



Institut Universitaire  
de Technologie  
Aix-Marseille Université



Organisation, Systèmes d'Informations, Conseil

**Institut Universitaire de Technologie,  
Aix-Marseille Université**

**RAPPORT DE STAGE  
Diplôme Universitaire de Technologie  
Spécialité Réseaux et Télécommunications**

Réalisation d'une interface graphique pour un  
client avec Java et Swing

**Michella MITSAKIS**

**OSIC**

Responsable entreprise : Georges Chardenot

Responsable académique : Ivan Madjarov

**2017**



## **Remerciements**

Tout d'abord, je tiens à remercier vivement mon maître de stage Monsieur Georges CHARDENOT, dirigeant de l'entreprise OSIC, pour son accueil et pour m'avoir accordé des missions valorisantes durant ce stage. Je remercie également Jean-Marc CORTEGGIANI de m'avoir encadré durant toute la durée de mon projet et de m'avoir permis d'effectuer et d'aboutir à la finalisation de mon projet.

Effectuer mon stage au sein de cette entreprise a été bénéfique pour moi d'un point de vue pédagogique et empirique.

Bien évidemment, je remercie également l'ensemble des intervenants de ma formation qui m'ont apporté une grande aide concernant la partie théorique. Je remercie particulièrement mon tuteur professionnel Monsieur Ivan MADJAROV pour ses précieux conseils et son aide.



## Table des matières

1	Introduction.....	7
2	Présentation de l'entreprise OSIC.....	8
2.1	L'entreprise OSIC.....	8
2.1.1	Prestations de OSIC.....	8
2.1.2	OSIC et le secteur bancaire.....	9
2.1.3	Exemples de solutions proposées par OSIC.....	9
2.2	Organisation interne de l'entreprise.....	10
2.2.1	Descriptions des moyens humains.....	10
2.2.2	Description des moyens matériels.....	10
2.2.3	Organigramme de OSIC.....	10
3	Présentation du stage.....	11
3.1	Notions élémentaires sur Java et Alchemy.....	11
3.1.1	Définition de la Programmation Orientée Objet.....	11
3.1.2	Définition de Java et Swing.....	12
3.1.3	A propos de Alchemy.....	13
3.2	Objectifs du stage.....	14
3.3	Missions effectuées.....	17
3.3.1	Installation des outils de développement Java.....	17
3.3.2	Réalisation de l'interface Swing.....	20
3.3.3	Création d'un exécutable .jar.....	22
3.4	Bilan personnel et expérientiel.....	24
3.4.1	Difficultés rencontrées et solutions apportées.....	24
3.4.2	Compétences acquises.....	26
4	Conclusion.....	27
5	Glossaire.....	28
6	Bibliographie.....	30
7	Annexes.....	31



## 1 Introduction

Dans le cadre de ma deuxième année de DUT Réseaux et Télécommunications, j'ai pu effectuer un stage de dix semaines au sein de l'entreprise OSIC située à Marseille. Il s'agit d'une entreprise de service permettant la gestion électronique de documents telle que la numérisation, le classement automatisé ou bien l'archivage. Celle-ci est aussi composée d'un service informatique où j'ai pu y jouer un rôle.

Grâce à ce stage, j'ai pu découvrir pour la première fois le monde professionnel et ainsi acquérir de nouvelles compétences pratiques. En effet, cela m'a été utile puisque je recherchais un stage dans le domaine de l'informatique avant de poursuivre mes études en école d'ingénieur informatique.

Les missions proposées par l'entreprise correspondaient à mes attentes et étaient en adéquation avec ma formation en Réseaux et Télécommunications. C'est la raison pour laquelle j'ai décidé d'intégrer cette société. En effet, ces missions ont été effectuées de manière autonome mais dans laquelle j'ai pu contacter mon tuteur pédagogique lorsque j'ai pu rencontrer de réelles difficultés.

L'objectif principal reposait sur la réalisation d'une interface dynamique permettant de consulter l'historique des documents archivés. Etant donné l'apparition de nouvelles attentes de la clientèle, l'entreprise a été amenée à se demander dans quelles mesures une interface dynamique Java peut-elle être réalisée et accessible facilement à un client ?

Dans un premier temps, nous nous concentrerons sur les points intéressants à savoir à propos de l'entreprise et son organisation. Ensuite nous analyserons les différentes missions effectuées durant ce stage. Et enfin nous observerons ce que ce stage a pu m'apporter.

## **2 Présentation de l'entreprise OSIC**

### **2.1 L'entreprise OSIC**

La Société OSIC est une société de conseil spécialisée en gestion électronique de documents (GED) créée le 3 novembre 1997. Elle se compose d'ingénieurs et de chef de projet expérimentés dans leur domaine respectif. Ces derniers maîtrisent les méthodes d'organisation standard du marché.

OSIC possède une expérience de dix ans dans le domaine de la GED qui s'appuie sur des interventions dans des secteurs d'activités tels que, les banques, les établissements publics et collectivités territoriales, l'industrie chimique et mécanique.

Les domaines d'expertise présente au sein de la société sont :

- L'audit et le conseil, l'audit de projets
- L'audit de conformité à la norme NF Z42-013\* des systèmes d'archivage électronique
- Le conseil pour la mise en œuvre de systèmes de gestion de documents
- L'assistance à la maîtrise d'ouvrage pour les projets de gestion de documents
- L'assistance à la maîtrise d'œuvre et gestion de projets
- La définition de stratégies d'archivage électronique
- La prestation de service dans la numérisation de documents

En 1994 l'entreprise n'était qu'éditeur de solution de GED pour le monde bancaire pour ensuite évoluer en 1997 vers le métier d'intégrateur, reposant sur :

- L'étude préalable sans préjugés de la solution
- Le choix de solutions adaptées à la demande
- La traduction et le paramétrage des produits en langues françaises pour les produits étrangers
- La qualification des nouvelles versions disponibles
- La maintenance des sites installés

#### **2.1.1 Prestations de OSIC**

L'entreprise propose différentes prestations telles que la formation adaptée pour le personnel, l'organisation de séminaires au sujet de la technologie utilisée et les normes applicables, la maintenance de l'ensemble des matériels et logiciels installés, en collaboration avec les différents fournisseurs qualifiés et le conseil. Ce conseil assure une étude préalable permettant d'analyser les caractéristiques d'un système telles que les volumes, les flux ou la confidentialité.

Voici comment se déroule une prestation de OSIC :

1. Diagnostic dans les locaux du client
2. Définition des besoins
3. Proposition de solutions personnalisées
4. Mise en place d'un plan d'action
5. Contrôle à chaque phase d'exécution
6. Suivi du client

Par exemple, la société OSIC met en place des solutions pour la généralisation de la GED dans le domaine des dossiers clients. Elle numérise et archive toute la documentation des sites réhabilités (classeurs, rapports techniques, plans, etc.). Elle gère également la complétude des dossiers en relançant des sous-traitants afin d'obtenir des documents manquants.

L'entreprise utilise sa propre méthodologie afin d'offrir une version complète des documents. Cette méthodologie consiste à analyser, organiser, diffuser, optimiser, normaliser et enfin archiver.

Tout d'abord, OSIC procède à une étude approfondie des besoins du client afin de leur proposer une solution, numérise et indexe l'ensemble des documents client quels que soient leurs volumes, leurs types, leurs origines, sans jamais les modifier. De plus, la société facilite la circulation de l'information au sein de l'entreprise demandeuse et s'insère dans leur système d'information sans remise en cause. Enfin, OSIC veille au respect des normes en vigueur et met en place un système pérenne d'archivage centralisé (sur un serveur) ou bien réparti (sur un CD-ROM).

### **2.1.2 OSIC et le secteur bancaire**

OSIC permet aux banques le déploiement de solutions de Gestion Electronique de Document adaptées à leur besoin. Elle intervient aussi dans la dématérialisation des dossiers clients qui sont alors consultables en réseau et dans l'archivage électronique conforme à la réglementation.

Concernant les enjeux du secteur bancaire, la société possède un fort environnement concurrentiel. Cela pousse chaque acteur du marché à s'inscrire dans un processus d'amélioration constante de la qualité et du service apportés aux clients.

Le secteur bancaire possède une réglementation de normes aussi bien nationales qu'internationales. Les banques sont soumises à des audits et doivent être en mesure de fournir sans délai les données ou documents requis.

### **2.1.3 Exemples de solutions proposées par OSIC**

La société répond à des solutions efficaces pour ses clients telles que la constitution d'un dossier client électronique afin de fournir aux collaborateurs une base documentaire commune, complète et accessible à l'ensemble des informations.

Elle propose également la numérisation et le classement des documents commerciaux, juridiques ou des biens comptables. Elle propose aussi comme solution, la mise en œuvre d'un système d'archivage électronique conforme à la réglementation imposée, tout en prenant en compte le cycle de vie des documents.

