## Teilnehmer/in des Teams:

|  |  |  |  |
| --- | --- | --- | --- |
| Name: | | Vorname: | |
|  | | | |
| Abgabedatum : | Klasse: | | Team: |

## Aufgabe 3: (Individuelle Aufgabe) Lösungsdesign mit Script-Sprache implementieren & kommentieren

Implementieren Sie die beschriebene Aufgabenstellung gemäss Ihrem Design. Definieren Sie den genauen Umfang des Programmcodes bevor Sie mit der Arbeit beginnen. Bei einem umfangreichen Projekt (mehrere Module, ext. Funktionen) können Sie als Team *ein* Projekt realisieren, wobei jedes Teammitglied genau definierte Teile implementiert. Die Teile sind klar mit dem Namen des Erzeugers zu kennzeichnen, so dass eine individuelle Bewertung möglich ist. Die Aufteilung ist mit der Lehrperson zu besprechen!

Der Programmcode ist "in-line" gemäss üblichem Standard zu kommentieren.

Erstellen Sie einen Projektordner und sammeln Sie darin alle Quelldateien ihres Projektes. Speichern Sie den fertigen Ordner als komprimierte ZIP-Datei auf dem BSCW ab: ***MP122\_ Aufgabe\_3\_IhrName.zip***

Im Fachgespräch mit der Lehrperson führen Sie das fertige Projekt zwecks Abnahme vor.

## Aufgabe 4: (Individuelle Aufgabe) Testdurchführung, Testprotokoll und Dokumentation

Nachvollziehbare protokollierte Testfälle stellen die Funktionalität der Applikation sicher. Führen Sie die in der Aufgabe 2 definierten Testfälle (*MP122\_Aufgabe\_2\_Team.docx*) mit ihrem implementierten Projekt durch und dokumentieren Sie diese im vorbereiteten Testprotokoll. Die so protokollierte Testvorschrift ist auf dem BSCW (Angaben dazu erhalten Sie von der Lehrperson) unter folgendem Namen abzuspeichern:  
 ***MP122\_Aufgabe\_4A\_Test\_IhrName.docx***

Ergänzen Sie ihr Dokument aus Aufgabe 1 (*MP122\_Aufgabe\_1\_Team.docx*):

Beschreiben Sie die Struktur ihres Programmcodes (z.B. Logik, Funktionen, Variablen, UML). Fügen Sie eine sinnvolle *Installations*- und/oder *Bedienungsanleitung* bei.

Speichern Sie die fertige Doku Dokument auf dem BSCW:  
 ***MP122\_Aufgabe\_4B\_Doku\_IhrName.docx***

## Bewertungstabelle:

Werden die mittig beschriebenen Kriterien der Auftragserfüllung erreicht, so ergibt dies die Note 4. Pro Kriterium kann je nach Erfüllungsgrad entweder eine halbe oder eine viertel Note abgezogen oder eine halbe oder eine viertel Note dazu gezählt werden.

|  |  |  |  |
| --- | --- | --- | --- |
| **Kriterium** | **- 0.5 Notenpunkte** | **Note 4** | **+ 0.5 Notenpunkte** |
| **Aufgabe 3:** Implementation  *(Quellcode)* | ⬜ Sehr rudimentäre Funktionen  ⬜ Implementation unlogisch oder fehlerhaft | ⬜ Implementation gemäss Problembeschreibung funktionsfähig  ⬜ Code kommentiert | ⬜ Korrekte und dem Design entsprechende Implementation  ⬜ Quellcode ausführlich und verständlich kommentiert |
| **Aufgabe 3:**  *Abgabe inkl. Fachgespräch 5-10min* | ⬜ Keine oder fehlerhafte Demo ohne Ursachenerkenntnis, Programm unfertig  ⬜ Aufgabe 3 oder 4 zu spät oder nicht abgegeben | ⬜ Demo und Erläuterung eines funktionierenden Programms durch den Lernenden  ⬜ Abgabe Aufgaben 3 und 4 termingerecht | ⬜ Lernender kann sein Programm mit weiterführender Erkenntnis verbinden / kommentieren.  ⬜ Komplexes Projekt erfolgreich umgesetzt |
| **Aufgabe 4:**  *Abgabe inkl. Fachgespräch 5-10min* | ⬜ Kein, falsch oder nicht ausgefülltes Testprotokoll | ⬜ Testprotokoll ausgeführt und signiert | ⬜ Testprotokoll von Drittperson korrekt ausgefüllt  ⬜ Testresultat klar dokumentiert |
| **Aufgabe 4:**  Dokumentation der Implementation | ⬜ Keine oder unklare Doku der Implementation vorhanden | ⬜ Einfache Beschreibung der Implementation und der Bedienung | ⬜ Ausführliche Beschreibung der internen Funktionen, der Ablaufstruktur und der Bedienung. (z.B. Struktogramm, UML, Storyboard) |

|  |  |
| --- | --- |
| ***Einzel Note*** |  |
| ***Team Note (Übertrag aus Teamaufgaben)*** |  |
| ***Total Einzel Note (auf 0.5 gerundet)*** |  |

|  |
| --- |
| ***Bemerkungen Expert/in:***  *Bei der Beurteilung ist das Nichterreichen der minimalen Anforderungen (Prüfungsnote < 4) zu begründen.* |

Unterschrift Expert/in: