

الجمهورية الجزائرية الديمقراطية الشعبية

République Algérienne Démocratique et Populaire

Ministère de l'enseignement supérieur et de la recherche scientifique

Université de 8 Mai 1945 – Guelma -

Faculté des Mathématiques, d'Informatique et des Sciences de la matière

Département d'Informatique



Mémoire de Fin d'études Master

Filière : Informatique

Option : Master Académique (AC)

Thème :

Conversion d'un algorithme vers un code source Java

Encadré Par :
Dr. Mohammed CHAOUI

Présenté par :
Aouamria Selma
Zgaoula Radia

Juin 2017

إهداء

إلى روح والدي العزيز رحمه الله.

إلى أمي الغالية أمد الله في عمرها.

إلى رباحين حياتي في الشدة والرخاء.

أخوتي. عبد الحق، إبراهيم، يوسف، أحمد، عبد الغاني ومنير

وأختي فتية وأمينة.

إلى الكتاكيت الصغار محمد، رزان، عبد الرحمان، عبد الباسط، مريم وماريا.

إلى انيسة، منال، بشري، صونيا، جهيدة، طليحة، ايمان كدقاتي.

إلى استاذي محمد شلوي.

إلى شريكتي في هذا العمل راضية زقاولة.

إلى من أتمنى أن أذكرهم.

وإلى كل من شجعني وساعدني على إتمام هذا العمل.

سلمى

Dédicace

Tous les mots ne sauraient exprimer la gratitude, l'amour, le respect, la reconnaissance, c'est tout simplement que : Je dédie cette thèse de projet du fin d'études licence à :

*A Ma tendre Mère **Kabiba***

*A Mon cher Père **Abdelkrim***

A mes chers frères et Beau frère.

A ma sœur et Belle sœur.

*A mes beaux petits **Fadi** et **Okrem**.*

*A ma grande famille **ZGAOULA**.*

*A ma binôme **Selma** et notre encadreur **Ch. Mohammed**.*

A mes très chères amies.

Son oublié mes amis dans les bons moments ou mauvais et précieux hommes dans ma vie.

A tous les citoyens qui m'ont aidé.

A tous les membres de ma promotion.

A tous mes enseignants depuis mes premières années d'études.

A tous ceux qui me sens chers et que j'ai omis de citer.

Zgaoula Radia.

Remerciements

Je remercie :

D'abord le tout puissant Allah, qui nous a accordé sa grâce, donné la santé et la force de réaliser ce travail.

Notre encadreur Mr *Mohammed CHAOU*, pour ses conseils, sa supervision et son aide qui ont été plus que précieux.

Tous les membres de jury pour nous avoir acceptés l'honneur d'examiner notre travail.

Tout le personnel de l'administration de l'université 8 Mai 1945 de Guelma, plus particulièrement au corps professoral du département de l'informatique.

Résumé

L'algorithmique est l'étude de production de règles et techniques qui sont impliquées dans la définition et la conception d'algorithmes, c'est-à-dire de processus systématiques de résolution de problème permettant de décrire précisément des étapes de résoudre un problème algorithmique.

Après avoir trouvé la solution algorithmique, nous devons l'écrire en utilisant un langage de programmation, les étudiants souffrent dans cette étape de conversion vers un code source.

Notre objectif est de permettre des non-programmeurs et aux programmeurs débutants d'écrire des programmes informatiques, avec le principe d'apprentissage des concepts de base qui lui sont associés.

Dans ce mémoire, nous avons décrit notre projet qui se focalise sur l'aide surtout des étudiantes en conversion d'un algorithme (plus précisément un organigramme) en code source Java.

Mots clés : algorithme, organigramme, programmation, java, code source, apprentissage, apprentissage assisté par ordinateur, enseignement assisté par ordinateur, enseignant, apprenant, évaluation.

Abstract

Algorithmic is the study of the production of rules and techniques that are involved in the definition and design of algorithms, that is to say, systematic processes of resolution of problems allowing describing precisely steps to solve an algorithmic problem.

If we have the algorithmic solution, we must write it in programming language, the students suffer in this step of conversion to a source code.

Our goal is to allow non-programmers and novice programmers to write computer programs, with the principle of learning the basic concepts associated with it.

In this brief, we describe our project that focuses on helping students in the conversion of an algorithm (specifically a flowchart) into Java source code.

Keywords: algorithm, organization chart, programming, java, source code, learning, computer-aided learning, computer-assisted learning, teacher, learner, evaluation.

