

SAE1.01 - Implémentation d'un besoin client

Simulation fonctionnelle d'une machine Enigma M3

Cindy Cappelle
cindy.cappelle@univ-lille.fr

Automne 2021

1 Contexte

AKBAOKETGPVYHGWBSGSVUDTZEENOXGFOBVOJVTWFPIKC ... Non, ce n'est pas une erreur de frappe ! Voici le message que serez capable de déchiffrer à la fin de cette SAE ¹.

La **machine Enigma** fut utilisée par les Allemands pendant la seconde guerre mondiale pour chiffrer des messages secrets. Alan Turing est connu pour avoir fortement contribué à la compréhension du fonctionnement du chiffrement de cette machine.

Dans cette SAE S1.01 (liée à la ressource DEV R1.01), il s'agit d'implémenter une simulation de cette machine en utilisant le langage iJava.

Le travail se fait en **binôme** (**sauf pour les redoublants qui sont en monôme**), binôme à constituer au sein d'un même demi-groupe. Toujours dans le cadre de cette SAE, le présent travail sera suivi d'un contrôle TP qui se fera en monôme pour tous.

Ce document présente tout d'abord le fonctionnement de la machine Enigma, puis propose une décomposition du problème en fonctions simples. Vous devrez alors implémenter ces fonctions en respectant strictement les structures de données et la décomposition proposées. Enfin, les rendus attendus sont listés.

2 Présentation de la machine Enigma M3

2.1 Un peu d'histoire pour commencer ...

Enigma est machine électromécanique inventée par l'ingénieur Allemand Arthur Scherbius en 1923. Elle se présente sous la forme d'une machine à écrire facilement transportable dans une valise (Fig. 1).

Elle va être utilisée par les Allemands et leurs alliés, pendant la seconde guerre mondiale, comme outil de communication chiffrée. Ce sont les renseignements polonais qui ont les premiers essayé de déchiffrer la machine Enigma. Ils travaillent en collaboration avec les Français et les Anglais et réussirent à se procurer une machine Enigma civile. Ils comprirent son mode de fonctionnement, mais les machines militaires étaient plus sophistiquées. Hans-Thilo Schmidt, Allemand ayant des comptes à régler avec sa patrie, vend en 1931 les plans de la version militaire à un agent secret français.

Lors de la seconde guerre mondiale, les Anglais décident d'installer les meilleurs mathématiciens au manoir de Bletchley Park afin de comprendre la machine. Parmi eux, Alan Turing (Fig. 2) va utiliser la machine qu'il a créée, **Colossus**, ancêtre de l'ordinateur moderne, dont la puissance de calcul importante va permettre de déchiffrer les messages jusqu'à la fin de la guerre. Il a été estimé que la guerre a ainsi été écourtée de deux ans. Alan Turing n'en aura la notoriété que de façon posthume !²

¹Pour la configuration suivante de la machine M3 :

- Rotor 1 : III, en position initiale W
- Rotor 2 : I, en position initiale D
- Rotor 3 : V, en position initiale E
- Réflecteur : B
- Câblage : AV - DE - HO - JK - LS - XQ

²En tant qu'informaticien, il serait bon d'en connaître un peu plus sur Alan Turing, considéré comme un des pères des ordinateurs modernes, par exemple en regardant le docudrama "*Codebreaker*" sorti en 2012 pour le centenaire de la naissance de Turing, ou le film "*Imitation Game*" de Morten Tyldum, ou encore la pièce de théâtre "*La machine de Turing*" de Benoit Solès.

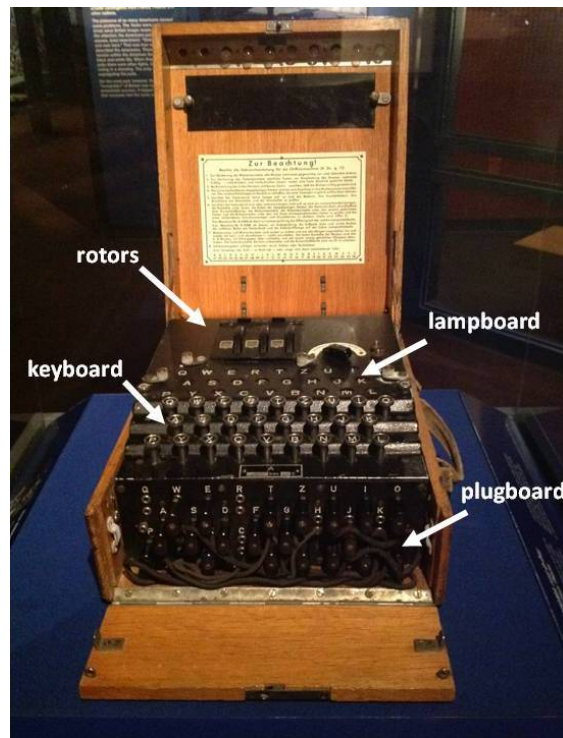


Figure 1: Les composants de la machine Enigma



Figure 2: Alan Turing

2.2 Mode de fonctionnement

Le modèle M3 de la machine Enigma est constituée de (Fig. 1) :

- un clavier
- un tableau lumineux de 26 lettres
- une interface de câblage : 6 câbles qui permettent de créer 6 paires de lettres associées (Fig. 3)
- de 3 rotors (à choisir parmi 5 et numérotés de I à V) (Fig. 4)
- un réflecteur (à choisir parmi 2 : A ou B).

