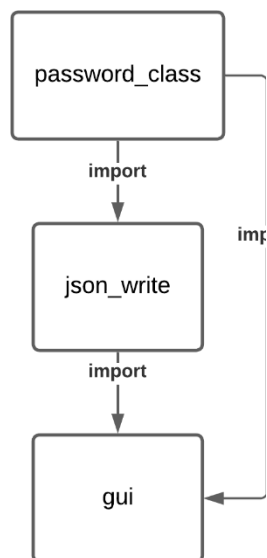


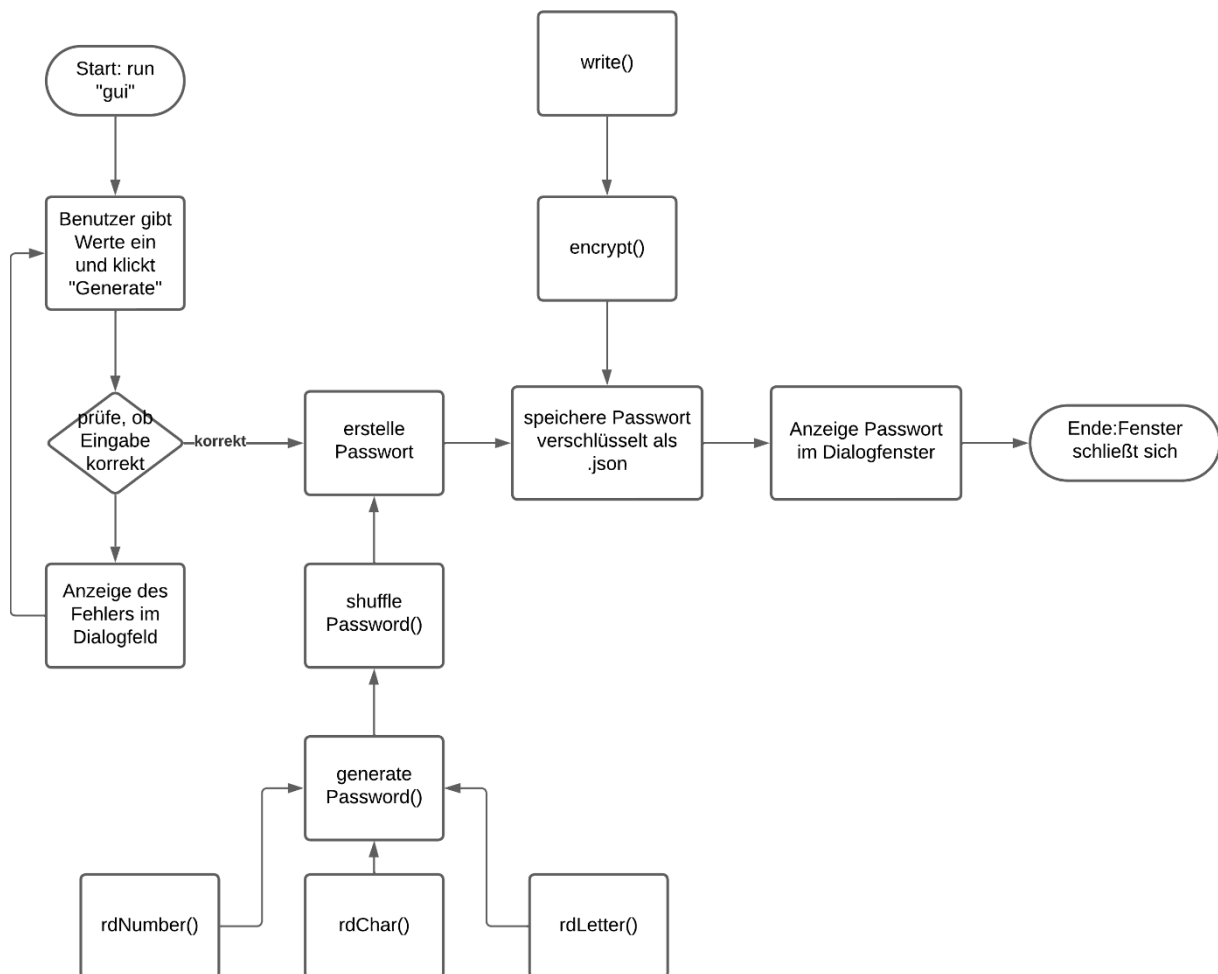
# Dokumentation Passwort Generator

## Allgemeine Projektstruktur

Datei	Beschreibung
Password_class	Klasse, in der das Objekt Passwort und dessen Methoden erstellt werden.
Json_write	Klasse, in der das Objekt JsonWrite und dessen Methoden erstellt werden. Importiert Password_class.
Gui	Main-Datei, in der die GUI definiert und das Programm gestartet wird. Importiert Password_class und Json_write.