المخلص

الخوارزمية هي دراسة قواعد وأساليب الإنتاج التي تشارك في تحديد وتصميم الخوارزميات، بتعريف آخر، هي منهجية عملية لحل مشكلة وبالتالي وصف بدقة الخطوات من أجل حل مشكلة الخوارزمية.

بعد العثور على حل الخوارزمية، بعدها نحن بحاجة لكتابته باستخدام لغة البرمجة، والطلاب يعانون في هذه الخطوة التي يحول فيها الحل إلى برنامج تنفيذي.

هدفنا هو تمكين غير المبرمجين والمبرمجين المبتدئين من كتابة برامج الكمبيوتر، مع مبدأ تعلم المفاهيم الأساسية المرتبطة به.

في هذه المذكرة، هدف مشروعنا الأساسي هو مساعدة الطلاب في تحويل الخوارزمية (على وجه التحديد مخطط خوارزمية) إلى برنامج بلغة جافا.

كلمات مفتاحية: الخوارزمية، مخطط خوارزمية، برمجة، جافا، التعلم، التعلم القائم على الحاسوب، التعليم بمساعدة الحاسوب، التدريس، التعلم، التقييم.

TABLE DES MATIERES

إهداء.....	I
DÉDICACE.....	II
REMERCIEMENTS	III
RÉSUMÉ	IV
ABSTRACT	V
الملخص	VI
TABLE DES MATIERES	1
TABLE DES ILLUSTRATIONS.....	4
LISTE DES FIGURES	4
LISTE DES TABLEAUX	5
INTRODUCTION GENERALE	6
CONTEXTE ET CADRE DE LA RECHERCHE.....	6
OBJECTIF ET APPROCHE.....	6
PLAN DU MEMOIRE.....	6
PARTIE – I : ETUDE DE L’EXISTANT.....	8
CHAPITRE 1 : LES ALGORITHMES.....	9
1. INTRODUCTION	9
2. NOTION D’ALGORITHME	9
3. HISTORIQUE	9
4. CARACTERISTIQUES D’UN ALGORITHME	10
5. LES PROBLEMES FONDAMENTAUX EN ALGORITHMIQUE	10
6. REPRESENTATION D’UN ALGORITHME	11
7. ORGANIGRAMMES ET PSEUDO-CODE	11
7.1. PSEUDO-CODE.....	11
7.2. ORGANIGRAMME	13
8. ORGANIGRAMME VS PSEUDO CODE	14
9. EXEMPLES D’OUTILS QUI FAIRE LA CONVERSION ALGORITHME CODE SOURCE	16
10. SYNTHESE	18
11. CONCLUSION.....	18
CHAPITRE 2 : ENSEIGNEMENT ASSISTE PAR ORDINAEUR	19
1. INTRODUCTION	19
2. NOTION DE BASE	19
2.1. APPRENTISSAGE	19
2.1.1. <i>Définition de l'apprentissage</i>	<i>19</i>
2.1.2. <i>Les trois approches principales de l'apprentissage</i>	<i>19</i>
2.1.3. <i>Apprentissage assisté par ordinateur</i>	<i>21</i>
2.2. L’ENSEIGNEMENT ASSISTE PAR ORDINATEUR (EAO)	21
2.2.1. <i>Définition de l’enseignement</i>	<i>21</i>
2.2.2. <i>Définition de l’enseignement assisté par ordinateur</i>	<i>22</i>
2.2.3. <i>Définition des didacticiels</i>	<i>22</i>