Les modèles évolués possédaient cinq rotors, ce qui augmentait considérablement le nombre de combinaisons différentes, mais nous avons ici choisi un modèle simplifié à 3 rotors.

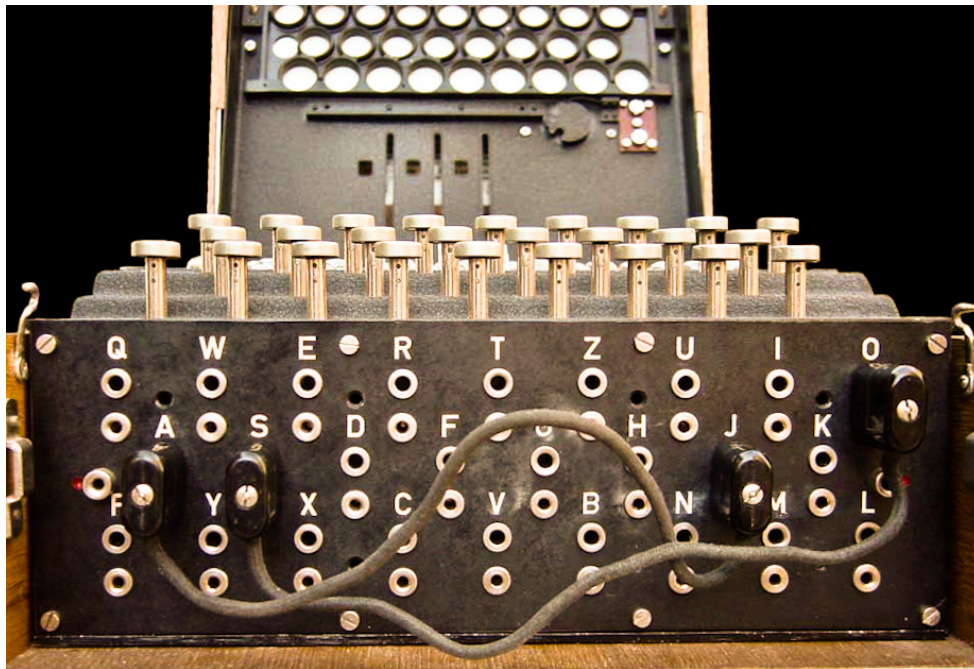


Figure 3: Interface de câblage (ici avec 2 paires uniquement)



Figure 4: Les 5 rotors à disposition

Pour utiliser la machine, elle devait être configurée de la même façon que celle du codeur, à savoir mêmes câblages, mêmes rotors dans la même position et même réflecteur. Cette configuration était transmise par poste radio et changeait toutes les vingt-quatre heures, ce qui compliquait davantage le travail de déchiffrement.

Par exemple, pour déchiffrer le message donné au début de ce sujet, sont utilisés : les rotors historiques **III**, **I**, **V**, dans les positions initiales respectives **W**(22), **D**(3), **E**(4), le réflecteur **B** et les paires de lettres des câbles : **AV-DE-HO-JK-LS-XQ**.

Pour comprendre la situation, le rotor historique III peut être représenté comme suit :

Rotor III

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

Il faut comprendre que la lettre A est chiffrée par B, la lettre B est chiffrée par D, C par F et ainsi de suite (sur la dernière ligne est indiquée la position de la lettre dans l'alphabet (en commençant par 0 : A=0, B=1, ..., Z=25, comme l'usage en est souvent fait en cryptographie).

Si on configure ce **rotor III** en position **22** (c'est-à-dire **W**), on "tourne"/décale les 2 dernières lignes de sorte que le W soit en premier et code le A, le chiffrement est alors comme suit :

Rotor III en position W (22)

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

Voici les tableaux correspondants aux rotors I et V dans la configuration attendue dans l'exemple, ainsi que celui du réflecteur B :

Rotor I en position D (3)

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

Rotor V en position E (4)

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

Réflecteur B

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

Après avoir configuré correctement la machine, voici la démarche à suivre :

