|  |  |
| --- | --- |
|  | |
|  |  |
|  |  |
| Modul 122 | |
| Projektdokumentation LB1  Thema: | |
|  |  |
|  |  |
| Modul | IET-122 |
| Eingereicht von | Praxisprojekt – Stammgruppe |
| Eingereicht bei | Oliver Schramm |
| Datum | 26. Mai 2022 |

Dokumentinformation

| Auftraggeber | Oliver Schramm |
| --- | --- |
| Team |  |
| Lernpartner | Ian Kraft, Jonas S. Büchi, Michel Lam, Ilian Häsler |

Änderungsverzeichnis:

| Datum | Version | Änderung | Autor |
| --- | --- | --- | --- |
| 31.01.2022 | 0.1 | Vorlage | Urs Dummermuth |
| 09.05.2022 | 0.2 | Ziele (SMART), Realisierung, Integration, Realisierungsbericht | Jonas & Michel |
| 16.05.2022 | 0.3 | Anforderung, Potenzial, Vorgehensmodell, Python, Realisierungsbericht | Jonas & Michel |
| 23.05.2022 | 0.4 | Debugging/Testing, Diagramme/Prozesse, Realisierungsbericht, Skript & Programmiersprache | Jonas, Michel & Ian |

Inhaltsverzeichnis

[1 Praxisprojekt - Auftragsbeschrieb 3](#_Toc104209583)

[1.1 Zweck 3](#_Toc104209584)

[1.2 Ziele 3](#_Toc104209585)

[1.3 Realisierung 4](#_Toc104209586)

[1.4 Integration 4](#_Toc104209587)

[2 Analyse 5](#_Toc104209588)

[2.1 Anforderungen 5](#_Toc104209589)

[2.2 Potenzial 5](#_Toc104209590)

[2.3 Vorgehensmodell 5](#_Toc104209591)

[2.4 Python 6](#_Toc104209592)

[2.5 Debugging/Testen 6](#_Toc104209593)

[3 Design 10](#_Toc104209594)

[3.1 Diagramme/Prozesse 10](#_Toc104209595)

[PAP/Nassi/Flussdigramm 10](#_Toc104209596)

[3.2 Kontrollstrukturen 10](#_Toc104209597)

[4 Praxisprojekt - Realisierung 11](#_Toc104209598)

[4.1 Skript-/Programmiersprache 11](#_Toc104209599)

[5 Dokumentation 12](#_Toc104209600)

[5.1 Realisierungsbericht / Arbeitsjournal 12](#_Toc104209601)

[6 Präsentation, Reflexion 13](#_Toc104209602)

[6.1 Vortrag, Screencast-Video 13](#_Toc104209603)

[6.2 Demo 13](#_Toc104209604)

[6.3 Reflexion, Lessons Learned 13](#_Toc104209605)

[7 Anhang 14](#_Toc104209606)

[8 Verzeichnisse 15](#_Toc104209607)

[8.1 Abbildungsverzeichnis 15](#_Toc104209608)

[8.2 Quellenverzeichnis 15](#_Toc104209609)

# Praxisprojekt - Auftragsbeschrieb

## Zweck

Das wird ein Taschenrechner mit den vier Grundoperationen, Wurzel, Quadrat und den Trigonometrie-Funktionen. Dieser Taschenrechner wird zusätzlich noch eine grafische Benutzeroberfläche (User Interface UI) besitzen.