3.	FONCTIONS D'ORDINATEUR DANS L'ENSEIGNEMENT	23
4.	DIFFERENTS TYPES DE LOGICIELS E.A.O	24
5.	DES EXEMPLES D'E.A.O	25
6.	NIVEAUX D'E.A.O	25
7.	PRINCIPES D'APPRENTISSAGE E.A.O.....	25
8.	LOGICIELS D'ENSEIGNEMENT ASSISTE PAR ORDINATEUR	26
9.	CATEGORIES D'ENSEIGNEMENT ASSISTE PAR ORDINATEUR.....	28
10.	TECHNIQUE D'APPRENTISSAGE.....	30
11.	LES AVANTAGES DE L'E.A.O.....	30
12.	LES INCONVENIENTS DE L'E.A.O	30
13.	CONCLUSION	31
	PARTIE – II : NOTRE PROPOSITION.....	32
	CHAPITRE 3 : CONCEPTION.....	33
1.	INTRODUCTION	33
2.	L'OBJECTIF DE NOTRE SYSTEME	33
3.	MECANISME DE CONVERSION.....	33
4.	ARCHITECTURE GLOBAL DU SYSTEME	35
5.	PROCESSUS D'APPRENTISSAGE.....	37
6.	CONCEPTION DETAILLEE	38
6.1.	CHOIX DE LA METHODE	38
6.2.	DIAGRAMMES DE CAS D'UTILISATION	38
6.3.	DIAGRAMME DE SEQUENCE	40
6.4.	DIAGRAMME DE CLASSE	40
7.	CONCLUSION	41
	CHAPITRE 4 : IMPLEMENTATION.....	42
1.	INTRODUCTION	42
2.	OUTILS ET LANGAGE DE DEVELOPPEMENT.....	42
2.1.	LANGAGE DE PROGRAMMATION	42
2.1.1.	<i>Eclipse néon 3</i>	<i>42</i>
2.1.2.	<i>XAMPP.....</i>	<i>43</i>
2.2.	LES OUTILS	43
2.2.1.	<i>JDOM.....</i>	<i>43</i>
2.2.2.	<i>JGraphX - Bibliothèque Java Swing</i>	<i>44</i>
3.	DESCRIPTION DU SYSTEME NOTRE SYSTEME	44
3.1.	INTERFACE DE CONNEXION	44
3.2.	INTERFACE DES OBJETS D'ORGANIGRAMME.....	45
3.2.1.	<i>Comment</i>	<i>46</i>
3.2.2.	<i>Input</i>	<i>46</i>
3.2.3.	<i>Output.....</i>	<i>47</i>
3.2.4.	<i>Declare.....</i>	<i>47</i>
3.2.5.	<i>Assign.....</i>	<i>48</i>
3.2.6.	<i>If.....</i>	<i>48</i>
3.2.7.	<i>Call.....</i>	<i>49</i>
3.2.8.	<i>While</i>	<i>49</i>

Table des Matières

3.2.9. For.....	50
4. PARTIE ENSEIGNANT	50
4.1. MANIPULATIONS.....	52
4.1.1. Interface de gestion des sessions.....	52
4.1.2. Interface de gestion des niveaux	53
4.2. PEDAGOGIES	53
4.2.1. Interface de gestion des cours	53
4.2.2. Interface de gestion des exercices	54
4.2.3. Interface de gestion des méthodes.....	55
5. PARTIE ETUDIANT	56
5.1. INTERFACE DE S'ENTRAINER	56
5.2. INTERFACE DE CONSULTATION DES COURS D'UN NIVEAU	57
5.3. INTERFACE DE RESOLUTION DES EXERCICES D'UN NIVEAU	58
5.4. INTERFACE DE CHOIX D'UN NIVEAU	59
6. CONCLUSION	60
CONCLUSION GENERALE	61
BIBLIOGRAPHIE	62