## Ablauf des Programms



## Klassen/Dateien und deren Methoden/Funktionen

Klassen/Dateien	Methoden/Funktionen	Beschreibung
Password_class	shufflePassword(self)	Eine Methode, die die Zeichen eines Passworts zufällig durchmischt. :return: zufällig durchmisches password
	generatePassword(self)	Ein Generator, der randomStr Methode nutzt um anhand der Attribute des Objekts Password ein Generator Objekt mit vorgegebener Länge und Anzahl der Buchstaben, Zahlen, Zeichen erstellt. :return: Generator Objekt, das randon generierte Chars anhand von Länge und Anzahl der Buchstaben, Zahlen, Zeichen des Passworts enthält.

	rdChar(self)	Eine Methode, die ein zufällig und kryptographisch sicher generiertes Zeichen nach ASCII Englisch (ohne Leerzeichen) erstellt :return: ein zufällig und kryptographisch sicher generiertes Zeichen nach ASCII Englisch (ohne Leerzeichen)
	rdNumber(self)	Eine Methode, die eine zufällig und kryptographisch sicher generierte Zahl nach ASCII Englisch (0-9) erstellt :return: ein zufällig und kryptographisch sicher generierte Zahl nach ASCII Englisch (0-9)
	rdLetter(self)	Eine Methode, die einen zufällig und kryptographisch sicher generierten Buchstaben nach ASCII Englisch (A-Z, a-z) erstellt :return: ein zufällig und kryptographisch sicher generierter Buchstabe nach ASCII Englisch (A-Z, a-z)
Json_write	write(self):	Funktion, die ein dict des Passworts als JSON abspeichert
	encrypt(self):	Methode verschlüsselt das übergebene Passwort und speichert es in json datei ab :return: key
	decrypt(self, key):	Methode entschlüsselt das verschlüsselte Passwort und gibt dieses wieder als json datei aus :param key: Key, der erstellt wurde bei der Verschlüsselung
	formatString(self, input_string):	Methode filternt einen String nach dem enthaltenen Passwort :param input_string:
	deletePasswordManager(self):	Löscht die unverschlüsselte Datei in dem das Passwort enthalten ist
Gui	functionWrapper():	Functionwrapper, der beim klicken des generiere Passwort Buttons aufgerufen wird. Initialisiert Passwort und json_write Objekte
	checkEntries():	Prüft die eingegebenen Parameter auf Typsicherheit und Values. :return: True falls Parameter in Ordnung sind. False falls es Fehler gibt
	checkSum(length, letters_amount, numbers_amount, chars_amount):	Prüft, ob eingegebene Passwortlänge gleich summe der eingegebenen Zeichentypen ist. :param length: Passwortlänge :param letters_amount: Anzahl Buchstaben :param numbers_amount: Anzahl Zahlen :param chars_amount: Anzahl Sonderzeichen

	<code>checkSmallerZero(letters_amount, numbers_amount, chars_amount):</code>	Prüft, ob Eingabeparameter größer 0 sind. :param letters_amount: Anzahl Buchstaben :param numbers_amount: Anzahl Zahlen :param chars_amount: Anzahl Sonderzeichen
	<code>strParse(string_var):</code>	Parsed stringVar in String. :param string_var: zu parsender stringVar :return: String
	<code>intParse(string_var):</code>	Parsed stringVar in Integer. :param string_var: zu parsender stringVar :return: Integer

## Probleme

Aktuell noch nicht behobenes Problem: Programm merkt, dass ungültige Werte eingegeben wurden und gibt das in einem Fenster auch dem Benutzer aus. Trotzdem wird das Passwort mit den angegebenen Parameters erstellt. Muss irgendwie mit einer Schleife gelöst werden aber im Versuch nur endlos Schleife bekommen.

Ansonsten:

- Verwendung von secrets package für kryptografisch sichere Zufallsvariablen
- Nachsehen welche Ordnungszahlen verschiedene Zeichen bei ASCII haben
- Verschlüsselung eigentlich mit Hash und Salz geplant aber war unpraktisch für Übung, aktuelle Verschlüsselung eigentlich nicht sicher
- Theoretisch kann Passwort auch noch entziffert werden mit decrypt() Methode, diese wird in der functionWrapper() Funktion aktuell aber nicht verwendet. Könnte man nochmal als extra Button „Passwort anzeigen“ implementieren
- Problem war, dass Verschlüsselungsalgorithmus byte type braucht, json aber dann wieder keine byte type abspeichern kann. Musste also geparsed werden.
- Anfangs erst alle Funktionen der Json\_write in der Password\_class gehabt. Dann aber eigene Klasse dafür erstellt, war übersichtlicher.
- Versucht class decorator Methode zu verwenden aber nicht so recht damit zurecht gekommen. Dazu entschieden, dass bei Initialisieren eines Password Objekts Erstellung des Passworts durch „self.password = self.shufflePassword()“ angestoßen wird.
- Programm könnte man jetzt noch zu einem Passwortmanager, zu dem mehrere Passwörter hinzugefügt werden können, erweitern. Diesen dann wiederum durch Passwort schützen.