## 2.2 Organisation interne de l'entreprise

### 2.2.1 Descriptions des moyens humains

La société OSIC est une PME (Petit et Moyenne Entreprise) composée de moins de dix salariés. En effet, selon les années l'effectif des employés de la société a quelque peu varié comme nous pouvons le constater dans le tableau ci-dessous :

Année	Nombre de salariés	Nombres de cadres
2017	4	2
2016	6	2
2015	9	4
2014	7	2
2013	7	2

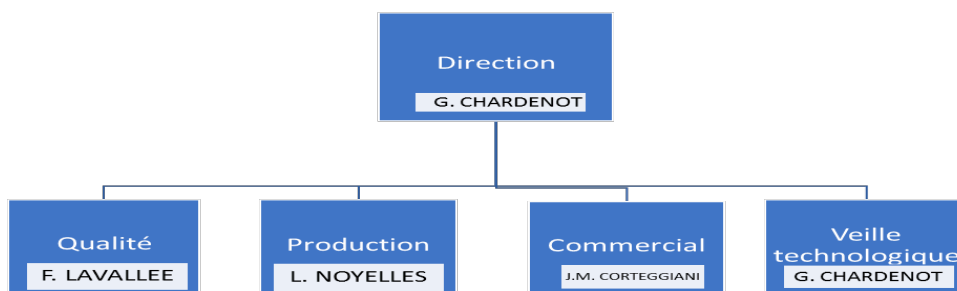
En 2017, les deux personnes chargées de l'encadrement de l'entreprise sont Monsieur CHARDENOT Georges qui est notamment le directeur technique possédant 18 ans d'expérience dans la direction technique et Monsieur CORTEGGIANI Jean-Marc possédant 15 ans d'expérience dans la responsabilité commerciale.

### 2.2.2 Description des moyens matériels

Concernant le matériel de l'entreprise, elle est constituée de :

- Deux serveurs d'entreprise
- Douze ordinateurs fixes et portables
- Un vidéo projecteur
- Six numériseurs A4/A3, plans, négatifs et diapositives
- Un robot de gravure CD et DVD
- Deux numériseurs de livres reliés

### 2.2.3 Organigramme de OSIC



### 3 Présentation du stage

#### 3.1 Notions élémentaires sur Java et Alchemy

##### 3.1.1 Définition de la Programmation Orientée Objet

La programmation orientée objet est une technique d'organisation du code d'un programme en le regroupant en ensemble d'objets. Cette technique permet de regrouper des éléments contenant des informations et qui sont elles-mêmes associées à des actions communes. On appelle objet, l'ensemble de ces éléments rassemblés en un seul.

Prenons l'exemple d'un enregistrement musical possédant les informations suivantes : titre de l'album, titre de la piste et nom de l'artiste. Ces informations sont associées à des actions ou des fonctionnalités telles que : l'ajout de la piste à une liste de lecture et la lecture de tous les enregistrements de l'artiste. Si nous rassemblons tous ces éléments en un seul, nous pouvons créer un objet nommé « album » (cf. Figure1).

Ainsi, les avantages procurés lorsque l'on intègre toutes ces valeurs ou fonctions sont la possibilité de ne suivre qu'une seule variable, comme par exemple la variable « album », plutôt que plusieurs d'entre elles qui sont représentés dans l'exemple comme « ajout de la piste » ou bien « titre de la piste ». La modélisation orientée objet est proche de la réalité ce qui fait qu'il sera relativement facile de modéliser une application de cette façon.

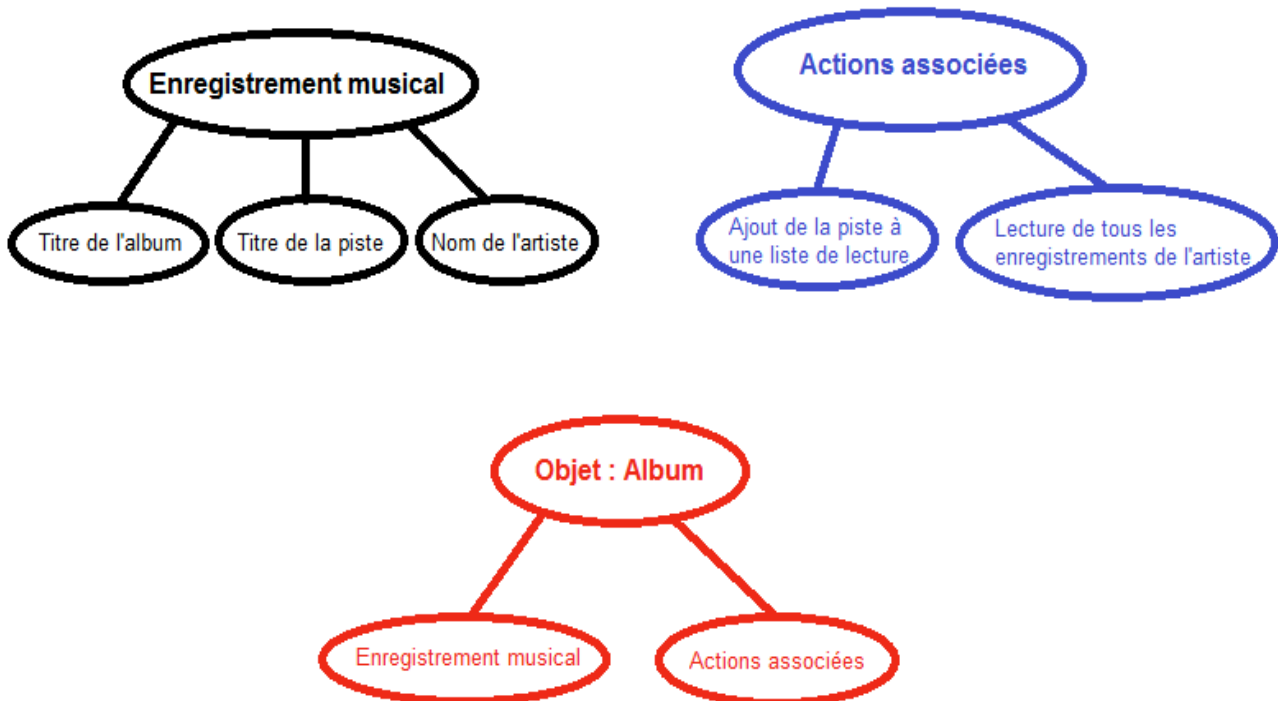


Figure 1 : exemple simplifié de la Programmation Orientée Objet

### 3.1.2 Définition de Java et Swing

Java se définit comme étant un langage de programmation orientée objet (POO) créée en 1982 par la société Sun Microsystems mais qui a été par la suite rachetée par la société Oracle durant l'année 2009. La principale particularité de Java est que ses logiciels sont portables sur plusieurs systèmes d'exploitation tels que Windows, Linux, ou bien Mac OS. Il existe un système d'exploitation destiné à Java sous le nom de Java OS possédant un environnement de développement complètement dédié à ce dernier.

Java permet de développer différents types d'applications. C'est pourquoi il existe des environnements Java (JRE)\* s'agissant d'une famille de logiciel permettant de créer des programmes pour des plates-formes différentes telles que :

- Java 2 Standard Edition ou J2SE : permet le développement d'applications de composants lourds (exemples : Open office, Word, Excel...)
- Java 2 Enterprise Edition ou J2EE : permet le développement d'applications web en Java autrement dit des applications de composants légers (exemples : Eclipse, Mozilla, Java Web Stat ...)
- Java 2 Micro Edition ou J2ME : permet le développement d'applications mobiles, application de PDA\*

Swing est une bibliothèque graphique conçue pour le langage de programmation Java. Swing fait partie du package Java Foundation Classes (JFC)\* qui est inclus dans J2SE. C'est une Application Programming Interface (API)\*, ce qui signifie que c'est une interface de programmation dont le but est le même que celui de l'API AWT mais dont les modes de fonctionnement et d'utilisation se diffèrent complètement. Swing a été intégré au JDK\* depuis sa version 1.2. Elle offre la possibilité de réaliser des interfaces graphiques quel que soit le système d'exploitation et utilise le principe Modèle-Vue-Contrôleur (MVC). La MVC est un cycle dans lequel le client voit une interface dynamique qu'il peut contrôler par des actions, rendues possibles par l'affichage de l'interface. Les composants Swing jouent le rôle de contrôleur au sens du MVC et se compose de plusieurs choix d'apparence.

Depuis les premières versions de Java, Abstract Windows Toolkit (AWT) fournit une API indépendante du système d'exploitation pour mettre en œuvre des composants graphiques. Dans AWT, chaque composant est dessiné et contrôlé par un composant tiers natif spécifique au système d'exploitation. C'est pourquoi les composants d'AWT sont appelés composants lourds.

Au contraire, les composants Swing sont décrits comme légers. En effet, ils ne requièrent pas d'allocation de ressources natives de la part du gestionnaire de fenêtres sous-jacent, mais empruntent les ressources de leurs ancêtres.

Une grande partie de l'API Swing est une extension complémentaire à AWT plutôt qu'un remplaçant direct. L'affichage est fourni par Java2D, un autre composant de JFC. Cependant, l'usage conjoint de composants légers et lourds au sein d'une même fenêtre est généralement déconseillé à cause de problèmes de gestion de la profondeur.

### 3.1.3 A propos de Alchemy

Alchemy est un logiciel permettant l'archive de documents, d'images et d'autres données. Il permet notamment la récupération et la distribution d'une quantité d'informations dans des disques durs externes tels que les CD-R, CD-RW, DVD, etc.

L'élément essentiel d'Alchemy est la base de données. Une base de données d'Alchemy, peu importe dans quel disque de stockage elle a été enregistrée, représente un véritable moyen pour l'archivage et la distribution de données. Les bases de données du logiciel qui ont été enregistrées peuvent être lu avec n'importe quel dispositif compatible installé tel que Windows sur PC ou serveur.