TABLE DES ILLUSTRATIONS

LISTE DES FIGURES

<i>FIGURE 1-1 : EXEMPLE D'UN ORGANIGRAMME</i>	13
<i>FIGURE 2-1 : LE MODELE TRANSMISSIF</i>	20
<i>FIGURE 2-2 : LE MODELE BEHAVIORISTE</i>	20
<i>FIGURE 2-3 : LE MODELE SOCIOCONSTRUCTIVISTE</i>	21
<i>FIGURE 2-4 : TRIANGLE DIDACTIQUE(DE HOUSSAYE) [26]</i>	23
<i>FIGURE 2-5 : LOGICIEL PEPIT</i>	27
<i>FIGURE 2-6 : LOGICIEL EDU'KEY</i>	27
<i>FIGURE 2-7 : LOGICIEL FORUM RETZ</i>	27
<i>FIGURE 2-8 : LOGICIEL CAHIERS PEDAGOGIQUES</i>	28
<i>FIGURE 3-1 : ARCHITECTURE GLOBALE DU SYSTEME</i>	36
<i>FIGURE 3-2 : PROCESSUS D'APPRENTISSAGE</i>	38
<i>FIGURE 3-3 : DIAGRAMME DE CAS D'UTILISATION DES DEUX ACTEURS</i>	39
<i>FIGURE 3-4 : DIAGRAMME DE CLASSES</i>	40
<i>FIGURE 4-1 : INTERFACE D'ACCES</i>	45
<i>FIGURE 4-2 : INTERFACE DES OBJETS D'ORGANIGRAMME</i>	45
<i>FIGURE 4-3 : INTERFACE DE COMMENT</i>	46
<i>FIGURE 4-4 : INTERFACE D'INPUT</i>	46
<i>FIGURE 4-5 : INTERFACE D'OUTPUT</i>	47
<i>FIGURE 4-6 : INTERFACE DE DECLARER</i>	47
<i>FIGURE 4-7 : INTERFACE D'ASSIGNER</i>	48
<i>FIGURE 4-8 : INTERFACE D'IF</i>	48
<i>FIGURE 4-9 : INTERFACE DE CALL</i>	49
<i>FIGURE 4-10 : INTERFACE DE CONDITION DE WHILE</i>	49
<i>FIGURE 4-11 : INTERFACE DE CONDITION DE FOR</i>	50
<i>FIGURE 4-12 : INTERFACE DE PARTIE D'ENSEIGNANT</i>	50
<i>FIGURE 4-13 : INTERFACE DE MENU DE MANIPULATIONS</i>	51
<i>FIGURE 4-14 : INTERFACE DE MENU DE PEDAGOGIES</i>	51
<i>FIGURE 4-15 : INTERFACE DE GESTION DES SESSIONS</i>	52
<i>FIGURE 4-16 : INTERFACE DE GESTION DES NIVEAUX</i>	53
<i>FIGURE 4-17 : INTERFACE DE GESTION DES COURS</i>	54
<i>FIGURE 4-18 : INTERFACE DE GESTION DES EXERCICES</i>	54
<i>FIGURE 4-19 : INTERFACE DE GESTION DES METHODES</i>	55
<i>FIGURE 4-20 : INTERFACE DE PARTIE D'ETUDIANT</i>	56
<i>FIGURE 4-21 : INTERFACE D'ENTRAINEMENT</i>	57
<i>FIGURE 4-22 : INTERFACE DE CHOIX D'UN COURS</i>	57
<i>FIGURE 4-23 : INTERFACE DE COURS CHOISI</i>	58
<i>FIGURE 4-24 : INTERFACE DE CHOIX D'UN EXERCICE</i>	58
<i>FIGURE 4-25 : INTERFACE D'UN EXERCICE CHOISI</i>	59
<i>FIGURE 4-26 : INTERFACE DE PASSAGE VERS LE NIVEAU SUIVANT</i>	59
<i>FIGURE 4-27 : INTERFACE DE CHOIX D'UN NIVEAU</i>	59

LISTE DES TABLEAUX

TABLEAU 1-1 : RECAPITULATIF DES MOTS ET SYMBOLES DU PSEUDO-CODE. 12
TABLEAU 1-2: SYMBOLES DE TRAITEMENT..... 13
TABLEAU 1-3 : ORGANIGRAMME VS PSEUDO-CODE. 16
TABLEAU 1-4 : LOGICIELS QUI FONT LA CONVERSION. 17

TABLEAU 2-1 : FONCTIONS D'ORDINATEUR DANS L'ENSEIGNEMENT (DEVRIES (2001))..... 24

TABLEAU 3-1 : MECANISME DE CONVERSION 35

Introduction générale

Contexte et cadre de la recherche

L'apprentissage est un ensemble des processus de mémorisation mis en œuvre par l'être humain pour élaborer ou modifier les schémas comportementaux spécifiques sous l'influence de son environnement et de son expérience.

Les fondements de l'EAO remontent au milieu des années 1980, avec l'avènement et le développement des premiers ordinateurs personnels, des premiers logiciels éducatifs et de périphériques d'interaction homme-machine spécialisés comme la souris ou le crayon optique. Les sciences cognitives réunissent des domaines qui permettent de faire évoluer un tel enseignement.

Les cours d'introduction à la programmation sont tristement remarquables pour leur taux d'échec – ou d'abandon (qui s'échelonne entre 25 et 80% de par le monde, D'après J. Kaasboll [1]).

Objectif et approche

L'objectif de notre travail est de réaliser un outil informatique pédagogique (milieu d'apprentissage) qui permet de maximiser l'apprentissage des concepts de base de la programmation (en Java) et d'augmenter les chances de réussite pour tous les étudiants, peu importe leurs niveau d'habileté ou leurs base de connaissances.

Dans ce cadre de travail nous s'intéresse à l'aide des étudiants débutants en programmation de conversion d'algorithme vers code source Java.