## Ziele

* **Spezifisch (S)**
  1. Funktional
     + Man kann die vier Grundoperationen ausführen, Wurzel ziehen, quadrieren und die Trigonometrie anwenden.
  2. Nicht funktional
     + Der Taschenrechner hat eine grafische Benutzeroberfläche, welche auch benutzerfreundlich ist. Man kann auf die Tasten im GUI drücken.
     + Der Taschenrechner stürzt nicht ab, egal was man bei den Benutzereingaben eingibt. Alle Fehler können aufgefangen werden.
  3. Organisation
     + Jedem wurde eine Aufgabe zugeteilt und jeder ist beschäftigt.
* **Messbar (M)**
  1. Funktional
     + Wir wissen, dass wir unser Ziel erreicht haben, indem wir alle vorgegebenen Operationen benutzen können.
  2. Nicht funktional
     + Sobald das GUI mit allen vorgegeben Operationen, sowie alle Zahlentasten vorhanden sind, können wir feststellen, dass wir unser Ziel erreicht haben.
     + Sobald der Taschenrechner fertig ist, kann man mithilfe von Test die Eingaben prüfen. Wenn das Programm nicht abstürzt, wurde dieses Ziel erreicht.
  3. Organisation
     + Wenn jeder etwas zu tun hat, sind die Aufgaben fair verteilt. Das heisst, sobald niemand «nichts» macht, stellen wir fest, dass wir dieses Ziel ebenfalls erreicht haben.
* **Akzeptanz (A)**
  + - Alle sind mit den Zielen einverstanden und möchten diese dementsprechend auch erreichen.
* **Realistisch (R)**
  1. Funktional orientiert
     + Die ausgewählten Operationen erfordern keine grossen Kenntnisse bezüglich der Programmiersprache Python. Somit ist das machbar.
  2. Nicht funktional orientiert
     + Aufgrund von unseren Vorkenntnissen aus der Berufsbildungscenter AG (Bbc), entsprechen sie auf unserem Niveau.
  3. Organisation
     + Es gehört immer dazu, dass in einer Gruppe jeder beschäftigt ist. Das ist obligatorisch. Das ist definitiv machbar und hat die oberste Priorität.
* **Terminiert (T)**
  + - Da alles parallel erledigt wird basierend auf das zugeteilte Individuum, werden alle Ziele um die gleiche Zeit abgeschlossen sein.

## Realisierung

Für das Programm wird lediglich die Anwendung PyCharm Community Edition genutzt. Für die Dokumentation wird Word benutzt. Für die Datenaustausch verwendeten wir GitHub. Mithilfe von Git konnten wir lokale Dateien auf GitHub hochladen und umgekehrt.

## Integration

Wie können Sie das gelernte aus dem Projekt und Unterricht in ihre berufliche Praxis intergieren/nutzen?

Wir lernten uns mit GitHub zu organisieren und Gruppenarbeiten zu machen. GitHub ermöglicht uns, Konfliktfrei gleichzeitig zu arbeiten. Parallel haben wir unser Wissen bezüglich Python erweitert. Mit diesen neuen Wissen können wir für andere kleinere Projekte einsetzen im beruflichen Alltag. Das gilt auch für GitHub. GitHub kann für kleinere und grössere Projekte verwenet werden.

# Analyse

## Anforderungen

Formulieren Sie hier die Anforderungen funktionale und nicht funktionale aus der von ihnen gewählten Aufgabenstellung

Funktionale

* Der Taschenrechner kann verschiedene Operationen durchführen
* Der Taschenrechner kann zwei Werte zusammenrechnen
* Der Taschenrechner kann das Resultat anzeigen
* Der Taschenrechner bietet ein User Interface dar

Nicht funktionale

* Im User Interface sind Buttons vorhanden
* Die Resultate sollen möglichst schnell zurückgegeben werden
* Der Taschenrechner kann mit grossen Zahlen arbeiten

## Potenzial

Selbstverständlich könnte man den Taschenrechner weiter ausbauen. Der Taschenrechner könnt eine grafische User Interface bzw. Benutzeroberfläche haben. Man könnte noch weiter gehen und zusätzlich noch auf die Benutzerfreundlichkeit achten. Auch könnte man die Fensterauflösung individualisierbar machen, sprich, ab eine bestimmte Grösse wird sich die Benutzeroberfläche neu anpassen.

Generell könnte man noch viele andere Operatoren einbauen wie z.B. Sin, Cos und Tan etc.

## Vorgehensmodell

Beschreiben sie hier stichwortartig das von ihnen gewählte Vorgehen, weshalb und mit welchen Tools haben sie genau dieses Szenario gewählt?