- on tape la lettre à déchiffrer sur le clavier ;
- si la lettre est reliée à une autre par un des six câbles, elle change, sinon elle reste identique ;
- passage dans le premier rotor, puis le deuxième rotor, puis le troisième rotor ;
- passage dans le réflecteur : le principe du réflecteur est de permuter les lettres par paires préalablement définies ;
- retour vers le clavier lumineux en passant par le troisième puis le deuxième et enfin le premier rotor en les parcourant "à l'envers" ainsi que par le tableau de connexion.
- mise-à-jour de la position des rotors :
 - le premier rotor tourne d'un cran,
 - le deuxième rotor tourne d'un cran si le premier rotor a effectué un tour complet (26 positions)
 - le troisième rotor tourne d'un cran si le deuxième a effectué un tour complet (toutes les 26×26 lettres codées ou décodées)

On peut illustrer chacune de ces étapes sur le début du message d'exemple en utilisant la configuration de départ (Fig.5) :

- on tape la lettre **A** sur le clavier ;
- la lettre **A** est reliée à la lettre **V** par un câble ;
- passage dans le premier rotor (en position **W-22**) : **V** devient **Z** ;
- passage dans le deuxième rotor (en position **D-3**) : **Z** devient **G** ;
- passage dans le troisième rotor (en position **E-4**) : **G** devient **R** ;
- passage dans le réflecteur (**B**) : **R** devient **A** ;
- passage inverse dans le troisième rotor : **A** devient **T** ;
- passage inverse dans le deuxième rotor : **T** devient **F** ;
- passage inverse dans le premier rotor : **F** devient **L** ;
- la lettre **L** est reliée à la lettre **S** par un câble : la lettre **S** s'affiche sur le clavier lumineux.

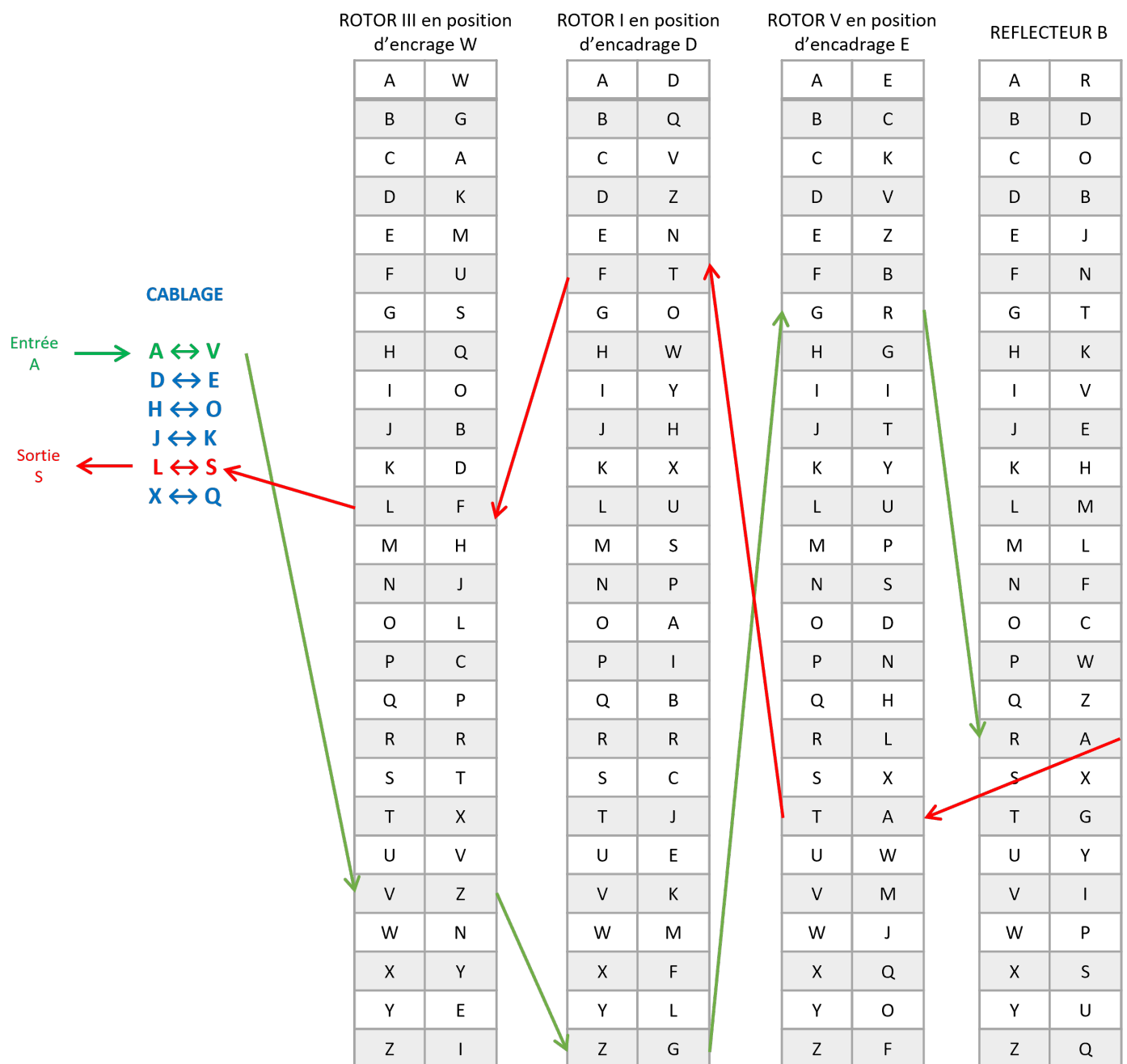


Figure 5

Avant de traiter la deuxième lettre du message, il faut mettre à jour les positions des rotors (le premier rotor avance après chaque lettre, le deuxième et le troisième avancent respectivement d'un cran quand le premier et le deuxième ont fait un tour complet). Ici, il faut donc faire tourner le rotor III d'un cran, il se retrouve alors en position G (Fig. 6).