Plan du mémoire

Notre mémoire est organisé en quatre chapitres de la manière suivante :

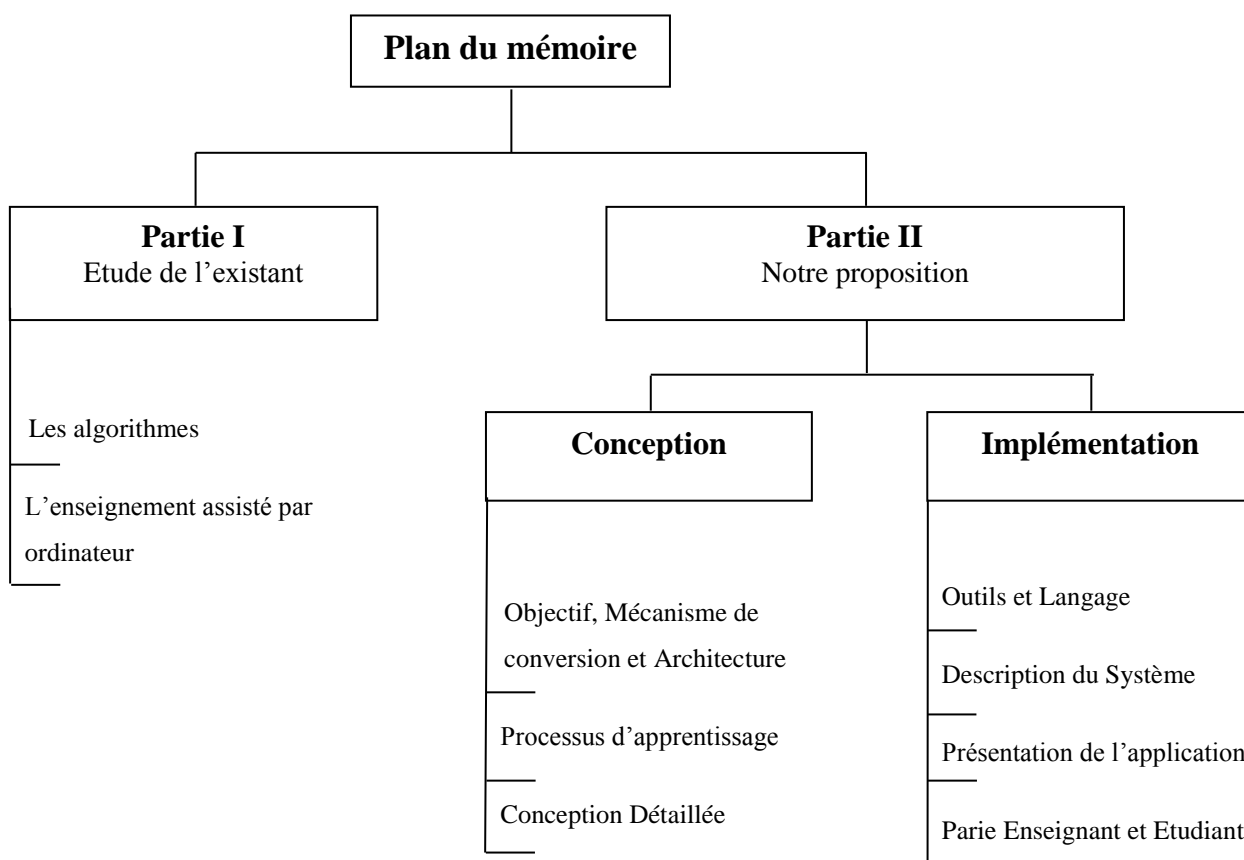
Dans la partie I, contient deux chapitres :

- Le 1er chapitre présente une étude sur les algorithmes et leurs différentes représentations.
- Le 2ème chapitre donne un aperçu global sur l'enseignement et l'apprentissage assisté par ordinateur.

La partie II : contient aussi deux chapitres :

- Le 3ème chapitre expose l'étude conceptuelle de notre application.
- Le dernier chapitre présente les outils de développement, et le langage de programmation utilisé pour développer notre application, et détaille les différentes étapes jusqu'à l'obtention de notre outil

En fin, nous achevons notre mémoire par une conclusion et quelques perspectives de notre travail



PARTIE – I : ETUDE DE L'EXISTANT

CHAPITRE 1 : LES ALGORITHMES

1. INTRODUCTION

Tout comme l'homme, la machine calcule ! Si vous avez déjà ouvert un livre de recettes de cuisine ou déchiffré un mode d'emploi traduit directement du coréen pour faire fonctionner un magnétoscope ou un répondeur téléphonique réticent, sans le savoir, vous avez déjà exécuté des algorithmes. Les algorithmes sont des outils pour la résolution de problèmes. Évaluer leur efficacité permet de classer ces problèmes selon leur difficulté intrinsèque.