1. Anforderungen überprüfen / analysieren (Einschätzen der Situation)
2. Arbeitsplatz einrichten (Vorbereitung)
3. Material organisieren (Organisation)
4. Ziele definieren (SMART) (Erreichbarkeit herausfinden)
5. Aufgaben-Teilgebiete basierend auf die Ziele definieren (Organisation der Gruppe)
6. Aufgaben verteilen (Organisation in der Gruppe)
7. Benötigte Dokumente vorbereiten (Vorbereitung)
8. Aufgaben prozessieren (Realisierung)
9. Statusbericht der Mitglieder (Einschätzen der Situation)
10. Dokumente aktualisieren (Administration)
11. Testfälle definieren (Testing & Überprüfung)
12. Testfälle durchführen (Fehler finden)
13. Testfälle protokollieren (Testprotokoll) (Für die Fehlerbehebung)
14. Code aktualisieren (Fehler effektiv beheben)
15. Div. Code kommentieren (Verwaltung)
16. Präsentation vorbereiten (Vorbereitung)
17. Reflexion schreiben und Rückblick anschauen (Selbstreflexion & Verbesserung)

## Python

* Entwicklungsumgebung
  + PyCharm Community v2022.1
* Interpreter
  + Python v3.10
* Betriebssystem
  + Windows 10 (Microsoft)
* Gerät
  + Ie203-ws21
  + Ie203-ws20
  + Ie203-ws19
  + Ie203-ws18

## Debugging/Testen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test-ID** | T01 | | | |
| **Vorbedingung** | * Der Computer muss Strom haben * Der Computer muss eingeloggt sein * PyCharm Software muss installiert sein und funktionieren * Interpreter Python wurde festgelegt * PyCharm muss geöffnet sein | | | |
| **Ablauf** | 1. Gehe mit der Maus nach oben rechts an das grüne Dreieck 2. Klicke auf das grüne Dreieck | | | |
| **Erwartung** | Das Projekt läuft | | | |
| **Resultat** | Erfolgreich | x | Fehlgeschlagen |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test-ID** | T02 | | | |
| **Vorbedingung** | * T01 muss funktionieren | | | |
| **Ablauf** | 1. Wähle den ersten Operator (Addition) 2. Schreibe eine valide Zahl bei der ersten Abfrage 3. Schreibe eine valide Zahl bei der zweiten Abfrage | | | |
| **Erwartung** | Der Rechner hat die beiden Zahlen addiert. | | | |
| **Resultat** | Erfolgreich | x | Fehlgeschlagen |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test-ID** | T03 | | | |
| **Vorbedingung** | * T01 muss funktionieren | | | |
| **Ablauf** | 1. Wähle den zweiten Operator (Addition) 2. Schreibe eine valide Zahl bei der ersten Abfrage 3. Schreibe eine valide Zahl bei der zweiten Abfrage | | | |
| **Erwartung** | Der Rechner hat die beiden Zahlen subtrahiert. | | | |
| **Resultat** | Erfolgreich | x | Fehlgeschlagen |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test-ID** | T04 | | | |
| **Vorbedingung** | * T01 muss funktionieren | | | |
| **Ablauf** | 1. Wähle den dritten Operator (Multiplikation) 2. Schreibe eine valide Zahl bei der ersten Abfrage 3. Schreibe eine valide Zahl bei der zweiten Abfrage | | | |
| **Erwartung** | Der Rechner hat die beiden Zahlen multipliziert. | | | |
| **Resultat** | Erfolgreich | x | Fehlgeschlagen |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test-ID** | T05 | | | |
| **Vorbedingung** | * T01 muss funktionieren | | | |
| **Ablauf** | 1. Wähle den vierten Operator (Division) 2. Schreibe eine valide Zahl bei der ersten Abfrage 3. Schreibe eine valide Zahl bei der zweiten Abfrage | | | |
| **Erwartung** | Der Rechner hat die beiden Zahlen dividiert. | | | |
| **Resultat** | Erfolgreich | x | Fehlgeschlagen |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test-ID** | T06 | | | |
| **Vorbedingung** | * T01 muss funktionieren | | | |
| **Ablauf** | 1. Wähle den fünften Operator (Wurzel ziehen) 2. Schreibe eine valide Zahl | | | |
| **Erwartung** | Der Rechner hat die Wurzel gezogen bei dieser Zahl. | | | |
| **Resultat** | Erfolgreich | x | Fehlgeschlagen |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test-ID** | T07 | | | |
| **Vorbedingung** | * T01 muss funktionieren | | | |
| **Ablauf** | 1. Wähle den sechsten Operator (Quadrieren) 2. Schreibe eine valide Zahl | | | |
| **Erwartung** | Der Rechner hat diese Zahl quadriert. | | | |
| **Resultat** | Erfolgreich | x | Fehlgeschlagen |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test-ID** | T08 | | | |
| **Vorbedingung** | * T01 muss funktionieren | | | |
| **Ablauf** | 1. Wähle irgendein Operator 2. Schreibe eine invalide Zahl bei der ersten Abfrage | | | |
| **Erwartung** | Der Rechner fordert eine Valide Zahl | | | |
| **Resultat** | Erfolgreich | x | Fehlgeschlagen |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test-ID** | T09 | | | |
| **Vorbedingung** | * T01 muss funktionieren | | | |
| **Ablauf** | 1. Wähle irgendein Operator 2. Schreibe eine valide Zahl bei der ersten Abfrage 3. Schreibe eine invalide Zahl bei der zweiten Abfrage | | | |
| **Erwartung** | Der Rechner fordert eine Valide Zahl | | | |
| **Resultat** | Erfolgreich | x | Fehlgeschlagen |  |

