

|  |  |
| --- | --- |
| 1. **DAO dazu implementieren und Application refactoring** | |
| Datum | Di 16.11.2021 |
| Zeit | 17:30 – 20:45 |
| Dozent | Walter Rothlin |

Inhalte

* Redesign der WEB-Service Application (vom Service Aufruf zu OO Methode calls)

Nachbereitung

* Redesign implementieren und testen

|  |  |
| --- | --- |
| 1. **Message queues in Python** | |
| Datum | Di 30.11.2021 |
| Zeit | 17:30 – 20:45 |
| Dozent | Walter Rothlin |

Vorbereitung

Funktion und Aufgabe eines Message Broker erklären können

Inhalte

* Message oriented programing in Python mit RabbitMQ

Nachbereitung

* Applikation testen

|  |  |
| --- | --- |
| 1. **Mobile Entwicklung und Marketing ein Rundgang** | |
| Datum | Di 14.12.2021 |
| Zeit | 17:30 – 20:45 |
| Dozent | Frank-Stefan Heinz |

Inhalte

* Mobile Entwicklung und Marketing ein Rundgang

|  |  |
| --- | --- |
| 1. **Eigener Web-Service in Python implementieren** | |
| Datum | Di 11.01.2022 |
| Zeit | 17:30 – 20:45 |
| Dozent | Walter Rothlin |

Inhalte

* Ein eigener REST-Service in Python implementieren und in einer Reference Applikation einbauen

|  |  |
| --- | --- |
| 1. **IoT Anwendung entwickeln (Compose services to a new service)** | |
| Datum | Di 25.01.2022 |
| Zeit | 17:30 – 20:45 |
| Dozent | Walter Rothlin |

Inhalte

* IoT Anwendung selber entwickeln (inkl. QR-Code)
* Beispiel Smart Home