Nous allons dans un premier temps définir précisément ce qu'est un algorithme. Nous présentons ensuite les différentes représentations d'algorithme.

2. NOTION D'ALGORITHME

- **Définition**

- **Algorithme :**

- Un algorithme est un ensemble de règles opératoires rigoureuses, ordonnant à un processeur d'exécuter dans un ordre déterminé une succession d'opérations élémentaires, pour résoudre un problème donné. [2]
- Procédure de calcul bien définie qui prend en entrée une valeur, ou un ensemble de valeurs, et qui donne en sortie une valeur, ou un ensemble de valeurs. Un algorithme est donc une séquence d'étapes de calcul qui transforment l'entrée en sortie. [3]
 - Exemple d'algorithmes
 - ✓ Recette de cuisine.
 - ✓ Notice de montage de meuble en kit.
 - ✓ calcule la puissance d'un nombre.
 - ✓ les calculs différentielsetc.
- **L'algorithmique :** est étude de la résolution de problèmes par la mise en œuvre de suites d'opérations élémentaires selon un processus défini aboutissant à une solution. [4]

3. HISTORIQUE

La signification du mot «algorithme» provient du nom d'un mathématicien perse ayant vécu au environ des années 820 après Jésus-Christ du nom «Al Khwarizmi».

Bien que l'histoire semble assez troué sur le sujet et qu'un algorithme n'est pas systématique attribuable à un ordinateur. On peut quand même déduire que la machine développée dans les années 1930, du nom de Turing, est en quelque sorte le point de départ concret des algorithmes. En effet, celle-ci permettait d'effectuer tous les algorithmes modernes.

Par la suite, après la deuxième guerre mondiale, les américains investirent beaucoup d'effort pour permettre de passer d'un code machine, puis l'assembleur (1950) pour atteindre un niveau assez proche des pseudo-codes d'algorithmes. En somme, il y eut plus de 3 000 langages imaginés entre les 1970 et 1980 et le nombre n'a jamais eu de cesse d'augmenter depuis.

Maintenant, les algorithmes purs sont surtout utilisés dans le domaine de l'enseignement et de la pédagogie. [5]

4. CARACTERISTIQUES D'UN ALGORITHME

L'algorithme est un moyen pour le programmeur de présenter son approche du problème à d'autres personnes. En effet, un algorithme est l'énoncé dans un langage bien défini d'une suite d'opérations permettant de répondre au problème. Un algorithme doit donc être : [6]

- lisible : l'algorithme doit être compréhensible même par un non-informaticien,
- de haut niveau : l'algorithme doit pouvoir être traduit en n'importe quel langage de programmation, il ne doit donc pas faire appel à des notions techniques relatives à un programme particulier ou bien à un système d'exploitation donné,
- précis : chaque élément de l'algorithme ne doit pas porter à confusion, il est donc important de lever toute ambiguïté,
- concis : un algorithme ne doit pas dépasser une page. Si c'est le cas, il faut décomposer le problème en plusieurs sous-problèmes,
- structuré : un algorithme doit être composé de différentes parties facilement identifiables.

5. LES PROBLEMES FONDAMENTAUX EN ALGORITHMIQUE

- complexité
 - en combien de temps un algorithme va-t-il atteindre le résultat escompté ?
 - De quel espace a-t-il besoin ?
- calculabilité
 - existe-t-il des tâches pour lesquelles il n'existe aucun algorithme ?
 - Etant donnée une tâche, peut-on dire s'il existe un algorithme qui la résolve ?

- correction

- peut-on être sûr qu'un algorithme réponde au problème pour lequel il a été conçu ? [7]

6. REPRESENTATION D'UN ALGORITHME

Un algorithme historiquement, deux façons pour représenter un algorithme :

L'organigramme: représentation graphique avec des symboles (carrés, losanges, etc.).

- offre une vue d'ensemble de l'algorithme
- représentation quasiment abandonnée aujourd'hui

Le pseudo-code : représentation textuelle avec une série de conventions ressemblant à un langage de programmation.

- plus pratique pour écrire un algorithme
- représentation largement utilisée. [8]

7. ORGANIGRAMMES ET PSEUDO-CODE

7.1.Pseudo-code

- **Définition :** En programmation, le pseudo-code est une façon de décrire un algorithme sans référence à un langage de programmation en particulier.