On peut alors décoder la deuxième lettre du message donné en introduction, à savoir le K :

- on tape la lettre **K** sur le clavier ;
- la lettre **K** est reliée à la lettre **J** par un câble ;
- passage dans le premier rotor (désormais en position **G-6**) : **J** devient **D** ;
- passage dans le deuxième rotor (en position **D-3**) : **D** devient **Z** ;
- passage dans le troisième rotor (en position **E-4**) : **Z** devient **F** ;
- passage dans le réflecteur (**B**) : **F** devient **N** ;
- passage inverse dans le troisième rotor : **N** devient **P** ;
- passage inverse dans le deuxième rotor : **P** devient **N** ;
- passage inverse dans le premier rotor : **N** devient **V** ;
- la lettre **V** est reliée à la lettre **A** par un câble : la lettre **A** s'affiche sur le clavier lumineux.

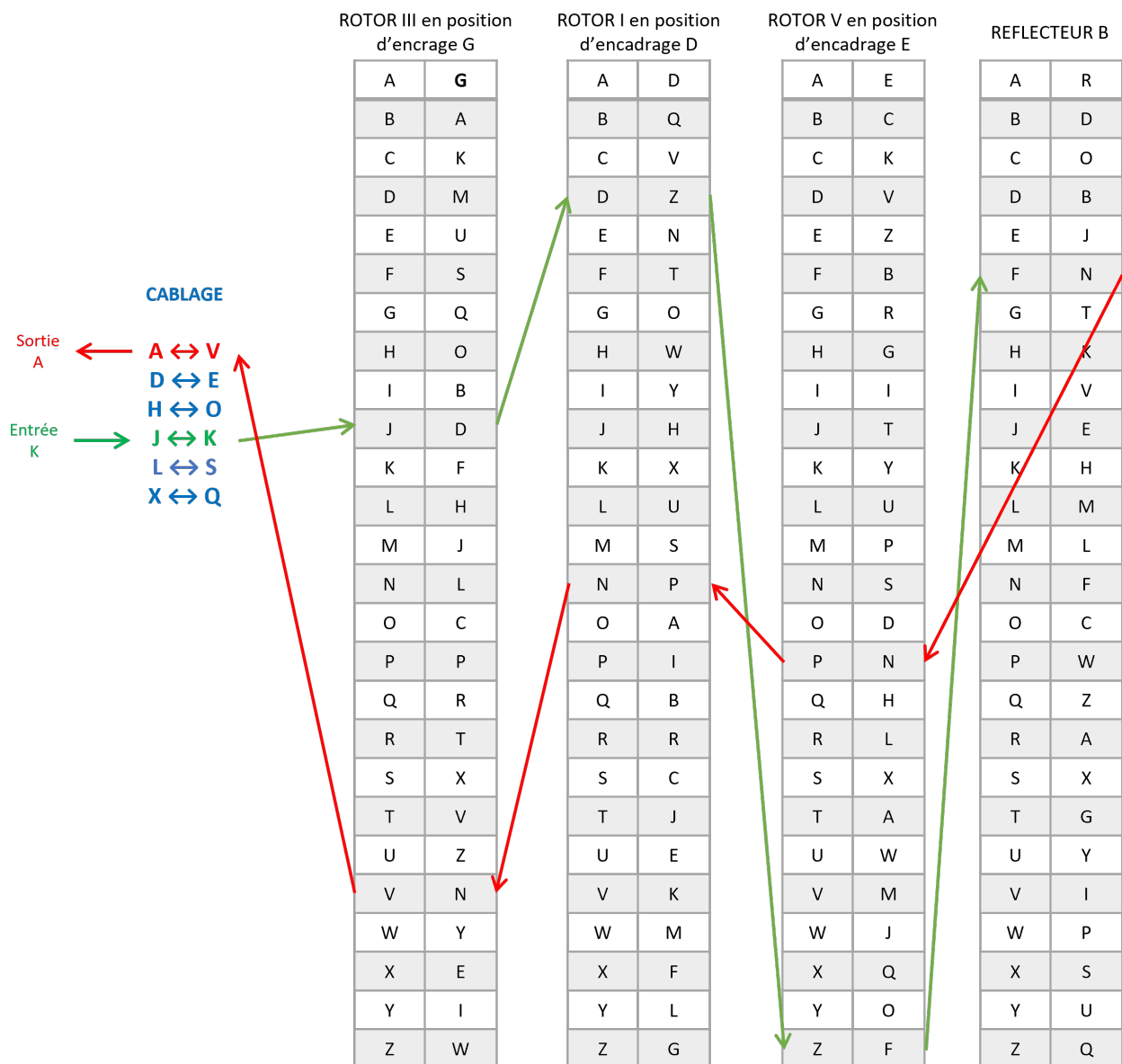


Figure 6

3 Implémentation d'une simulation de la machine Enigma en iJava

Après avoir expliqué et analysé le mode de fonctionnement de la machine Enigma, il s'agit maintenant d'en proposer une simulation en iJava. Le principe est d'entrer la configuration de la machine désirée, le message chiffré et le programme calcule le message déchiffré. On suppose que l'analyste de l'entreprise a déjà travaillé sur l'analyse et la décomposition du problème en fonctions. Votre rôle est alors :

- d'implémenter les fonctions demandées
- de tester les fonctions demandées
- dans le programme principal, il s'agira de faire un menu permettant de :
 1. tester manuellement chacune des fonctions
 2. simuler le codage/décodage d'un message selon un paramétrage de la machine choisi par l'utilisateur
 3. simuler le codage/décodage d'un message pré-défini selon un paramétrage pré-configuré (celui de l'exemple proposé en introduction du sujet)

Les fonctions à développer sont listées ci-après ainsi que les étapes du programme principal (ce fichier à compléter est fourni sur Moodle).

```
class enigma extends Program {

    // CONSTANTES DE LA MACHINE
    String ALPHABET = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
    String ROTOR1 = "EKMFLGDQVZNTOWYHXUSPAIBRCJ";
    String ROTOR2 = "AJDKSIRUXBLHWTMCQGZNPYFVOE";
    String ROTOR3 = "BDFHJLCPRTXVZNYEIWGAKMUSQO";
    String ROTOR4 = "ESOV郑JAYQUIRHXNLFTGKDCMWB";
    String ROTOR5 = "VZBRGITYUPSDNHLXAWMJQOFECK";
    String REFLECTEURA = "YRUHQSLDPXNGOKMIEBFZCWVJAT";
    String REFLECTEURB = "RDOBJNTKVEHMLFCWZAXGYIPSUQ";

    // -----
    // Fonction qui retourne un entier correspondant à la position d'une lettre donnée dans l'alphabet
    // (A=0, B=1, C=2, ...)
    // Exemples :
    // entrée : 'A' -> sortie 0
    // entrée : 'B' -> sortie 1
    // ...
    // entrée : 'Z' -> sortie 25
    int lettreEnNombre(char lettre){
        // A MODIFIER
        return 0;
    }
    // -----
    // Fonction qui retourne la lettre associée à une position (un entier) dans l'alphabet (0=A, 1=B,
    // 2=C, ...)
    // Exemples :
    // entrée : 0 -> sortie : 'A'
    // entrée : 1 -> sortie : 'B'
    // ...
    // entrée : 25 -> sortie : 'Z'
    char nombreEnLettre(int nombre){
        // A MODIFIER