# Design

Ein Bild, das Text, Screenshot, Monitor, Fernsehen enthält.

Automatisch generierte Beschreibung

## PAP/Nassi/Flussdigramm

Abb. 1 Diagramm

# Praxisprojekt - Realisierung

## Skript-/Programmiersprache

* Welche Vorteile / Nachteile bietet ihnen Python gegenüber Bash oder Power Shell, denk Sie auch an Plattformunabhängigkeit, Komfort, Entwicklungsumgebung$
* Vorteile Python
  + Einfach zu lesen, lernen und schreibt (Für Anfänger geeignet)
  + Plattform unabhängig
  + Flexibel
  + Objektorientiert
  + Schwache Typisierung (Keine klare Datentypen zu variablen)
* Nachteile Python
  + Langsam
  + Open-Source => Kann nicht verschlüsselt werden
  + Speicherverbrauch
  + Fehler werden nur zur Laufzeit angezeigt
* Einbinden der Skript Bestandteile ihres Projektes

# Dokumentation

## Realisierungsbericht / Arbeitsjournal

#### Arbeitsjournale (KW19)

An diesem Montag hat jeder seine Aufgabe erledigt. Ian Kraft arbeitete an der Logik der Taschenrechner und ist fertig geworden. Statt Potenzen, realisierte er Cosinus im Taschenrechner. Ilian Häsler arbeitete am User Interface. Er hat vor allem an Grids gearbeitet, damit alle Buttons schön strukturiert sind. Michel und Jonas haben ausschliesslich an der Dokumentation gearbeitet. Sie haben Ziele (S M A R T) definiert. Gemeinsam haben wir (bevor wir überhaupt gearbeitet haben) alle demokratisch abgestimmt, wer an was arbeitet. Michel und Jonas haben auch zusammengefasst, welche Technologie wir verwenden und welche Versionen diverse Softwares bzw. Applikationen / Programme hatten.

#### Arbeitsjournale (KW20)

An diesem Tag war Ian Kraft krank. Ilian arbeitete weiter an User Interface, doch es hatte Errors, also haben die anderen versucht zu helfen, um das Problem zu lösen. Michel und Jonas haben noch die Dokumentation gearbeitet. Michel konnte die Fehlermeldung mit dem GUI effektiv lösen.

#### Arbeitsjournale (KW21)

Da Ilian Häsler krank ist und er zuständig für die grafische Benutzeroberfläche zuständig ist, haben wir beschlossen, das ganze mit GUI abzusetzen, indem wir demokratisch abgestimmt haben. Jonas arbeitete an der Dokumentation, Ian Kraft am Flussdiagramm und Michel am Code. Michel versucht alle Wege zu verhindern, welche zu einem Absturz des Programms führen könnte.

# Präsentation, Reflexion

## Vortrag, Screencast-Video



## Demo



## Reflexion, Lessons Learned

Das Projekt war ein voller Erfolg. Alle Teammitglieder haben einiges, vor allem in der Teamorganisation, dazugelernt. Unser Pythonscript wurde bei uns relativ schnell fertig und dabei konnten wir auch einiges zu Python dazulernen. Was bei uns eher weniger gelungen ist, ist die Organisation am Ende des Projektes. Es wurde recht schnell stressig, da die Abgabe bevorstand. Jedoch sind wir alle zufrieden mit dem Projekt und werden das Gelernte gut in der Zukunft einsetzen können.