

Übungszettel 8 – Max. 8 Statuspunkte, 4 Bonuspunkte

Abgabe möglich bis: siehe Moodle

Speichern Sie Ihre Lösungen in Ihrem Ordner im Repository ab, bevor Sie sie vorführen und erklären.



Enigma (8 Statuspunkte + 4 Bonuspunkte)

38. Simulieren Sie eine vereinfachte Version der Schlüsselmaschine Enigma (vgl.

https://de.wikipedia.org/wiki/Enigma_%28Maschine%29). Orientieren Sie sich dabei an dem Holzmodell „Enigma III“

(http://www.creativecraffthouse.com/index.php?main_page=product_info&cPath=143&products_id=978).

38.1. Es gibt drei Walzen (5/6/7). Jede Walze hat zwei Seiten (5: 50+51 usw.). Auf jeder Walzensseite ist im Uhrzeigersinn eine Permutation der 26 Buchstaben A-Z zu sehen. Die Walzen sind:

38.1.1. Walzennummer 50: ABCDEFGHIJKLMNOPQRSTUVWXYZ

38.1.2. Walzennummer 51: ADCBEHFGILJKMPNQOTRSUXVWZY

38.1.3. Walzennummer 60: ACEDFHGIKJLNMOQPRTSUWVXZYB

38.1.4. Walzennummer 61: AZXVTRPNDJHFLBYWUSQOMKIGEC

38.1.5. Walzennummer 70: AZYXWVUTSRQPONMLKJIHGFEDCB

38.1.6. Walzennummer 71: AEBCDFJGHIKOLMNPTQRSUYVWXZ

38.2. Deklarieren Sie eine Klasse Walze mit folgenden Eigenschaften:

6 private Konstanten für die einzelnen Buchstabenpermutationen

Privates int-Attribut für die Walzennummer

Constructor, der einen int-Wert erhält und im Walzennummernattribut speichert

38.3. Schreiben Sie eine Methode `countClockwiseRotations()`, die zwei `char`-Parameter erhält. Der erste Parameter ist das Startzeichen und der zweite Parameter ist das Zielzeichen. Geben Sie in der Methode zurück, wie viele Positionen im Uhrzeigersinn das Zielzeichen vom Startzeichen entfernt ist.

38.3.1. Beispiel: Walze 50, Start ‚B‘, Ziel ‚E‘, Rückgabewert 3

38.3.2. Beispiel: Walze 51, Start ‚B‘, Ziel ‚E‘, Rückgabewert 1

38.3.3. Beispiel: Walze 60, Start ‚P‘, Ziel ‚P‘, Rückgabewert 0

38.3.4. Beispiel: Walze 61, Start ‚E‘, Ziel ‚B‘, Rückgabewert 15. Beachten Sie, dass nach dem letzten Zeichen der Permutation wieder beim ersten Zeichen weitergezählt wird. Drehen Sie stets im Uhrzeigersinn. Ihre Rückgabewerte liegen also immer im Intervall `[0..25]`.

38.4. Schreiben Sie eine Methode `countCounterClockwiseRotations()`, die wie `countClockwiseRotations()` funktioniert, aber gegen den Uhrzeigersinn zählt.

