Projektbeschreibung:

Das Softwarehaus SecureITy ist auf den Bereich Sicherheitssoftware spezialisiert. Das örtliche Jugendforschungszentrum möchte für interessierte Schüler einen Aktionstag zum Thema Verschlüsselung anbieten und fragt die Firma SecureITy diesbeüglich an.
Sie werden als Auszubildender mit der Durchführung des Aktionstages beauftragt.

Sie haben entschieden, ein Python-Programm zur Vigenére-Verschlüsselung zu realisieren, welches zu den symmetrischen Verschlüsselungsverfahren zählt (siehe Anlage 1).

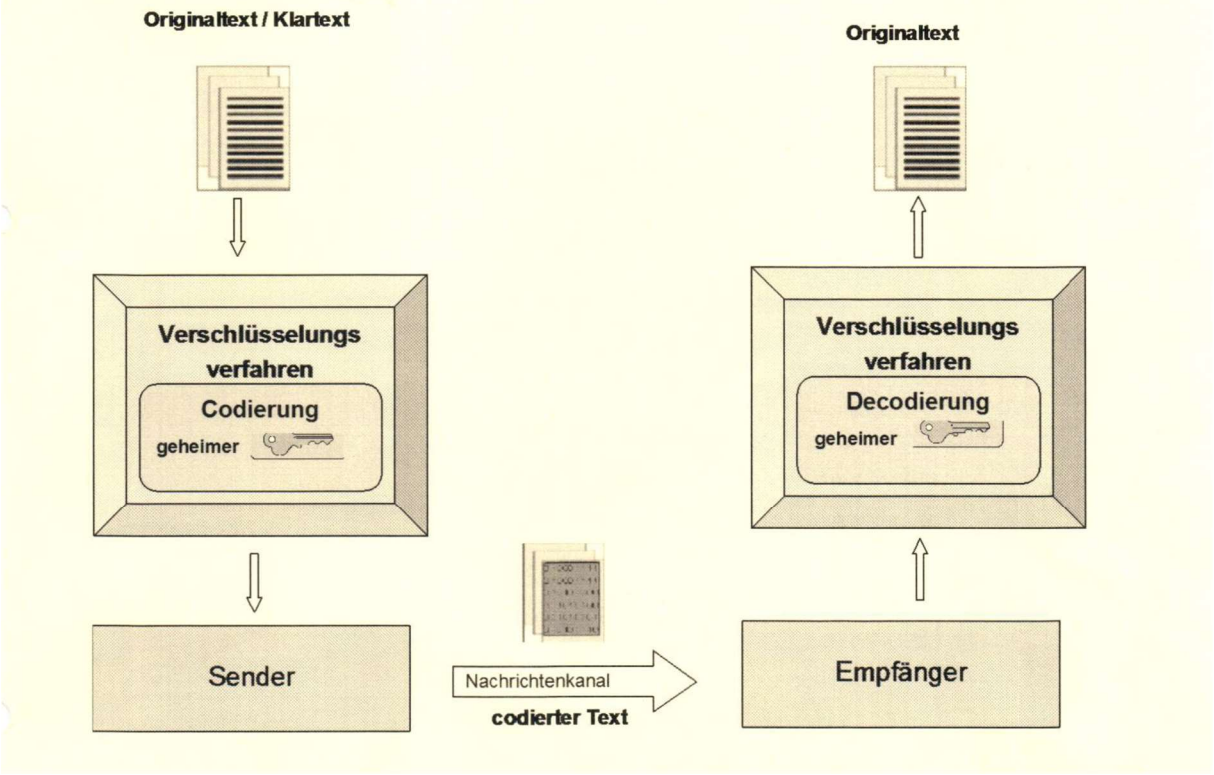
Das Prinzip der Vigenére-Codierung, welches Sie in der Methode codieren() implementieren sollen, wird in Anlage 2 erklärt.

Hinweis: Sie können davon ausgehen, dass im Klartext nur Großbuchstaben, keine Umlaute, keine Sonderzeichen und keine Leerzeichen enthalten sind.

Schreiben Sie ein Hauptprogramm, mit dem Sie die Methode codieren() mit folgenden Daten prüfen können.

klarText: “DERADLERISTGELANDET“
privateKey: “PRUEFUNG“
Zur Überprüfung: der codierte Text lautet: “SVLEIFRXXJNKJFNTSVN“

Ganzheitliche Aufgabe I	Fachinformatiker/-in *Anwendungsentwicklung*
Anlage 1: zu Aufgabe 1, Prinzip symmetrischer Verschlüsselungsverfahren	



Ganzheitliche Aufgabe I	**Fachinformatiker/-in**
Anlage 2: zu Aufgabe 1.1, Prinzip der Vigenère-Codierung	*Anwendungsentwicklung*

Zum Verschlüsseln benötigt sowohl der Sender als auch der Empfänger einen gemeinsamen privaten Schlüssel.
Der Klartext sowie der private Schlüssel wird ohne Leerzeichen, Umlaute, bzw. Sonderzeichen als Großbuchstaben angegeben:
Ist die Zeichenlänge des Schlüssels kleiner als die Zeichenlänge des Klartextes, wird der Schlüssel solange wiederholt bis die Schlüssellänge gleich der Länge des Klartextes ist (siehe unteres Beispiel).

Zeile 1: Klartext: DIESISTEINEBOTSCHAFT
Zeile 2: Schlüssel: GEHEIM

D	I	E	S	I	S	T	E	I	N	E	B	O	T	S	C	H	A	F	T
G	E	H	E	I	M	G	E	H	E	I	M	G	E	H	E	I	M	G	E

Nun wird jedem Buchstaben eine Zahl zugeordnet nach der folgenden Tabelle

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Die Zahl 4 entspricht nach der obigen Tabelle dem Buchstaben E.

`A→0, B→1, C→2,.. Z→25` diese Werte erhält man einfach über den ASCII-Code der Großbuchstaben:
'A' - 'A' = 0; 'B' - 'A' = 1; 'C' - 'A' = 2; ... 'Z' - 'A' = 25;
Die Codierung erfolgt, indem die zugeordneten Werte der einzelnen Buchstaben die übereinander stehen **addiert** werden.

Für den Wert des `i-ten` Buchstabens des codierten Textes ergibt sich die folgende Berechnung:

```
codewert[i] = (klartextwert[i] + keywert[i]) mod 26
```

Die Modulooperation wird verwendet, weil zwei Buchstabenwerte den Wert von 25 überschreiten könnten und so außerhalb des Alphabetes lägen.

Beispiel:

Zeile 1: Klartext: DIESISTEINEBOTSCHAFT
Zeile 2: Schlüssel: GEHEIM
Zeile 3: Der sich ergebende chiffrierter Text

D	I	E	S	I	S	T	E	I	N	E	B	O	T	S	C	H	A	F	T
G	E	H	E	I	M	G	E	H	E	I	M	G	E	H	E	I	M	G	E
J	M	L	W	Q	E	Z	I	P	R	M	N	U	X	Z	G	P	M	L	X

Beispiel (grau markiert): 'S' + 'M' → 18 + 12 = 30 → 30 mod 26 = 4 (entspricht 'E')
`codewert[5] = (18 + 12) mod 26 → codewert[5] = 4.`