L'approche de gestion de contenu du logiciel Alchemy permet la récupération d'information non seulement grâce au nom du répertoire mais aussi grâce au contenu du document. Cette capacité est due aux fonctionnalités du logiciel Alchemy qui permettent aux utilisateurs d'indexer une partie spécifique des documents et d'associer les éléments indexés aux groupes de documents contenus dans les dossiers.

Voici quelques exemples de ce que Alchemy peut produire :

- Archive de documents
- Récupération et stockage d'images
- Archive de fichiers Web
- Distribution d'un large nombre de fichiers textes et images
- Stockage supérieur à 650 MB sur un CD-ROM

### 3.2 Objectifs du stage

Durant mes 10 semaines de stage, j'ai été amenée à effectuer plusieurs missions mais ma principale mission a été la création d'une interface graphique grâce au logiciel Eclipse pour un client. Le langage Java a été choisi pour la programmation de cette application car c'est un langage moderne et notamment pour son excellente portabilité.

L'objectif de ce stage était de réaliser une interface composée de trois éditeurs de textes ainsi que de trois boutons. Chacun des boutons sont nommés : bouton initial, bouton complément et bouton déstockage. Cette application doit permettre au client de pouvoir saisir un numéro de dossier et ainsi accéder à l'historique de son dossier, c'est-à-dire, lorsqu'il a été rempli pour la première fois, ensuite lorsqu'il a été complété ou bien déstocké. En résumé, l'interface est un historique d'archivage de documents. Les dossiers sont stockés dans la base de données du logiciel Alchemy qui est installé au préalable et programmé en langage batch.

Le batch ou un fichier .bat est un fichier de commandes MS-DOS (Microsoft Disk Operating System) permettant de concevoir des scripts ensuite interprétés par un interpréteur de commandes afin d'exécuter le fichier. Le MS-DOS est un système d'exploitation de type DOS déployé par Microsoft. Il est équipé d'une interface en ligne de commande.

La mission a nécessité également de pouvoir intégrer des fichiers batch dans un programme Java qui aurait permis le lancement de l'interface Alchemy (cf. Figure 1). Pour cela, les commandes batch devaient être intégrées en tant que texte dans le programme Java et en ayant pour variable le numéro de dossier saisi. Ainsi, lorsque le client saisit un numéro de dossier puis clique sur un des trois boutons, les commandes batch composées de la variable sont générées dans un fichier batch afin que celui-ci puisse être lancé (cf. Figure 2) par le MS-DOS.

The screenshot shows the 'Alchemy Build' application window. The title bar reads 'Alchemy Build'. The menu bar includes 'Fichier', 'Edition', 'Affichage', 'Création', and 'Outils'. The toolbar contains various icons for file operations. On the left, a tree view shows a project structure under 'Bases de données' with folders like 'traitements dossiers de crédits', 'Dossiers', 'Numérisation temporaire', 'Copies Exécutoires ok', 'Délégation d'assurance', 'Cautions reçues', 'Délégation de créance', 'TEST', 'dossier-compléments.jpg', 'dossier-initial.jpg', and 'BPMED'. The main area is titled 'Profil de document' and contains two columns of form fields: 'INFORMATION DOSSIER:' and 'INFORMATION NUMERISATION:'. Below these are sections for 'STOCKAGE:' and 'DESTOCKAGE:'. The 'DOSSIER INITIAL' logo is visible in the bottom left corner of the application window.

INFORMATION DOSSIER:		INFORMATION NUMERISATION:	
Banque:	CHAIK	Agence:	27
NDC1:	456	Collaborateur:	V.DORBE-COULOMB
NDC2:		Date de réception:	
Nom du client:		Date de demande de num.:	
N° etude:		Date de Traitement:	
Type de prêt:	VCI	Tache de numérisation:	Non
Montant:		Check-list/Crédit-liste:	Appel/Mail
Type de dossier:	Complément siège	DRC:	Oui
Date réception (Compl):	12/02/2012	Accord:	Non
Date demande (Compl):	02/02/2017	Assurances:	Oui
Date traitement (Compl):	21/12/2017	Docs Règlementaires:	Non
		Docs Prêts Règlementaires:	Sans Objet
		DSP:	Sans Objet
		Temps de traitement:	
		Mise en Suspens:	Oui
		Dossier Archivé:	Oui
STOCKAGE:			
Zone de stockage:			
Ref Boite:			
Dossier Entrée:			
DESTOCKAGE:			
Date de déstockage:		N de prêt:	
		Date échéance de prêt:	

Figure 1 : Interface de la base de données Alchemy

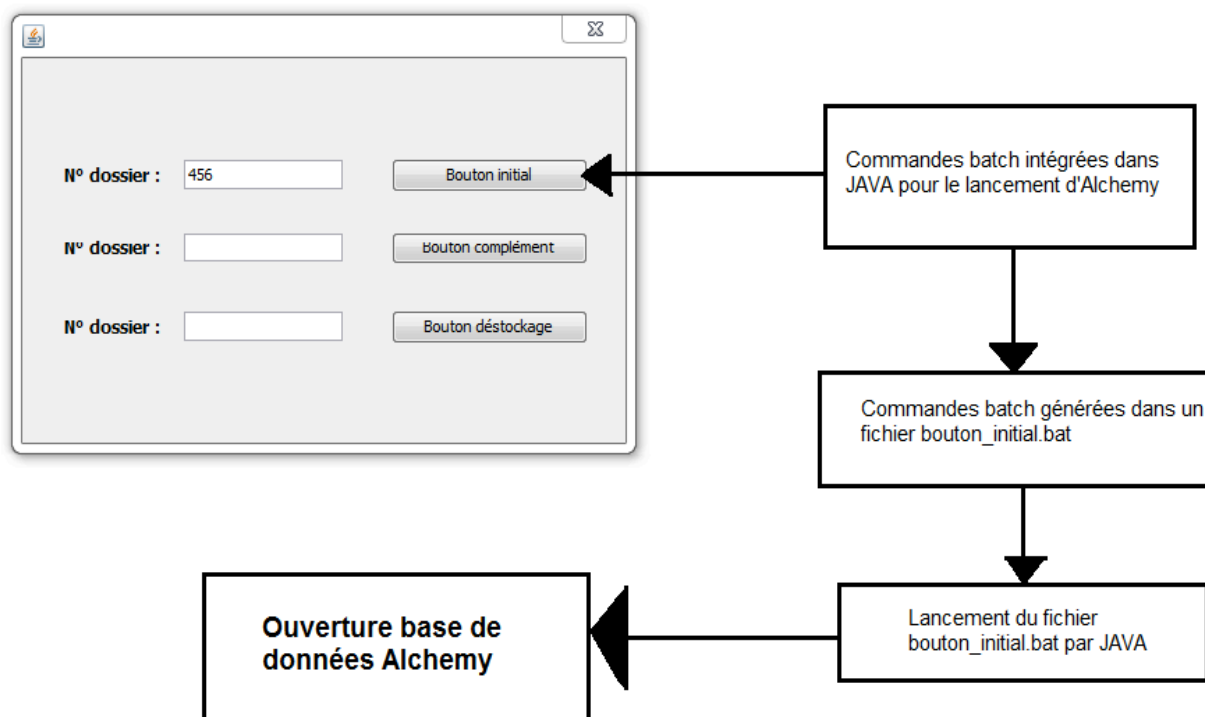


Figure 2 : Schéma expliquant l'intégration de commandes batch dans un programme JAVA

Cependant, après avoir essayé d'intégrer des commandes batch en tant que texte dans un programme Java et ayant rencontré certaines difficultés, mes objectifs ont été quelque peu modifiés afin que je puisse aboutir au fonctionnement de l'application. La nouvelle mission qui m'a été confiée ne différait pas vraiment de la précédente hormis le fait que l'interface ne posséderait que deux champs de saisies : un pour le numéro de dossier et un autre pour le numéro d'étude du dossier (cf. Figure 3). En somme, les commandes batch n'auront plus à être intégrées dans le programme Java.

A partir de cette nouvelle interface, le client devra saisir deux éditeurs de texte qui seront composés d'une vérification du remplissage des champs. Lorsque ces derniers seront remplis, ils seront générés dans un fichier texte puis intégrés dans un fichier batch non intégré à mon programme Java (cf. Figure 3). Nous retrouverons donc les deux fichiers texte NDC1.txt et Numero\_etude.txt qui contiendront chacun un numéro saisi par le client sur l'interface. Les fichiers textes NDC1.txt et numero\_etude.txt seront donc ajoutés dans un fichier batch qui sera lui-même lancé sur le MS-DOS et permettra l'ouverture de la base de données Alchemy.

Cependant, nous ne nous concentrerons désormais que sur la partie programmation Java, c'est-à-dire, sur la génération des fichiers textes NDC1.txt et numero\_etude.txt en Java. Il existe une partie batch traitée par mon cadre professionnel afin qu'il puisse lancer des documents archivés d'Alchemy. De plus, il a ajouté les fichiers NDC1.txt et num\_etude.txt à son fichier batch afin qu'Alchemy puisse prendre en compte les variables saisies dans l'interface.