        return 'a';
    }
    // -----
    // Fonction qui permet de sélectionner un rotor
    // A partir d'un entier (entre 1 et 5) passé en paramètre, cette fonction retourne le rotor correspondant
    // (la chaîne de caractère correspondante)
    // Exemples :
    // entrée : 1 -> sortie : "EKMFLGDQVZNTOWYHXUSPAIBRCJ" (ROTOR1)
    // entrée : 4 -> sortie : "ESOV郑JAYQUIRHXNLFTGKDCMWB" (ROTOR4)
    // ...
}
```

```

String choixRotor(int numeroRotor){
    // A MODIFIER
    return "";
}
// -----
// Fonction qui permet à l'utilisateur de sélectionner le réflecteur
// A partir d'une lettre ('A' ou 'B') passée en paramètre, cette fonction retourne le réflecteur
correspondant (la chaîne de caractère)
// Exemples :
// entrée : 'A' -> sortie : "YRUHQSLDPXNGOKMIEBFZCWVJAT" (REFLECTEURA)
// entrée : 'B' -> sortie : "RDOBJNTKVEHMLFCWZAXGYIPSUQ" (REFLECTEURB)
String choixReflecteur(char lettreReflecteur){
    // A MODIFIER
    return "";
}
// -----
// Fonction qui permet à l'utilisateur de la machine de brancher les câbles reliant les paires (6)
de lettres
// Cette fonction doit retourner une chaîne de caractères de 6 lettres majuscules saisies au clavier
par l'utilisateur (on supposera que ces 6 lettres sont distinctes)
// Exemple :
// Si l'utilisateur saisit les 6 paires suivantes : AV puis DE puis HO puis JK puis LS puis XQ,
la fonction doit retourner "AVDEHOJKLSXQ"
String cablageInitial(){
    // A MODIFIER
    return "";
}
void affichageCablageInitial(){
    println(cablageInitial());
}
// -----
// Fonction qui permet de décaler le rotor d'un rang vers la gauche
// A partir d'une chaîne de caractères passée en paramètre, cette fonction retourne la chaîne de
caractères décalée d'un cran vers la gauche, c'est-à-dire que la première lettre est déplacée à la fin
de la chaîne.
// Exemples :
// entrée : "ABCDEFGHGIJKLMNOPQRSTUVWXYZ" -> sortie : "BCDEFGHGIJKLMNOPQRSTUVWXYZA"
// entrée : "IFHUQSMVDVHNQOIVHZ" -> sortie : "FHUQSMVDVHNQOIVHZI"
String decalageUnRang(String rotor){
    // A MODIFIER
    return "";
}
// -----
// Fonction qui retourne le rotor après avoir défini sa position initiale, c'est-à-dire après nb
décalages.
// A partir d'un rotor donné (une chaîne de caractères) et d'un entier nb donné, cette fonction
retourne le rotor décalé de nb crans vers la gauche
// Exemples :
// entrées : "ABCDEFGHGIJKLMNOPQRSTUVWXYZ" et 3 -> sortie : "DEFGHGIJKLMNOPQRSTUVWXYZABC"
// entrées : "IFHUQSMVDVHNQOIVHZ" et 5 -> sortie : "SMDVHNQOIVHZIFHUQ"
String positionInitialeRotor(String rotor, int position){
    // A MODIFIER
    return "";
}
// -----
// Fonction qui recherche une lettre dans une chaîne de caractères
// A partir d'une lettre donnée et d'une chaîne de caractère donnée, cette fonction retourne l'indice
(la position) de la lettre dans la chaîne (-1 si absent)
// Exemples :
// entrées 'C' et "ABCDE" -> sortie 2
// entrées 'A' et "ABCDE" -> sortie 0
// entrées 'E' et "ABCDE" -> sortie 4
// entrées 'F' et "ABCDE" -> sortie -1
// entrées 'F' et ROTOR1 -> sortie 3
int indiceLettre(char lettre, String cablage){
    // A MODIFIER
    return 0;
}