L'écriture en pseudo-code permet souvent de bien prendre toute la mesure de la difficulté de la mise en œuvre de l'algorithme, et de développer une démarche structurée dans la construction de celui-ci. En effet, son aspect descriptif permet de décrire avec plus ou moins de détail l'algorithme, permettant de ce fait de commencer par une vision très large et de passer outre temporairement certains aspects complexes, ce que n'offre pas la programmation directe. [9]

- **Structure de pseudo code**

Un algorithme a à peu près toujours la même organisation générale. Trois grandes parties

Se distinguent :

Le nom de l'algorithme

Partie déclaration: déclaration des variables et des constantes.

Partie traitement: actions, conditions... [10]

➤ **Tableau récapitulatif des mots et symboles du pseudo-code : [11]**

Mots et symboles du pseudo-code	Opérations réalisées
Début	Début de l'algorithme, permet de le nommer
Fin	Fin de l'algorithme
Faire	Exécution d'une opération
Entrer	Acquisition ou chargement d'une donnée
Sortir	Edition ou sauvegarde d'un résultat
←	Affectation d'une valeur à une variable
Symboles d'opérateur	Opérations arithmétiques ou logiques
Aller à	Branchement inconditionnel
Si...alors...[sinon]	Branchement inconditionnel
Selon cas...[autrement]	Branchement conditionnel généralisé
Itérer...sortir si...	} Répétition conditionnelle
Tant que...faire...	
Répéter...jusqu'à ce que...	
Pour...de...à...	Répétition contrôlée

Tableau 1-1 : Récapitulatif des mots et symboles du pseudo-code.➤ **Exemple:** calcul de $1+2+\dots+n$ (n entier ≥ 1 fixé).

```

Programme qui fait la somme des n premiers Termes
// partie déclaration
n : entier; s : entier; i : entier; (ou n, i, s : entier)
Début
// Lecture des données
Écrire (" n = ? «);
lire(n);
// calcul de la somme
s := 0;
pour i := 1 à n faire s := s + i;
fpour;
// affichage du résultat
écrire("1+ 2 + ... + n = ", s);
Fin

```

➤ **Avantage :**

Cette forme est adaptée au langage de programmation structurée (langage C) à la description de procédures administratives et au langage de macro-commandes.[13]

7.2.Organigramme

➤ **Définition:** l'organigramme est la représentation graphique de l'algorithme, il permet de représenter chaque opération élémentaire au moyen d'un symbole graphique normalisé. [11]

➤ **Symboles de traitement : La norme ISO 5807.** [12]

Symbole	Désignation	Symbole	Désignation
SYMBOLES DE TRAITEMENT			
1)	Symbole général "traitement" Opération ou groupe d'opérations sur des données, instructions, etc., ou opération pour laquelle il n'existe aucun symbole normalisé.	6)	Mode synchrone ; mode parallèle Ce symbole est utilisé lorsque plusieurs instructions doivent être exécutées simultanément.
2)	Fonction ou sous-programme Portion de programme considérée comme une simple opération.	7)	SYMBOLES AUXILIAIRES Renvoi Symbole utilisé deux fois pour assurer la continuité lorsqu'une partie de ligne de liaison n'est pas représentée.
3)	Entrée - Sortie : Mise à disposition d'une information à traiter ou enregistrement d'une information traitée.	8)	Début, fin, interruption Début, fin ou interruption d'un organigramme, point de contrôle, etc..
4)	Préparation Opération qui détermine partiellement ou complètement la voie à suivre dans un embranchement ou un sous-programme. Symbole également utilisé pour préparer une décision ou mettre un aiguillage en position.	9)	Commentaire Symbole utilisé pour donner des indications marginales.
3)	SYMBOLES LOGIQUES Embranchement Exploitation de conditions variables impliquant le choix d'une voie parmi plusieurs. Symbole couramment utilisé pour représenter une décision ou un aiguillage.	Sens conventionnel des liaisons : Le sens général des lignes doit être : - de haut en bas - de gauche à droite. Lorsque le sens ainsi défini n'est pas respecté, des pointes de flèches, à cheval sur la ligne, indiquent le sens utilisé.	
	NON		OUI