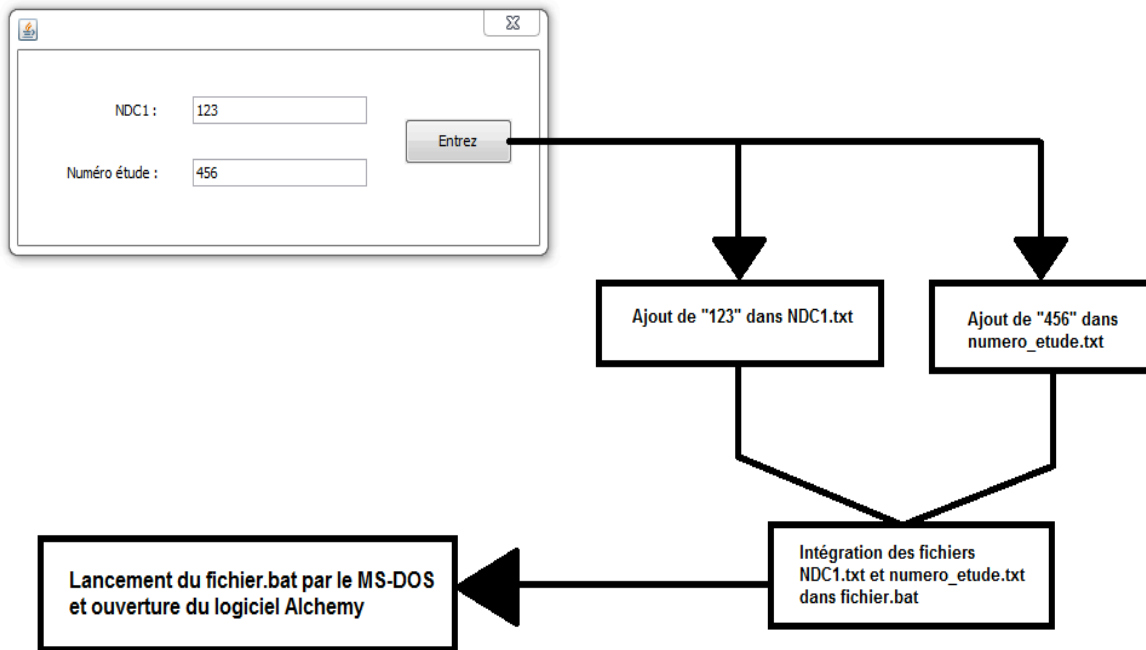


Figure 3 : Nouvelle interface à réaliser

Après réalisation de l'interface, un export de l'application en fichier JAR m'a été demandé afin de rendre l'application exécutable par Windows et tous les autres systèmes d'exploitation contenant un environnement Java. Afin d'éviter que le client ait à installer cet environnement Java, la mission qui m'a été confiée était de créer un exécutable de l'application ne nécessitant pas l'installation d'un JRE et donc de faire en sorte que l'application inclut cet environnement dans l'application.

Un fichier exécutable est un fichier informatique contenant un programme. Lors du chargement de ce fichier, cela va entraîner l'exécution du programme.

En effet, le but était de développer une application disponible pour un bon nombre de postes informatiques, c'est la raison pour laquelle il a été choisi de ne pas procéder à l'installation de l'environnement Java pour chacun des postes et ainsi trouver une alternative.

### 3.3 Missions effectuées

#### 3.3.1 Installation des outils de développement Java

Tout d'abord, il est important de vous présenter la manière dont j'ai pu installer les outils de développement Java. En effet, j'ai été amené à installer le produit que je devais utiliser car l'entreprise ne disposait pas de l'environnement Java. Le site d'Oracle est le site par référence permettant d'installer un JRE ou bien un JDK. Le JRE possède tous les éléments nécessaires pour que le programme Java puisse être exécuté sur l'ordinateur tandis que le JDK est un kit contenant déjà un JRE et tout le nécessaire pour développer, compiler, etc.

Cependant, il existe un IDE (Integrated Development Environment) qui est une interface de développement intégré permettant le développement, la compilation et l'exécution d'un programme donc il n'est pas nécessaire d'installer un JDK car l'IDE contient déjà tout. C'est pourquoi l'installation du JRE est amplement suffisante.

Pour installer un JRE, il faut tout d'abord accepter la licence de Java SE puis choisir le système d'exploitation qui correspond à celui que l'on utilise et qui sera pour nous Windows 64 bits. (cf. Figure 1)

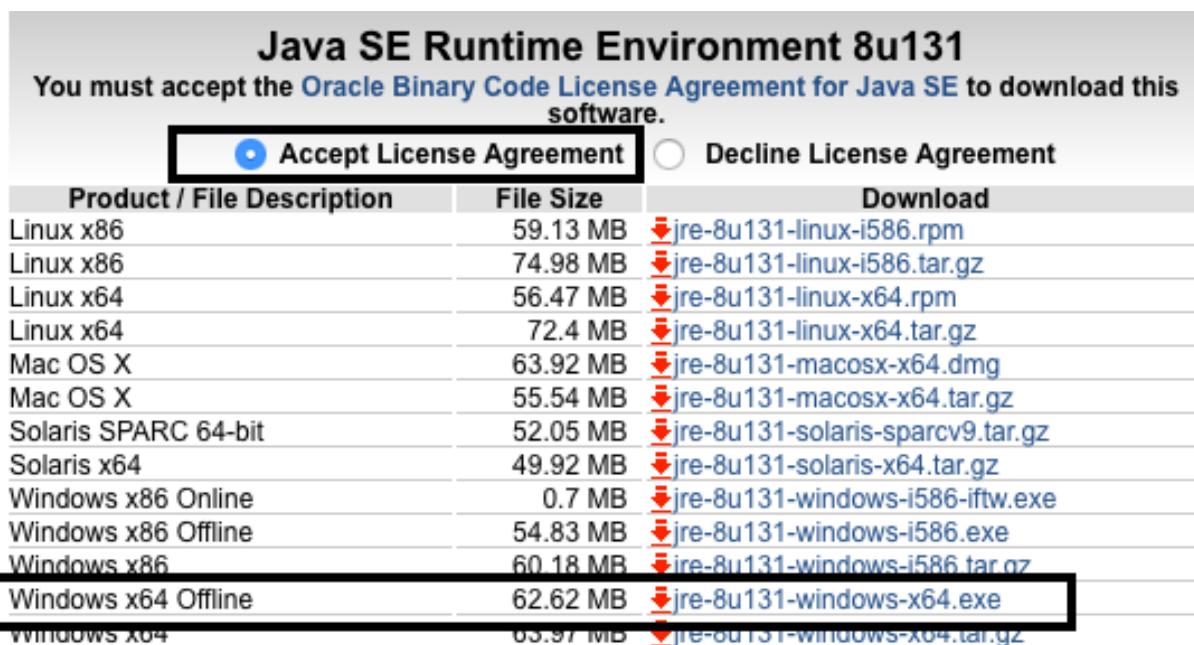


Figure 1 : Installation JRE pour Windows 64 bits

Une fois que le JRE est installé, j'ai dû procéder à l'installation du logiciel Eclipse IDE qui est nécessaire afin de posséder une interface de développement. Eclipse permet de simplifier la programmation car il propose des raccourcis et des aides à celle-ci. C'est une application gratuite et disponible pour plusieurs systèmes d'exploitation. Ce logiciel permet donc de programmer et compiler des applications et des applets\* mais aussi traduire les programmes Java en langage Java uniquement compréhensible par le JRE.

Afin d'effectuer le téléchargement de Eclipse IDE, il faut se situer sur sa page de téléchargement [www.eclipse.org](http://www.eclipse.org) puis choisir la version de Eclipse correspondant à notre OS ainsi que le miroir le plus proche de chez soi qui est ici la France. Un fichier.tar.gz sera donc téléchargé contenant le logiciel Eclipse Installer. Lorsque Eclipse Installer est lancé, il faudra choisir d'installer « Eclipse IDE for Java Developers » (cf. Figure 2).

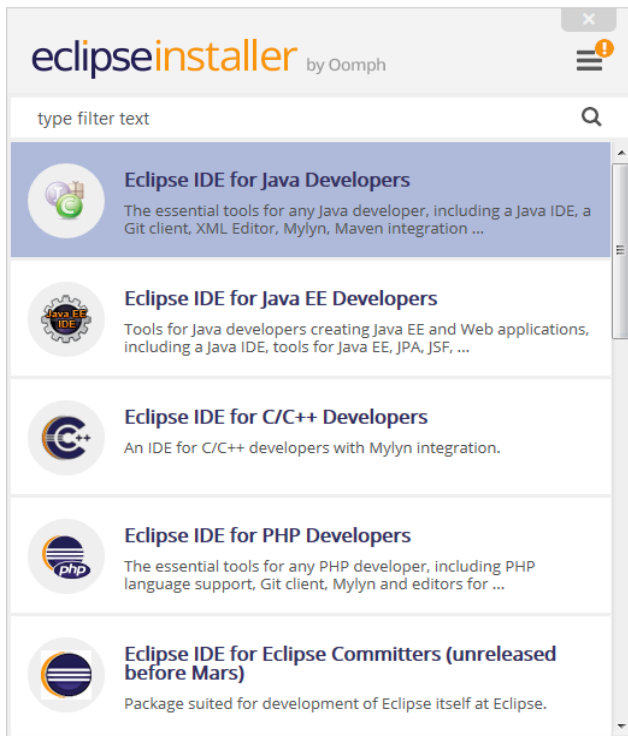


Figure 2 : Eclipse Installer

Le logiciel Eclipse IDE ainsi installé et exécuté, il m'a été demandé d'indiquer dans quel répertoire il était souhaitable d'enregistrer nos projets, autrement dit, les différents programmes que nous avons utilisé. En général, le projet est automatiquement ajouté dans un répertoire C:\workspace mais il est possible de modifier le chemin ainsi que le répertoire de destination. J'ai donc décidé de placer mon projet dans le répertoire C:\Users\Utilisateur\workspace (cf. Figure 3).