```



```

}
// -----
// Fonction qui permet de vérifier si la lettre à décoder est reliée par un câble à une autre lettre.
// Si oui, elle est transformée en cette lettre, sinon elle reste identique.
// A partir d'une lettre donnée et d'un cablage donné (une chaîne de 12 caractères), cette fonction
retourne la lettre transformée si elle fait partie d'une paire de lettres d'un des 6 câbles, la même lettre
sinon.
// Exemples :
// entrées : 'H' et "AVDEHOJKLSXQ" -> sortie 'O' (car 3ème paire HO)
// entrées : 'A' et "ABCDEFGHijkl" -> sortie 'B' (car 1ère paire AB)
// entrées : 'B' et "ABCDEFGHijkl" -> sortie 'A' (car 1ère paire AB)
// entrées : 'K' et "ABCDEFGHijkl" -> sortie 'L' (car 6ème paire KL)
// entrées : 'D' et "ABCDEFGHijkl" -> sortie 'C' (car 2ème paire CD)
// entrées : 'M' et "ABCDEFGHijkl" -> sortie 'M' (car M absent de la chaîne du cablage)
char valeurApresCablageDeDepart(char lettre, String cablage){
    // A MODIFIER
    return 'a';
}
// -----
// Fonction qui retourne la nouvelle valeur après le passage dans un rotor
// A partir d'une lettre donnée et d'un rotor donné (une chaîne de caractères), cette fonction retourne
la lettre correspondante à la lettre passée en paramètre après passage dans le rotor donné.
// Exemples :
// entrées : 'A' et ROTOR1 -> sortie : 'E'
// entrées : 'B' et ROTOR1 -> sortie : 'K'
// entrées : 'Z' et ROTOR1 -> sortie : 'J'
// entrées : 'E' et "AJDKSIRUXBLHWTMCQGZNPYFVOE" -> sortie : S
char passageDansUnRotor(char lettre, String rotor){
    // A MODIFIER
    return 'a';
}
// -----
// Fonction qui retourne la nouvelle valeur après le passage dans le réflecteur
// A partir d'une lettre donnée et d'un réflecteur donné (une chaîne de caractères), cette fonction
retourne la lettre correspondante à la lettre passée en paramètre après passage dans le réflecteur.
// Exemples :
// entrées : 'A' et REFLECTEURA -> sortie : 'Y'
// entrées : 'B' et REFLECTEURA -> sortie : 'R'
// entrées : 'Z' et "YRUHQSLDPXNGOKMIEBFZCWVJAT" -> sortie : 'T'
char passageDansLeReflecteur(char lettre, String reflecteur){
    // A MODIFIER
    return 'a';
}
// -----
// Fonction qui retourne la nouvelle valeur après le passage dans un rotor dans le sens inverse
(pour le retour)
// A partir d'une lettre donnée et d'un rotor donné (une chaîne de caractères), cette fonction retourne
la lettre correspondante à la lettre passée en paramètre après passage en sens inverse dans le rotor.
// Exemples :
// entrées : 'E' et ROTOR1 -> sortie : 'A'
// entrées : 'K' et ROTOR1 -> sortie : 'B'
// entrées : 'J' et ROTOR1 -> sortie : 'Z'
// entrées : 'S' et "AJDKSIRUXBLHWTMCQGZNPYFVOE" -> sortie : E
char inverseRotor(char lettre, String rotor){
    // A MODIFIER
    return 'a';
}
// -----
// Fonction qui transforme une chaîne de caractères en majuscule
String enMajuscule(String message){
    // A MODIFIER
    return "";
}
// -----
// PROGRAMME PRINCIPAL
void algorithm(){

