| **Thema / Inhalt** | **Methode** | **Zeitbedarf** | **Hausaufgaben** |
| --- | --- | --- | --- |
| 1.Abend   * Sie können die Methoden \_\_init\_\_ \_\_str\_\_ in einer Klasse erläutern und deren Einsatzmöglichkeiten aufzeigen. (K2) * Sie können erläutern was eine ENUM ist. (K2) | | | |
| **Bruchklasse**   * \_\_init\_\_, \_\_str\_\_ * properties * private / public / properties * statische Elemente * API docs / Doc-Strings * Enumerations * sub-classes | Test-Driven Approach mit Reviews und Refactoring  Theoretischen Einschüben | 200’ | Bruchrechner **ohne**  Operator-Overloading fertigstellen |
| 2.Abend   * Sie können eine eigene Klasse designen, implementieren und testen. (K3) * Sie können die Sichtbarkeitsstufen von Methoden und Properties richtig nutzen. (K3) * Sie können ENUMS sinnvoll nutzen. (K3) | | | |
| **Operator-Overloading**   * \* and \*\* parameters * Refactoring Logger-Class * Bruch-Klasse (Operator Overloading) | Test-Driven Approach mit Reviews und Refactoring  Theoretischen Einschüben | 200‘ | Bruchrechner **mit**  Operator-Overloading fertigstellen |
| 3.Abend   * Sie können den Unterschied zwischen Modul, Packages und Klassen sowie Bibliotheken erläutern. (K3) * Sie können bestehende Python-Applikationen untersuchen und deren Funktionalität sowie Schwachstellen identifizieren. (K4) | | | |
| **Leistungskontrolle 1 (Classes)**   * Packages and Module * Exception-Handling / Eigene Exceptions | Von einer vorgegebenen Klasse den \_\_init\_\_ erweitern.  Bei einer vorgegebenen Bruch-Klasse den \_\_init\_\_ so erweitern, dass die Parameter als Strings und Floats, negativ, … übergeben werden können.  Test-Driven Approach mit Reviews und Refactoring  Theoretischen Einschüben | 60‘  140‘ | Bruchrechner mit eigenen Exceptions erweitern |
| 4.Abend   * Sie können komplexere Applikationen in Python schreiben und Konzepte von Sub-Classing, Multithreading, ENUMS und  Template-Mechanism (JINJA) verwenden. (K3) * Sie können einen http Request und eine JSON/XML-Response beschreiben. (K2) * Sie können API-Dokumentationen analysieren und daraus ableiten, wie externe Pakete in eigene Anwendungen integriert werden können. (K4) | | | |
| **Multithreading**   * fork and join * time-events | Test-Driven Approach mit Reviews und Refactoring  Theoretischen Einschüben | 200‘ |  |
| 5.Abend   * Sie können die Architektur einer Python-Anwendung analysieren, um Verbesserungspotenziale zu erkennen (K4). * Sie können verschiedene Ansätze für die Steuerung von Aktoren auf einem Raspberry Pi kritisch vergleichen und Empfehlungen aussprechen. (K5). | | | |
| **Leistungskontrolle 2**    **Template-Mechanism (JINJA)**   * Pain001 Meldung * Geo-Taggen (KML-Files) | Test-Driven Eine Bestehende Klasse muss abgeleitet werden und eine Methode überschrieben, sowie eine zusätzliche Methode implementiert werden.  Eine eigene (MySenseHat) Classe als Ableitung von SenseHat implementieren. Die set\_pixel() Methode ist zu overloeaden, so dass x,y auch ausserhalb der Grenzen übergeben werden können und alle Parameter auch als Float und Str übergeben werden können. draw\_line() ist noch zu implementieren.  Test-Driven Approach mit Reviews und Refactoring  Theoretischen Einschüben | 60’  140‘ | Eine JINJA Template Applikation fertig implementieren und testen. |
| 6.Abend   * Sie können öffentliche REST-Services analysieren und in eigenen Python-Applikationen nutzen. (K4). | | | |
| **FLASK Web-Apps**   * Anotations * Request / Response * static HTML / templates * Mime-Type (JSON Responses) * Parameter Uebergabe (get/put) * Endpoints | Test-Driven Approach mit Reviews und Refactoring  Theoretischen Einschüben | 200‘ |  |
| 7.Abend   * Sie können die Vor- und Nachteile von service-orientierter gegenüber funktionaler Programmierung in Python bewerten. (K5) | | | |
| **FLASK Web-Apps (Cont.)**   * Anotations * Request / Response * static HTML / templates * Mime-Type (JSON Responses) * Parameter Uebergabe (get/put) * Endpoints | Test-Driven Approach mit Reviews und Refactoring  Theoretischen Einschüben | 200‘ |  |
| 8.Abend   * Sie können die Effizienz und Sicherheit verschiedener Methoden für den Zugriff auf Raspberry Pi-Systeme beurteilen. (K5) | | | |
| **Leistungsnachweis (Modullernzielkontrolle MILZ)**:  Eine Aufgabe unter Zeitdruck gemäss Spezifikationen nach dem Test-Driven Approach implementieren. | Implementation eines REST Services mit FLASK und JINJA.  Implementieren Sie für einige Methoden in ihrer MySenseHat-Class ein REST API und die entsprechenden Test-Fälle. | 200’ |  |
| 9.Abend   * Sie können den Einsatz von Sense-HAT-Modulen und PiPlates in einem Projekt evaluieren und deren Nutzen für spezifische Anwendungen bewerten.(K5) | | | |
| **Fachgespräche über MLZ**   * GPIO (LED-Steuerung) * PiPlates * Shellys Home-Automation | Einzelgespräche gemäss Zeitplan  Selbstorganisiertes Lernen (SOL) | 200‘ |  |

|  |  |
| --- | --- |
| **Bemerkungen:** | * Jeder Abend dauert 4 Lektionen. * Der Unterrichtsplan kann bei Bedarf dem vorhandenen Wissen der Klasse angepasst werden. * Die Studierenden lösen die Übungen auf ihren privaten Notebooks und dem eigenen RaspberryPi. * Der Leistungsnachweis (**MLZ**) am 8.Abend ist in Einzelarbeit in der vorgegebenen Zeit zu erstellen |