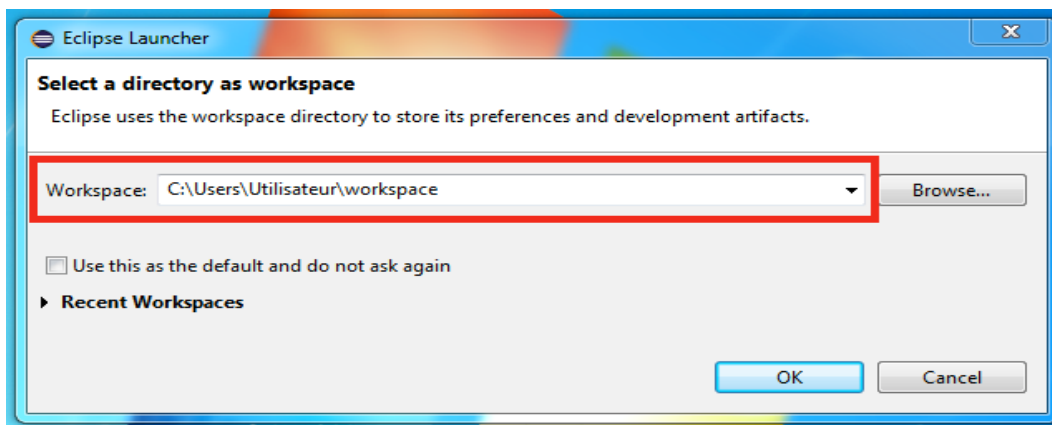


Figure 3 : Répertoire d'enregistrement des projets Eclipse

Suite à cela, l'interface d'Eclipse va s'afficher et c'est ainsi qu'à partir de ce dernier j'ai pu créer mon projet que j'ai nommé PROJET\_BP MED en allant dans File > New > Java Project (cf. Figure 4). Un projet est composé d'un fichier source nommé « \src » contenant lui-même un paquet par défaut autrement appelé default package qui est un fichier regroupant des classes Java\* (cf. Figure 5). En effet, je n'ai pas créé de package c'est la raison pour laquelle celui-ci s'affiche par défaut dans mon projet.

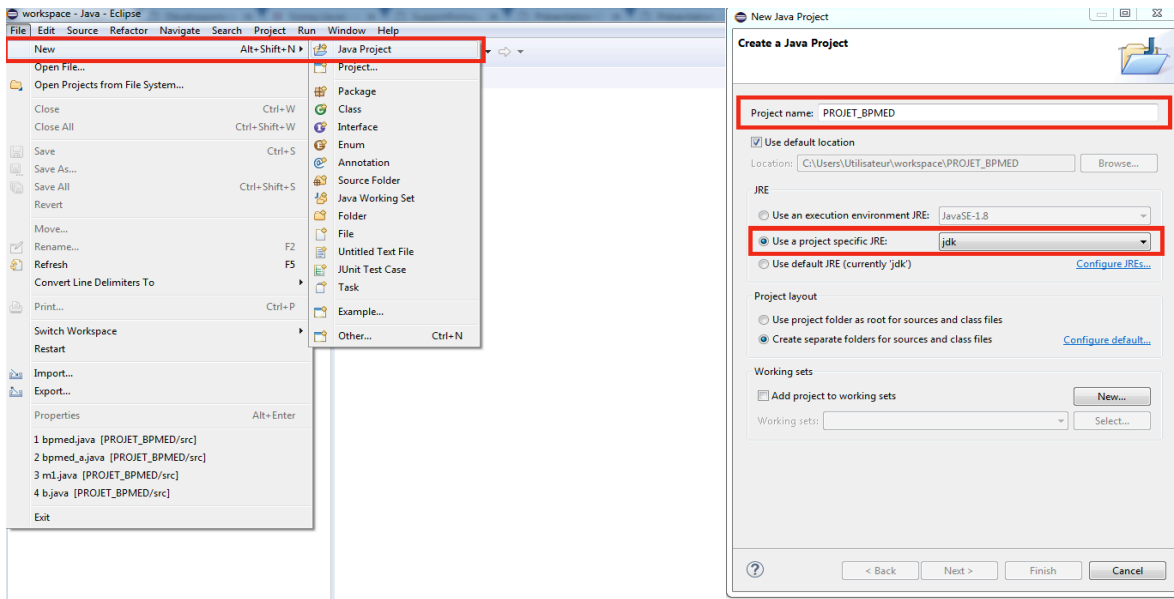


Figure 4 : Création du projet « PROJET\_BP MED »

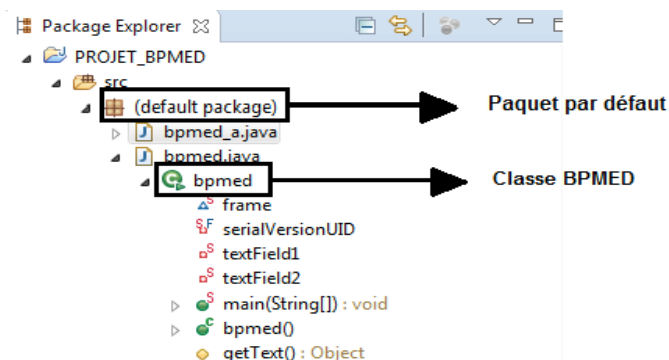


Figure 5 : Composants du projet « PROJET\_PB MED »

J'ai donc pu procéder à l'installation de l'environnement Java tel que le JRE et l'IDE Eclipse. Cependant, ma mission consistait à créer une interface dynamique pour Eclipse qui ne disposait pas d'interface Swing\*. En effet, la principale spécificité d'Eclipse est qu'elle soit développée autour de la notion de plugin qui se définit par un paquet complétant un logiciel hôte afin de lui apporter de nouvelles fonctionnalités.

La fonctionnalité permettant l'utilisation d'une interface graphique avec Swing\* se nomme Window Builder qui est un outil de conception d'applications Java GUI\* qui regroupe des composants SWT\* et Swing. J'ai donc dû procéder à l'installation de ce plugin. L'onglet « Help » d'Eclipse IDE propose un menu « Install New Software » permettant d'installer de nouvelles fonctionnalités dans lequel j'ai pu installer Window Builder. L'installation fût rapide car il m'a suffi d'ajouter l'url <http://download.eclipse.org/releases/luna> puis coché les options de « Swing designer » à « Window Builder XML Core » (cf. Figure 6). C'est ainsi que l'interface principale d'Eclipse a pu être doté d'un menu « Design » correspondant au nouveau plugin installé.

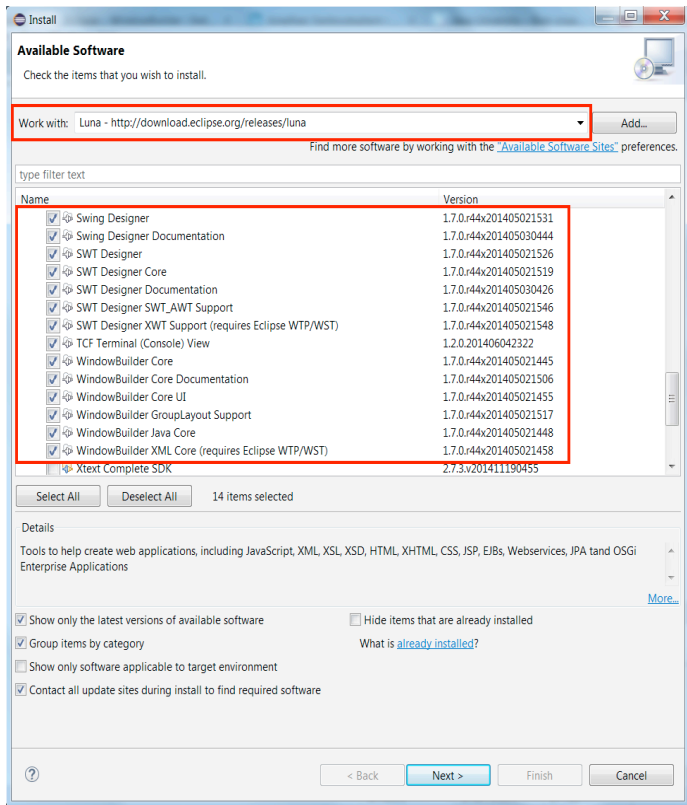


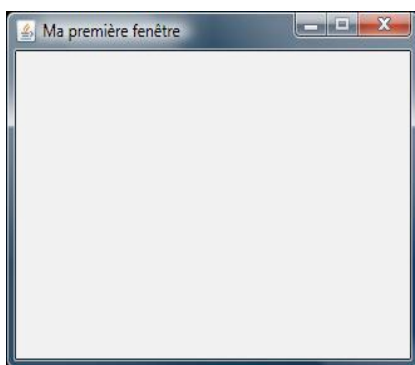
Figure 6 : Installation Window Builder

Nous nous sommes alors concentré sur le plugin Window Builder qui était la principale fonctionnalité utilisée durant mon stage pour la réalisation de mon projet. Ce plugin permet d'ouvrir une interface vierge qui nous servira de support de travail.