38.4.1. Beispiel: Walze 61, Start ‚E‘, Ziel ‚B‘, Rückgabewert 11.

WIN-EPR Einführung in die Programmierung

- 38.5. Schreiben Sie eine Methode `rotateClockwise()`, die einen `char` und einen `int` als Parameter erhält und einen `char` zurückgibt. Der `char`-Parameter gibt das Startzeichen an, der `int`-Wert die Anzahl Rotationen im Uhrzeigersinn. Der Rückgabewert gibt das Zeichen an, das man nach entsprechend vielen Rotationen an gleicher Stelle ablesen kann.
Beispiel: `rotateClockwise('B', 3)` sollte für Walze 50 den Wert `'E'` liefern.
- 38.6. Schreiben Sie eine Methode `rotateCounterClockwise()`, die wie `rotateClockwise()` funktioniert, jedoch gegen den Uhrzeigersinn rotiert.
- 38.7. Eine Enigma hat drei Walzen. Die erste Walze wird im Uhrzeigersinn rotiert, die zweite gegen den Uhrzeigersinn und die dritte wieder im Uhrzeigersinn. Eine Rotation einer Walze führt zur Rotation der übrigen Walzen.
Der Schlüssel legt die Startstellung der Walzen fest. Der Schlüssel ist drei Zeichen lang. Das erste Zeichen des Schlüssels bestimmt die Startposition der ersten Walze, das zweite Zeichen die Startposition der zweiten Walze, und das dritte Zeichen des Schlüssels bestimmt die Startposition der dritten Walze. Die Angabe des Schlüssels in Verbindung mit der Angabe, auf welchem Stab welche Walze liegt, gibt die Konfiguration der Enigma an.
- 38.8. Deklarieren Sie eine Klasse `Enigma` mit folgenden Eigenschaften:
Privates `Walze[]`-Array für die verwendeten Walzen
Privates `String`-Attribut des verwendeten Schlüssels
Constructor, der drei `int`-Werte (für die Walzennummern) und einen `String`-Parameter erhält (für den Schlüssel). Der Constructor soll abhängig von den Walzennummern die entsprechenden Walzen instantiieren.
- 38.9. Schreiben Sie eine Methode `encrypt()` der Klasse `Enigma`, die einen `String` als Parameter erhält und einen `String` zurückgibt.
- 38.9.1. Es werden abwechselnd die erste und die zweite Walze verwendet. Das Ausgabezeichen wird an der dritten Walze abgelesen. Das Startzeichen für die erste Walze ist das erste Zeichen des Schlüssels, das Startzeichen für die zweite Walze das zweite Zeichen des Schlüssels, das Startzeichen für die dritte Walze das dritte Zeichen des Schlüssels.
- 38.9.2. Ermitteln Sie für das erste Zeichen des zu verschlüsselnden Strings, um wie viele Positionen die erste Walze im Uhrzeigersinn rotiert werden muss (ausgehend vom Startzeichen). Ermitteln Sie dann, welches Endzeichen Sie auf der dritten Walze erhalten, wenn Sie ausgehend vom Startzeichen der dritten Walze um die gleiche Anzahl Positionen im Uhrzeigersinn rotieren.
Beispiel: Walzen 51/61/71, Schlüssel „WIN“ (=Startzeichen ‚W‘), Plaintext „HTWG“;
Anzahl Positionen auf Walze 51 von ‚W‘ nach ‚H‘: 8. Auf Walze 71 von ‚N‘ nun ebenfalls 8 Positionen bis Zeichen ‚V‘.
- 38.10. Lesen Sie über die Tastatur die Walzennummern ein und den Schlüssel.
- 38.11. Lesen Sie über die Tastatur den zu verschlüsselnden Text ein („Plaintext“). Nur Großbuchstaben.
- 38.12. Testen Sie Ihr Programm, indem Sie für die Walzen 60, 51, 71 und den Schlüssel „EPR“ die Zeichenfolge „JAICHKANNS“ verschlüsseln. Als verschlüsselten Text („Ciphertext“) sollten Sie „ZFWCYUTQEL“ erhalten.

39. Implementieren Sie die Entschlüsselung eingegebener Texte.

- 39.1. Lesen Sie über die Tastatur die Walzennummern ein und den Schlüssel.
- 39.2. Lesen Sie über die Tastatur den zu entschlüsselnden Text ein („Ciphertext“).
- 39.3. Schreiben Sie eine Methode `decrypt()` der Klasse `Enigma`, die einen `String` als Parameter erhält und einen `String` zurückgibt. Führen Sie die Entschlüsselung als Umkehroperation der Verschlüsselung in `encrypt()` durch. Für die Zeichen des Ciphertexts errechnen Sie die Anzahl Rotationen der dritten Walze ausgehend von deren Startzeichen und lesen abwechselnd an der ersten und zweiten Walze ab, welches Zeichen Sie dort ausgehend vom Startzeichen und der entsprechenden Anzahl Rotationen in richtiger Richtung erhalten.
- 39.4. Testen Sie Ihr Programm, indem Sie für die Walzen 71, 51, 61 und den Schlüssel „BIT“ die Zeichenfolge „SMFAXRIKKG“ entschlüsseln. Ob Sie richtig entschlüsseln, erkennen Sie am Ergebnis.