Task 1 – Enigma Emulator: Class Summary

This document briefly describes the main classes used in the Enigma M3 emulator design. It is intended to accompany the UML class diagram and to explain the role of each component in a concise way.

Substitutor

- Abstract base class for all components that perform letter substitution.
- Provides helper methods for converting between letters and indices (A–Z <-> 0–25) and for circular shifts modulo 26.
- Declares abstract methods translate() and reverseTranslate() that subclasses implement according to their specific behaviour.

Translator

- Implements a simple permutation of the alphabet on top of Substitutor.
- Holds forwardPermutation[26] and reversePermutation[26], where the reverse mapping is computed automatically from the forward one.
- translate() uses the forward permutation, reverseTranslate() uses the reverse permutation. This class is reused by reflector, plugboard and rotor.

Reflector

- Specialisation of Translator that models the Enigma reflector.
- Uses a symmetric permutation: if A is mapped to Y, then Y is mapped back to A.
- Because of this symmetry, forward and reverse permutations are identical and the same translate() method can be used in both directions.

Plugboard

- Another Translator-based component that represents the plugboard at the front of the machine.
- It is configured by up to ten letter pairs (e.g., A–T, C–E). These pairs are converted into a full permutation where paired letters swap and all others map to themselves.
- The resulting permutation is then used exactly like in Translator for forward and reverse translation.

Rotor

- Translator subclass that adds the mechanical behaviour of a rotor.
- Stores its internal wiring, ring setting, current offset and a turnover notch.
- translate()/reverseTranslate() take these values into account by shifting the indices before and after applying the permutation.
- Provides step/rotate and isAtNotch methods so the Enigma can advance the rotor positions and detect when neighbouring rotors must also step.

Enigma

- High-level class that represents the complete Enigma M3 machine.
- Aggregates three Rotor objects (right, middle, left), one Reflector and one Plugboard and is responsible for configuring them.
- For each key press, it first steps the rotors (including the double-stepping rule) and then sends the letter through: plugboard -> rotors forward -> reflector -> rotors reverse -> plugboard.
- The final output of this process is the encrypted (or decrypted) letter.