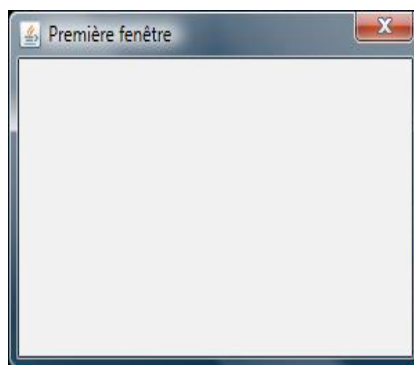
### 3.3.2 Réalisation de l'interface Swing

Swing propose plusieurs types de fenêtre telles que :

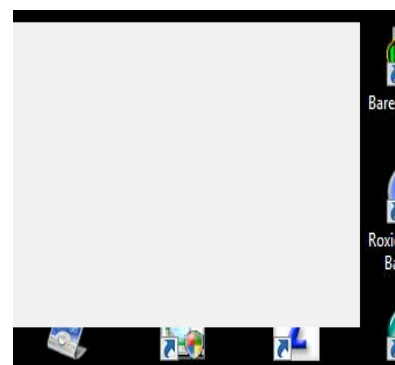
- JFrame : destinée à être la fenêtre principale de l'application et ne dépend d'aucune autre. Elle possède une barre de titre et des boutons de redimensionnement/fermeture.
- JDialog : permet de créer des boîtes de dialogue, comme par exemple pour indiquer qu'il y a une erreur, et qui va de pair avec la fenêtre principale. Elle possède une barre de titre avec un seul bouton de fermeture.
- JWindow : conteneur ne contenant pas de barre de titre, de boutons de fermeture/redimensionnement et n'est pas redimensionnable par défaut.



JFrame



JDialog



JWindow

Sous la demande de mon tuteur professionnel, qui souhaitait avoir une interface composée d'un bouton de fermeture sans redimensionnement et dépendant d'une boîte de dialogue, j'ai donc choisi d'utiliser un JDialog (cf. Figure 1).

Tout d'abord, le menu design affichait un JDialog dont le contenu était vierge ainsi que différentes options à utiliser pour réaliser une interface dynamique. Lors de la réalisation de l'interface j'ai principalement utilisé les composants JTextPane qui un composant de texte où peut s'ajouter des attributs tels que des images, JTextField correspondant à une zone de texte éditable d'une ligne et JButton correspondant à un bouton à cliquer avec un texte (cf. Figure 2).

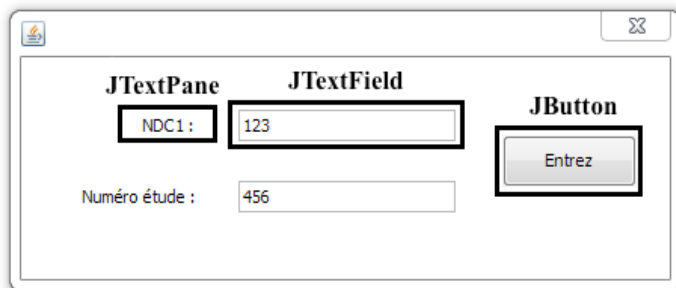


Figure 2 : Les différents composants Swing utilisés

Il suffisait donc de choisir le composant pour que ce dernier puisse s'ajouter dans le programme Java. Ce programme a surtout été effectué autour de la méthode\* JButton() car toutes les actions se déroulaient au moment où l'on cliquait sur le bouton « Entrez ». Dans ce projet, le but du programme était donc d'ajouter une action dans le JButton permettant la récupération des valeurs des JTextField afin de les générer dans un fichier texte. Après avoir créé l'interface composée de deux titres, deux champs de saisies et un bouton de validation, il a fallu que je réalise une vérification du remplissage de champs à l'aide d'une boîte de dialogue. Cette boîte de dialogue devait afficher un message préventif lorsque les deux champs n'étaient pas remplis (cf. Figure 3 et 4).

```

JButton btnEntrez = new JButton("Entrez");
btnEntrez.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

if (textField1.getText().trim().length() > 0 && textField2.getText().trim().length() > 0) // Si la valeur des champs "textField1" et "textField2" est supérieure à 0,
                                                                                          // c'est-à-dire si les champs ne sont pas vides
    {
        // Ajout des actions liées au bouton Entrez
    } else { // Sinon
        new JOptionPane();
        JOptionPane.showMessageDialog(null, "Veuillez saisir une valeur dans le champ ", "Attention", JOptionPane.WARNING_MESSAGE);
        // Ajout d'une boîte de dialogue indiquant un message préventif
    }
    }
});
contentPane.add(btnEntrez);

```

Figure 3 : Ajout d'une boîte de dialogue dans un programme Java

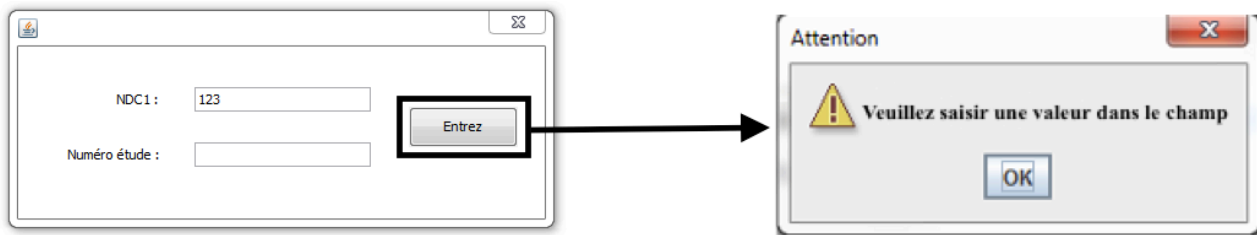


Figure 4 : Affichage de la boîte de dialogue

Dans mon programme Java, j'ai dû créer des variables « var-lettre-du-disque.txt » correspondant au disque dur C : et « var-repertoire.txt » correspondant au répertoire /BPMED/JAVA afin d'indiquer les chemins NDC1.txt et num\_etude.txt en utilisant ces variables. En effet, cela permettrait au client de modifier le chemin de ces deux fichiers en modifiant seulement var-lettre-du-disque et var-repertoire si ce dernier ne possède pas le même emplacement.

### 3.3.3 Création d'un exécutable .jar

Un fichier JAR est un fichier d'archive dans un format spécifique à Java. Ce sont donc des fichiers compressés et leurs intérêts s'appuie autour du fait qu'ils soient auto-exécutables cela veut dire qu'ils contiennent forcément un programme Java. Cependant, il existe des fichiers JAR ne contenant pas de programme Java et qui ne sont donc pas auto-exécutables. Un fichier jar est un fichier portable et dépend d'un environnement Java tel que le JRE.

Suite à la réalisation de l'interface, j'ai donc exporté l'application en tant que fichier JAR auto-exécutable afin de rendre mon application portable par Windows 7. Pour cela, j'ai effectué un clique droit sur le fichier.java à exporter puis choisi le menu Export (cf. Figure 1). J'ai ensuite choisi « Runnable JAR File » signifiant fichier .jar auto exécutable puis j'ai indiqué le répertoire de destination pour l'exportation de mon exécutable qui était C: \BPMED\JAVA\ (cf. Figure 2)

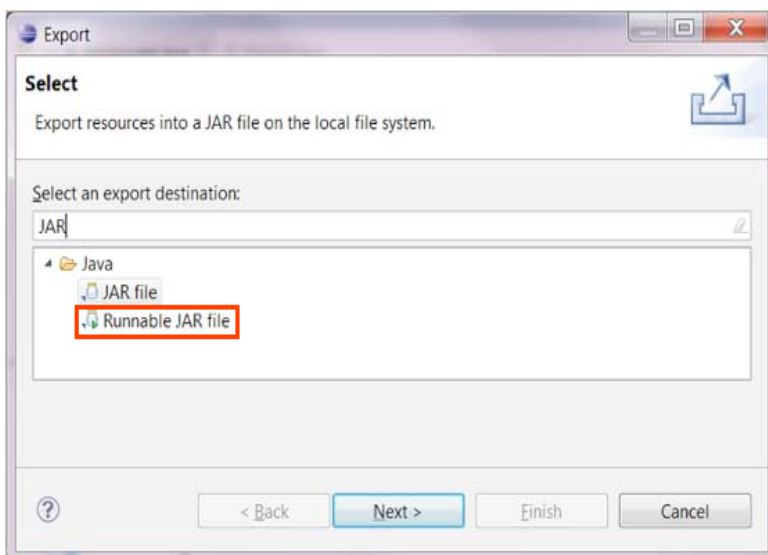



Figure 1 : Exportation en tant que fichier auto-exécutable

## JAR Package Specification

 The export destination will be relative to your workspace.



