



---

# PROTOKOLL - QUEUES

---

TEKIN Abdurrahim Burak



20. NOVEMBER 2016

4AHITM

## Aufgabenbeschreibung

# Thread Synchronisation in Python: Queues

Schreibe ein Programm, welches ein simples Erzeuger-Verbraucher-Muster implementiert!

### Grundanforderungen:

- Zwei eigene Klassen (Consumer und Producer) erben von Thread (E1 und V1)
- Die zwei Klassen sind über einen Queue verbunden
- Der Erzeuger E1 sucht nach Primzahlen. Jede gefundene Primzahl wird über die Queue an den Verbraucher V1 geschickt
- Der Verbraucher gibt die empfangene Zahl in der Konsole aus und schreibt sie außerdem in eine simple Textdatei
- Erzeuger und Verbraucher stimmen sich über Queue.task\_done() und Queue.join() ab
- Kommentare und Sphinx-Dokumentation
- Kurzes Protokoll über deine Vorgangsweise, Aufwand, Resultate, Beobachtungen, Schwierigkeiten, ... Bitte sauberes Dokument erstellen! (Kopf- und Fußzeile etc.)

### Erweiterungen:

- Ein weiterer Thread nimmt Benutzereingaben entgegen
- Dieser Thread kann als ein weiterer Erzeuger E2 gesehen werden
- Wird eine (potentiell sehr große) Zahl eingegeben, so wird in einem weiteren Verbraucher V2 überprüft, ob es sich bei dieser produzierten Zahl um eine Primzahl handelt
- E2 und V2 müssen sich nicht über task\_done() absprechen, d.h. E2 kann mehrere Aufträge in die Queue schicken, bevor V2 mit der Bearbeitung fertig ist
- Wird "exit" eingegeben, so werden **alle** Threads sauber beendet
- Achte auf Fehlerfälle!

## Vorgangsweise

Es war wichtig zu wissen wie Queues funktionieren und wie wir sie anwenden. Queues sind Threadsicher!

Wichtig war es noch zu erfahren was der Fehler bei meiner Sphinx-Dokumentation ist! Üblich war es bei meinen Aufgaben, dass wenn ich mit „make html“ die Sphinx-Dokumentation erstelle, die Index.html einfach leer bleibt!

Vom Lehrer habe ich vorgeschlagen bekommen, meine Sphinx-Dokumentation mit PyCharm zu machen, da PyCharm so eine Funktion anbietet!

## Aufwand

Die Aufgabe allgemein hat fast 3 Stunden gedauert, da ich eine lange Zeit probiert habe den Fehler bei meiner Sphinx-Dokumentation zu beheben!

➔ Leider OHNE Erfolg!!!

Nach einer Zeit habe ich auch aufgegeben und einfach alles was ich habe, hochgeladen!

## Resultat

Die Grundanforderungen habe ich größtenteils geschafft, doch da mir die Probleme bei der Sphinx-Dokumentation die Motivation weggenommen haben, habe ich nicht weiter Versucht mir Gedanken zu machen wie man die Erweiterungen machen könnte.

## Schwierigkeiten

UPDATE –

Dank den Debugging-Fähigkeiten vom Herrn Prof. Dolezal habe ich es nun geschafft die Sphinx-Dokumentation zu erstellen und pushe sie im Nachhinein! 😊 😊 😊

Die Schwierigkeiten waren, wie vorher schon erwähnt, dass die erstellten Index.html Dateien leer waren. Nach einer Recherche im Internet habe ich erfahren, dass man für die Behebung, in der conf.py Datei in die Extensions „sphinx.ext.autodoc“ eintragen muss.

Dies habe ich getan, doch danach, wenn ich probiert habe mit PyCharm die Sphinx-Dokumentation zu erstellen, fing PyCharm an das Programm auszuführen!

Der Herr Prof. Dolezal ist darauf gekommen, dass mein Programm nicht überprüft von WEM die Software „gerunned“ wird! ➔ `if __name__ == '__main__':`