**Logbuch Programmierpraktikum**

**Überprüfungen von sinnvollen Problemstellungen!**

Sonderfall dreieckiger Raum 🡪 eine Lampe an beliebigen Punkt ist optimal!

Oder Raum weniger als zwei Seiten oder nicht zusammenhängend + geschlossen + durchdringend? 🡪 nicht notwenidg da rechteckiger Raum? Aber hier kann es auch Durchdringungen etc geben

Internal Data Model in File\_Processing Komponente (-> Diese Datenstruktur stellt nur Basis Funktionalität zur Verfügung. Unterstützt Algorithmen nicht optimal, aber damit bleiben Datenstruktur und Algorithmen entkoppelt austauschbar)

Baumstruktur von Listen in Behältern Wandcontainer -> findet nächste Wand effizienter

Teste Raumspezifikationen gegen maximale double größe

Packages umbennen in tests -> auch package access code testbar

Persistence als Singleton implementieren?

Aufteilung auf Pakete feiner -> bessere Kapselung?

Abstrakte Fabrik für CandidateSearcher, PositionOptimizer, IlluminationTester (diese sind ja eher als einheit zu betrachten und sollten aufeinander abgestimmt sein?

Limit time has to be larger than 0

Tests anders auswerten?

Soll auch funktionieren wenn Dateien im selben Verzeichnis liegen wie jar liegen!