```

```

println("\u2500-----\n|\u2500Simulation\u2500d'une\u2500machine\u2500Enigma

// Initialisation des \u00e9l\u00e9ments
// -----
println("Quel\u2500type\u2500de\u2500configuration\u2500souhaitez-vous\u2500?\u2500\n\u25001\u2500:\u2500Configuration\u2500par\u2500d\u00e9f
int choix = readInt();
while (choix<1 || choix>2){
    println("Erreur\u2500de\u2500saisie,\u2500choix\u2500:\u25001\u2500ou\u25002");
    choix = readInt();
}

int rotor1,rotor2,rotor3;
String R1,R2,R3;
char position1,position2,position3;
int decalage1,decalage2,decalage3;
char choixRef;
String refl;
String cables;

String message="";

if (choix == 1){// Configuration pr\u00e9-\u00e9tablie
    // // Rotors choisis
    rotor1 = 3;
    rotor2 = 1;
    rotor3 = 5;
    R1=choixRotor(rotor1);
    R2=choixRotor(rotor2);
    R3=choixRotor(rotor3);

    // // Position initiale des rotors choisis
    position1 = 'W';
    position2 = 'D';
    position3 = 'E';
    decalage1 = indiceLettre(position1,R1);
    R1 = positionInitialeRotor(R1,decalage1);
    decalage2 = indiceLettre(position2,R2);
    R2 = positionInitialeRotor(R2,decalage2);
    decalage3 = indiceLettre(position3,R3);
    R3 = positionInitialeRotor(R3,decalage3);

    // // R\u00e9flecteur choisi
    choixRef = 'B';
    refl = choixReflecteur(choixRef);

    // // Initialisation de la configuration du cablage de la machine par l'utilisateur
    cables = "AVDEHOJKLSXQ";

    // // Message \u00e0 tester
    message = "AKBAOKETGPVYHGWBSGSVUDTZEBNQXGFOBVOJVTWFPKIC";
}
else{// Configuration choisie par l'utilisateur
    // // Choix des 3 rotors (distincts) parmi les 5
    // NB : On supposera que les num\u00e9ros des 3 rotors sont bien compris entre 1 et 5 et qu'ils
    sont tous diff\u00e9rents
    println("Entrez\u2500le\u2500num\u00e9ro\u2500du\u2500premier\u2500rotor\u2500choisi\u2500(1,\u25002,\u25003,\u25004,\u25005)");
    rotor1 = readInt();
    println("Entrez\u2500le\u2500num\u00e9ro\u2500du\u2500deuxi\u00e8me\u2500rotor\u2500choisi");
    rotor2 = readInt();
    println("Entrez\u2500le\u2500num\u00e9ro\u2500du\u2500troisi\u00e8me\u2500rotor\u2500choisi");
    rotor3 = readInt();

    R1=choixRotor(rotor1);
    R2=choixRotor(rotor2);