Select the resources to export:

<input type="checkbox"/> GrapheX	<input checked="" type="checkbox"/> .classpath
<input type="checkbox"/> INF421	<input checked="" type="checkbox"/> .project
<input checked="" type="checkbox"/> INF431	<input checked="" type="checkbox"/> blabla.txt

Export generated class files and resources  
 Export all output folders for checked projects  
 Export java source files and resources

Select the export destination:

JAR file:

Options:

Compress the contents of the JAR file  
 Add directory entries  
 Overwrite existing files without warning

Figure 2 : Affichage d'un menu Export pour indiquer le répertoire de destination

## 3.4 Bilan personnel et experientiel

### 3.4.1 Difficultés rencontrées et solutions apportées

Durant la réalisation du projet, j'ai souvent rencontré des difficultés notamment au début du stage lorsqu'il a fallu intégrer un fichier batch dans le programme Java. C'est la raison pour laquelle mon tuteur m'a proposé de ne rien intégrer dans mon programme. En effet pour intégrer un batch il aurait fallu copier les commandes batch dans le programme Java et utiliser une commande permettant de l'exécuter dans un fichier batch afin d'ouvrir le logiciel Alchemy directement à partir de ce programme. Cependant, certains caractères n'étaient pas acceptés par Java et il a fallu modifier le programme afin de l'adapter ce qui générait la plupart du temps des erreurs notamment des erreurs d'inattention de ma part et cela retardait donc le projet.

Ajouté à cela, j'ai pu rencontrer une autre difficulté à savoir lorsque j'ai du rendre mon fichier .jar exécutable par le client contenant déjà un JRE sans qu'il n'ait à l'installer directement. Suite à de nombreuses recherches j'ai essayé de trouver une solution pour éviter au client d'installer un JRE et qui se nomme Java Web Start (JWS). C'est un logiciel intégré à l'environnement JRE et automatiquement ajouté lors de l'installation de ce dernier. Le logiciel Java Web Start permet de télécharger et d'exécuter des applications Java à partir du Web. Par le biais à Java Web Start j'ai pu mettre en place un site web portant mon application Java afin que le client puisse le télécharger sans avoir à installer de JRE. L'utilisation d'une application grâce à JWS implique la réalisation de plusieurs étapes :

- Packager l'application dans un fichier jar en le signant si nécessaire
- Créer le fichier de lancement .JNLP
- Copier les deux fichiers sur le serveur web

J'ai tout d'abord utilisé l'outil « keytool » afin de créer une signature de mon application car il est nécessaire pour la sécurité de l'application. Java Web Start utilise un protocole nommé Java Network Launching Protocol (JNLP) dont son fichier est au format XML\* permettant de décrire l'application et que j'ai pu nommer bp.jnlp (cf. Figure 1). En effet ce fichier contient l'URL\* du serveur web ainsi que l'exécutable bpm.jar.

DUT R&T - Stage 2017 - Michella MITSAKIS - OSIC -

J'ai ensuite configuré un serveur web IIS\* qui fournissait tous mes fichiers permettant l'exécution de l'application et je l'ai configuré en ajoutant une extension JNLP ainsi qu'un type de contenu (MIME)\*. Puis j'ai créé un fichier HTML afin d'ajouter l'URL pointant vers mon fichier bp.jnlp sur le serveur web (cf. Figure 2). Afin de tester mon application, j'ai ouvert un navigateur internet puis j'ai saisi l'URL de la page contenant mon fichier jnlp.

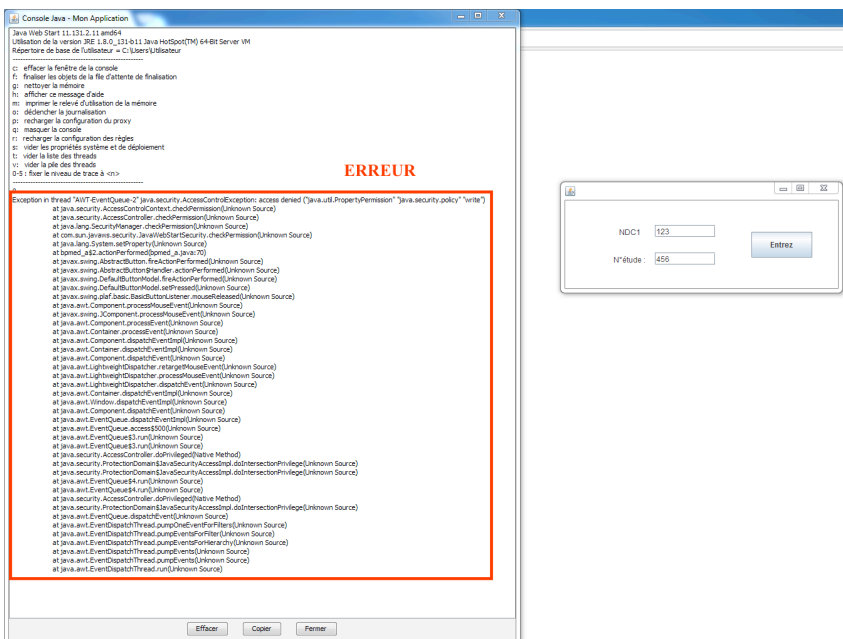
```

1 <html>
2   <head>
3     <title>Mon Application</title>
4   </head>
5
6   <body>
7     Mon programme à déployer :
8     <br>
9     <a href="http://localhost/bp.jnlp">PROG</a>
10  </body>
11 </html>

```

Figure 2

Cependant, pour en arriver à la difficulté rencontrée, je n'ai pas réussi à faire fonctionner mon application sur le serveur web car ce dernier possédait un problème de permission concernant l'exécution de l'application (cf. Figure 2). J'ai donc essayé de résoudre le problème en créant un fichier java.policy (cf. Figure 4) permettant de modifier les permissions pour l'exécution d'une application mais cela était sans conséquence. Mon tuteur professionnel a donc utilisé une commande batch dans son fichier .bat afin d'exécuter l'application sans avoir à installer de JRE.



```

1 /* AUTOMATICALLY GENERATED ON Thu Jun 01 12:12:06 CEST 2017*/
2 /* DO NOT EDIT */
3
4 grant codeBase "http://localhost/" {
5   permission java.io.FilePermission "C:\\\\BPMED\\Test_JAVA", "read, write, delete, execute";
6   permission java.net.SocketPermission "*", "accept, connect, listen, resolve";
7   permission java.lang.RuntimePermission "*";
8   permission java.util.PropertyPermission "*", "read, write";
9   permission java.awt.AWTPermission "showWindowWithoutWarningBanner";
10 };
11
12

```

### 3.4.2 Compétences acquises

Lors de mon stage, j'ai pu acquérir diverses compétences techniques concernant le langage de programmation Java et son environnement. Je suis désormais capable de réaliser une interface dynamique notamment grâce à l'utilisation de Window Builder.

J'ai notamment pris connaissance de nouveaux programmes comme par exemple celui permettant la création d'une boîte de dialogue indiquant des messages d'erreurs (JDialog) ou bien celui de récupérer un champ saisi en variable afin de l'ajouter dans un fichier texte. J'ai également pu approfondir mes connaissances dans l'écriture de conditions « if » car cette fonction m'a beaucoup servi pour la réalisation de mon programme Java.

Je me suis expérimentée dans la recherche d'informations afin de pouvoir répondre à diverses questions et de résoudre certains problèmes. A travers les tâches qui m'ont été confiées j'ai pu développer mon savoir-être en matière d'organisation. En effet, le client souhaitait voir l'avancement du projet dans les plus brefs délais ce qui demandait de la rapidité et de l'efficacité.

Ce stage m'a notamment rendu plus autonome dans mon travail en entreprise, ce dont je ne pensais pas être capable auparavant. J'ai aussi appris à être plus responsable car une grande partie du travail reposait entre mes mains. En effet, j'étais la seule à posséder certaines compétences dans la programmation en Java.

## 4 Conclusion

Ce stage au sein de la société OSIC fut une expérience professionnelle très enrichissante car j'ai pu approfondir mes compétences en informatique plus précisément dans la programmation Java.

A travers ce projet, nous pouvons affirmer qu'une interface dynamique Java autrement dit Swing peut être réalisée, à condition que l'environnement Java soit installé suivi d'une interface de développement où l'on peut ajouter des plugins. En somme, cette interface Swing peut devenir accessible à un client si ce dernier possède un environnement Java déjà installé sinon il faudrait que l'application soit portée par un serveur web.

Cette première expérience professionnelle dans la programmation informatique m'a fait mûrir et ouvrir les yeux sur la cohérence de mon projet professionnel avec ma personnalité. Ainsi, ce stage m'a notamment permis de confirmer mon projet de poursuite d'études qui est de continuer ma formation en école d'ingénieur informatique.

## 5 Glossaire

**JDK**, Java Development Kit : Ensemble de bibliothèques logicielles de base du langage de programmation Java, ainsi que les outils avec lesquels le code Java peut être compilé, transformé en bytecode destiné à la machine virtuelle Java.

**JRE**, Java Runtime Environment : Environnement d'exécution Java correspondant à une famille de logiciels permettant l'exécution des programmes écrits en langage de programmation Java sur différentes plateformes informatiques.

**JFC**, Java Foundation Classes : Ensemble de classes permettant le développement d'IHM (boutons, tableaux, listes, etc.)

**API**, Application Programming Interface : Correspond à une interface de programmation.

**PDA**, Personal Digital Assistant : Appareil électronique destiné à un usage personnel. C'est le cas des agendas électroniques, téléphones, tablettes, appareils photo, etc.

**Swing** : Ensemble de classes représentant une extension d'AWT et une amélioration des fonctionnalités des composants AWT ; le package `java.awt.swing` est intégré à la version 1.2 du JDK.

**SWT**, Standard Widget Toolkit : Bibliothèque graphique libre pour Java. Elle se compose d'une bibliothèque de composants graphiques (texte, label, bouton, panel).

**GUI**, Graphical User Interface : Outil graphique visuel permettant de communiquer avec un utilisateur du programme.

**Classe Java** : Une classe Java définit un modèle d'objet. Chaque objet créé à partir d'une classe est appelé instance de cette classe.

