Anforderungen: KryptoProjekt

# Rahmenbedingungen

* Entwicklungsumgebung: Netbeans 6.8 (wegen einfacher GUI-Erstellung)
* UML-Tool: Visual Paradigm
* Bedienung schwerer gewichtet als Performance
* Versionskontrolle über Git (Hoster: GitHub)
* Tests durch JUnit und Assertions
* Dokumentation der jeweiligen Funktionen
* Aufgaben im Maple-Praktikum-Stil
* GUI evtl. im Maple-Stil
* starke Trennung zwischen GUI und Funktionen (schwache Kohäsion im Code; Model-View-Control)
* später mit Möglichkeit, verschiedene Schwierigkeitsgrade einzubauen
* netzwerkfähig
* Coding Conventions
  + Variablen- und Funktionsnamen in Englisch
  + keine Tabs, Leerzeichen verwenden (4 an der Zahl)
  + Kommentare über die jeweilige Code-Zeile
  + Klassenattribute an Anfang der Klasse, dann public- zuletzt private-Methoden

# Anforderungen

* Hamming-Code
* Zyklische Codes
* Eulersche Phi-Funktion
* Square-and-Multiply
* mathematische Grundlagen (Modulorechnung, euklidischer Algorithmus, GGT)
* zunächst einfache Eingabe von Aufgaben, später Autogenerierung der Aufgaben
* absichtliche Fehlerproduktion bei Verschlüsselung
* Fragekataloggenerierung in Klausurstil (Klausurvorbereitung)
* Messung und Angabe der benötigten Zeit, Anzahl der Schritte der einzelnen Algorithmen
* Quiz (Multiple Choice, You don’t know Jack-Stil)
* Druckfunktion
* Hilfefunktion (Text oder evtl. mit Unterstützung von Flash-Videos)
* später evtl. Anzeige der GUI über Sprachpakete zur Internationalisierung

# Offene Fragen

Sollen die Rechenschritte der Algorithmen explizit angezeigt werden, oder nur Aufgabe 🡪Lösung?