```

```

R3=choixRotor(rotor3);

// // Choix de la position initiale des rotors choisis
println("Entrez la position initiale du premier rotor choisi (A à Z)");
position1 = readChar();
println("Entrez la position initiale du deuxième rotor choisi (A à Z)");
position2 = readChar();
println("Entrez la position initiale du troisième rotor choisi (A à Z)");
position3 = readChar();

decalage1 = indiceLettre(position1,R1);
R1 = positionInitialeRotor(R1,decalage1);
decalage2 = indiceLettre(position2,R2);
R2 = positionInitialeRotor(R2,decalage2);
decalage3 = indiceLettre(position3,R3);
R3 = positionInitialeRotor(R3,decalage3);

// // Choix du réflecteur parmi les 2
println("Entrez la lettre du réflecteur choisi (A ou B)");
choixRef = readChar();
refl = choixReflecteur(choixRef);

// // Initialisation de la configuration du câblage de la machine par l'utilisateur
cables = cablageInitial();

// // Le message à coder
println("Entrez le message à coder :");
message = readString();
message = enMajuscule(message);
}

String messageDecode="";

// Boucle principale du programme Enigma
// -----
for (int tour=0 ; tour < length(message) ; tour=tour+1){// la boucle s'arrête quand
on a codé chaque lettre du message
    // 1. Récupération de la lettre courante dans le message à décoder
    // A COMPLETER => charAt
    // 2. Passage par le câblage
    // A COMPLETER => valeurApresCablageDeDepart
    // 3. Passage par les 3 rotors (premier, deuxième, troisième)
    // A COMPLETER => passageDansUnRotor
    // 4. Passage par le réflecteur
    // A COMPLETER => passageDansLeReflecteur
    // 5. Passage par les 3 rotors (dans le sens inverse : troisième, deuxième, premier)
    // A COMPLETER => inverseRotor
    // 6. Passage par le câblage
    // A COMPLETER => valeurApresCablageDeDepart

    // 7. Ajout de la lettre décodée au message
    // A COMPLETER => concaténation

    // 8. Préparation de l'itération suivante : décalage du premier rotor (à chaque fois);
    // décalage du deuxième rotor si le premier a fait un tour complet (après 26 itérations) ; décalage du troisième
    // rotor si le deuxième a fait un tour complet (après 26*26 itérations)
    // 8.1 Le rotor 1 tourne d'un rang vers la gauche après chaque lettre (donc à chaque tour)
    // A COMPLETER => decalageUnRang sur R1
    // 8.2 Si le rotor 1 a effectué un tour (toutes les 26 itérations) alors le rotor 2 tourne
    // d'un cran vers la gauche
    // A COMPLETER
    // 8.3 Si le rotor 2 a effectué un tour (toutes les 26*26 itérations) alors le rotor 3
    // tourne d'un cran vers la gauche
    // A COMPLETER
}

```

```

    println("Le message décodé est : \n" + messageDecode);
}
}

```

4 Calendrier des rendus intermédiaires et livrables

- constitution des binômes
 - au plus tard le **vendredi 8 octobre 2021 à 11h59** : renseignez la constitution de votre binôme en complétant le fichier suivant : <https://nextcloud.univ-lille.fr/index.php/s/3X7MkEjypWpXM2q>
 - **Attention, les binômes doivent être constitués au sein d'un même groupe.**
 - Exception pour les **redoublants** : vous travaillerez **seul** (indiquez alors **RDB** dans la colonne NOM ETUDIANT 2).
- dépôts intermédiaires hebdomadaires : il s'agit de déposer sur Moodle (page de la SAE1.01) l'état d'avancement de votre code c'est-à-dire un fichier nommé **enigma_NOM1_NOM2.java** (ou **enigma_NOM1_RDB.java** pour les redoublants)
 - dépôt intermédiaire 1 :
 - * date limite : **lundi 18 octobre à 8h**
 - * a minima les fonctions suivantes sont attendues : `lettreEnNombre`, `nombreEnLettre`
 - dépôt intermédiaire 2 :
 - * date limite : **lundi 25 octobre à 8h**
 - * a minima les fonctions suivantes sont attendues : `lettreEnNombre`, `nombreEnLettre`, `choixRotor`, `choixReflecteur`, `cablageInitial`, `decalageUnRang`, `positionInitialeRotor`, `indiceLettre`
 - dépôt intermédiaire 3 :
 - * date limite : **lundi 8 novembre à 8h**
 - * a minima les fonctions suivantes sont attendues : `lettreEnNombre`, `nombreEnLettre`, `choixRotor`, `choixReflecteur`, `cablageInitial`, `decalageUnRang`, `positionInitialeRotor`, `indiceLettre`, `valeurApresCablageDeDepart`, `passageDansUnRotor`, `passageDansLeReflecteur`, `inverseRotor`
- rendu final : dépôt sur Moodle d'une archive nommée **enigma_NOM1_NOM2.tar.gz** (ou **enigma_NOM1_RDB.tar.gz** pour les redoublants)
 - date limite : **lundi 15 novembre à 10h**
 - l'archive devra comprendre :
 - * le code source complet (toutes les fonctions et le programme principal),
 - * un document illustrant tous les tests que vous avez effectués pour vérifier vos fonctions, c'est-à-dire les exemples sur lesquels les fonctions ont été testées et les résultats obtenus.
 - * un fichier **readme.txt** (ou **readme.md**) indiquant le binôme, comment compiler et exécuter le fichier, toute autre information jugée utile,
 - * une (ou des) capture(s) d'écran ou illustrations(s) illustrant votre projet (en vue de la construction de votre e-portfolio),
 - * une courte vidéo de la démonstration de votre projet (en vue de la construction de votre e-portfolio).
- contrôle TP : les compétences que vous avez développées pour répondre au cahier des charges du présent sujet (simulation d'une machine enigma simplifiée), seront à réemployer pour l'implémentation d'un autre demande "client"
 - à réaliser seul (pour tous)
 - date : jeudi 18 Novembre
 - durée : 1h30