**Applet** : Petit programme ou application originellement le plus souvent en Java, insérée dans une page et exécutée par le navigateur.

**XML**, Extensible Markup Language : Langage de balises complétant HTML et permettant de structurer l'information d'un document. Il est de plus en plus utilisé car il permet de séparer l'information de la présentation.

**URL**, Uniform Resource Locator : Désignation d'une ressource n'importe où sur le web.

**IIS**, Internet Information Server : Serveur Web de l'édition Microsoft.

**MIME**, Multipurpose Internet Mail Extension : Permet au navigateur de connaître l'application qui devra être utilisée lors de la réception des données du serveur web.

## 6 Bibliographie

Présentation de l'entreprise OSIC, *Référence interne*

[https://www.java.com/fr/download/faq/java\\_webstart.xml](https://www.java.com/fr/download/faq/java_webstart.xml), *Qu'est ce que Java Web Start et comment le lancer ?*

<http://vanhouteghem-jonathan.fr/2015/09/eclipse-installation-de-windowbuilder/>, *Installation et premiers pas sur WindowBuilder*

<http://jmdoudoux.developpez.com/cours/developpons/java/chap-swing.php>, *Le développement d'interface graphique avec Swing*

<http://www.objis.com/formation-java/tutoriel-java-creation-archive-jar-eclipse.html>, *Création d'une archive JAR Java*

## 7 Annexes

```
import java.awt.Color;
import java.awt.EventQueue;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.InputStream;

import javax.swing.JButton;
import javax.swing.JDialog;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JTextField;
import javax.swing.JTextPane;
import javax.swing.border.LineBorder;
import java.awt.Font;
import javax.swing.SwingConstants;
import java.awt.Cursor;
import java.util.Locale;
import org.eclipse.wb.swing.FocusTraversalOnArray;
import java.awt.Component;
import java.awt.event.KeyAdapter;
import java.awt.event.KeyEvent;

public class bpmed extends JDialog {

    /**
     *
     */
    private static final long serialVersionUID = 1L;
    private static JTextField textField1;
    private static JTextField textField2;
    static bpmed frame = new bpmed();

    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                frame.setVisible(true);
                textField1.requestFocusInWindow();
            }
        });
    }

    /**
     * Create the dialog.
     */
    public bpmed() {

        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        setUndecorated(true);
        setBounds(100, 100, 365, 130);
    }
}
```

Programme bpmed.java

```

*/
public bpméd() {

    setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
    setUndecorated(true); //La barre de titre Windows n'est plus présente
    setBounds(100, 100, 365, 139);
    JPanel contentPane = new JPanel();
    contentPane.setLocale(Locale.FRANCE);
    contentPane.setCursor(Cursor.getPredefinedCursor(Cursor.DEFAULT_CURSOR));
    contentPane.setFocusCycleRoot(true);
    contentPane.setFocusTraversalPolicyProvider(true);

    contentPane.setBackground(Color.WHITE);
    contentPane.setBorder(new LineBorder(new Color(0, 0, 0)));
    setContentPane(contentPane);
    contentPane.setLayout(null);

    textField1 = new JTextField();
    textField1.setHorizontalAlignment(SwingConstants.LEFT);
    textField1.setBounds(140, 31, 161, 20);
    contentPane.add(textField1);
    textField1.setColumns(10);

    textField2 = new JTextField();
    textField2.setBounds(140, 62, 161, 20);
    contentPane.add(textField2);
    textField2.setColumns(10);

    JButton btnEntrez = new JButton("Entrez");
    btnEntrez.addKeyListener(new KeyAdapter() {
        @Override
        public void keyPressed(KeyEvent e) {

            if (e.getKeyCode() == KeyEvent.VK_ENTER)
            {
                btnEntrez.doClick(); //Le click du bouton Entrez peut être effectué sur la touche "Entrée" du clavier
            }
        }
    });
    btnEntrez.setBounds(140, 93, 86, 28);
    btnEntrez.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {

            if (textField1.getText().trim().length() > 0 && textField2.getText().trim().length() > 0) //Si textField1 et textField2 sont remplis a
            {

                try {

                    File fileIn2 = new File("c:\\BPMED\\JAVA\\var-lettre-du-disque.txt"); //ouverture du fichier var-lettre-du-disque.txt
                    FileInputStream fis2 = new FileInputStream(fileIn2);
                    byte[] buffer2 = new byte[fis2.available()]; //Récupération du contenu du fichier dans une variable de type byte
                    fis2.read(buffer2); //Lecture du contenu de la variable buffer
                    String var_lettre = new String(buffer2); //On parse le contenu de buffer dans notre chaîne
                    fis2.close(); //Fermeture du fichier
                }
            }
        }
    });
}

```

Suite 1 du programme bpméd.java

```

File fileIn1 = new File("c:\\BPMED\\JAVA\\var-repertoire.txt"); //ouverture du fichier var-repertoire.txt
FileInputStream fis1 = new FileInputStream(fileIn1); //Récupération du contenu du fichier dans une variable de type by
byte[] buffer1 = new byte[fis1.available()]; //On parse le contenu de buffer dans notre chaîne
fis1.read(buffer1); //Lecture du contenu de la variable buffer
String var_repertoire = new String(buffer1);
fis1.close(); //Fermeture du fichier

File file2 = new File(var_lettre + var_repertoire + "var_NDC1.txt"); //Ajout chemin du fichier var_NDC1.txt
file2.createNewFile();
String txt1 = textField1.getText(); //Récupération du champs textField1 saisi dans le fichier var_NDC1.txt

PrintStream printStream = new PrintStream(file2);
System.setOut(printStream);
System.out.println(txt1);

File file3 = new File(var_lettre + var_repertoire + "var_num_etude.txt"); //Ajout chemin du fichier var_num_etude.txt
file3.createNewFile();
String txt2 = textField2.getText(); //Récupération du champs textField1 saisi dans le fichier var_num_etude.txt

PrintStream printStream1 = new PrintStream(file3); //Récupération du champs textField1 saisi dans le fichier var_num_etude.txt
System.setOut(printStream1);
System.out.println(txt2);

textField1.setText(""); //Après validation des champs saisis, effacer les zones de texte textField1
textField2.setText(""); et textField2
dispose(); //Fermer l'interface

} catch (FileNotFoundException e3) {
// TODO Auto-generated catch block
e3.printStackTrace();
} catch (IOException e1) {
// TODO Auto-generated catch block
e1.printStackTrace();
}
} else {
new JOptionPane();
JOptionPane.showMessageDialog(null, "Veuillez saisir une valeur dans le champ ", "Attention", JOptionPane.WARNING_MESSAGE);
} //Ajout d'un message préventif si tous les champs ne sont pas remplis
}

});
contentPane.add(btnEntrez);

JTextPane txtpnNdc = new JTextPane();
txtpnNdc.setFont(new Font("Arial", Font.BOLD, 13));
txtpnNdc.setForeground(Color.BLACK);
txtpnNdc.setEditable(false);
txtpnNdc.setText("NDC1 : ");
txtpnNdc.setBounds(69, 31, 61, 20);
contentPane.add(txtpnNdc);

```

Suite 2 du programme bpmед.java

```

        PrintWriter printStream1 = new PrintWriter(1024); //Récupération du champs textuel saisi dans le fichier val_numm.
        System.setOut(printStream1);
        System.out.println(txt2);

        textField1.setText(""); //Après validation des champs saisis, effacer les zones de texte textFiel
        textField2.setText(""); et textField2
        dispose(); //Fermer l'interface

    } catch (FileNotFoundException e3) {
        // TODO Auto-generated catch block
        e3.printStackTrace();
    } catch (IOException e1) {
        // TODO Auto-generated catch block
        e1.printStackTrace();
    }

    } else {
        new JOptionPane();
        JOptionPane.showMessageDialog(null, "Veuillez saisir une valeur dans le champ ", "Attention", JOptionPane.WARNING_MESSAGE);
    } //Ajout d'un message préventif si tous les champs ne sont pas remplis

    }

});
contentPane.add(btnEntrez);

JTextPane txtpnNdc = new JTextPane();
txtpnNdc.setFont(new Font("Arial", Font.BOLD, 13));
txtpnNdc.setForeground(Color.BLACK);
txtpnNdc.setEditable(false);
txtpnNdc.setText("NDC1 : ");
txtpnNdc.setBounds(69, 31, 61, 20);
contentPane.add(txtpnNdc);

JTextPane txtpnNumrotude = new JTextPane();
txtpnNumrotude.setDragEnabled(true);
txtpnNumrotude.setSelectedTextColor(Color.BLACK);
txtpnNumrotude.setSelectionColor(Color.BLACK);
txtpnNumrotude.setFocusTraversalPolicyProvider(true);
txtpnNumrotude.setFont(new Font("Arial", Font.BOLD, 13));
txtpnNumrotude.setEditable(false);
txtpnNumrotude.setText("Num\u00E9ro \u00E9tude : ");
txtpnNumrotude.setBounds(13, 62, 117, 20);
contentPane.add(txtpnNumrotude);
contentPane.setFocusTraversalPolicy(new FocusTraversalOnArray(new Component[]{textField1, textField2, btnEntrez}));

    }

protected Object getText() {
    // TODO Auto-generated method stub
    return null;
}
}

```

Suite 3 du programme bpmmed.java