Tableau 1-2: Symboles de traitement

➤ **Exemple :**

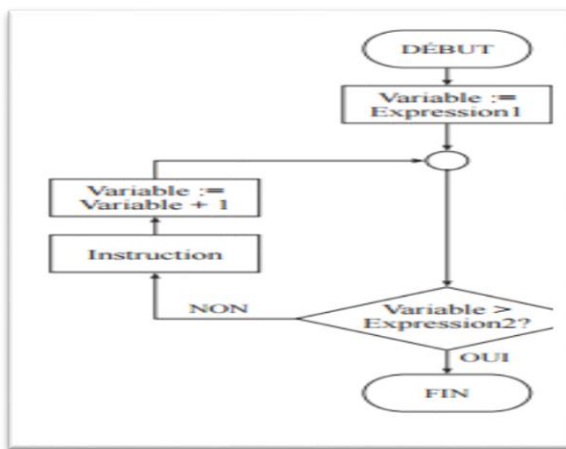
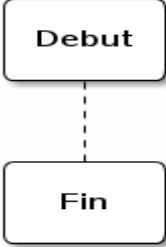

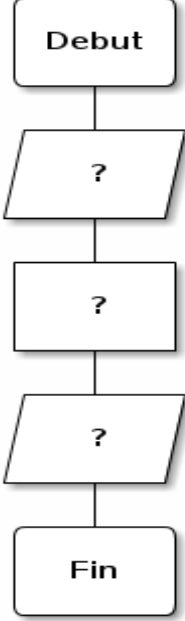
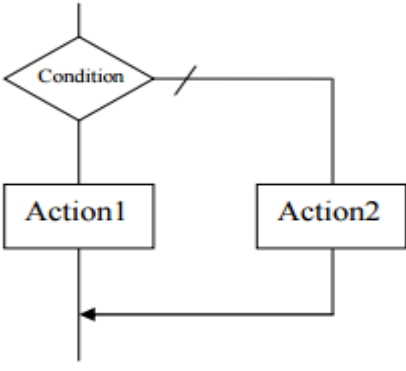
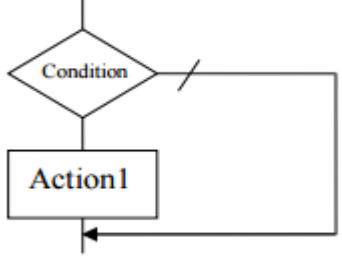
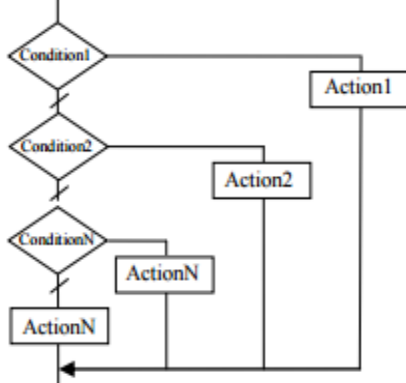
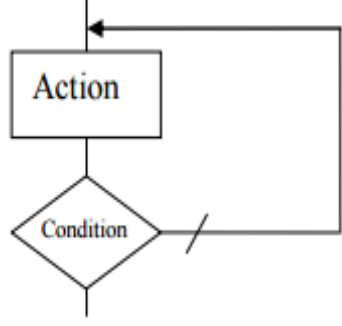
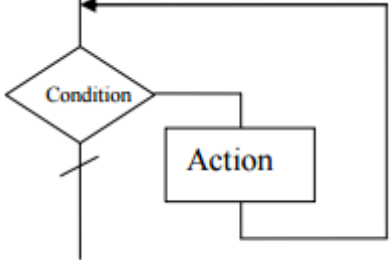


Figure 1-1 : Exemple d'un organigramme

- **Avantages** : Visualisation du traitement
- **Inconvénients** : devient vite fouillis, désuet. [13]

8. ORGANIGRAMME VS PSEUDO CODE

STRUCTURES ALGORITHMIQUE ELEMENTAIRES	ALGORIGRAMME	PSEUDO-CODE
1. Début et Fin		Début algorithme MON_ALGO ... Fin algorithme MON_ALGO
2. Entrées et Sorties		Demander... Afficher...
3. Séquence linéaire		Début algorithme MON_ALGO ... Entrée-sortie ... Traitement ... Entrée-sortie ... Fin algorithme MON_ALGO

<p>3. La structure conditionnelle ou alternative complète :</p>		<p>si condition alors Action1 ; Sinon Action2 ; fsi ;</p>
<p>4. Une structure alternative réduite :</p>		<p>si condition alors Action ; fsi ;</p>
<p>5. Caractéristique de la structure de choix</p>		<p>suivant valeur faire valeur1 : action1 ; valeur2 : action2 ; valeur N : action N ; sinon action N+1 fsuivant ;</p>
<p>6. Caractéristique de la structure itérative :</p>		<p>répéter action ; Jusqu'à condition vraie ;</p>
<p>7. Structure TANT QUE ... FAIRE ...</p>		<p>tant que condition faire action ; Ftant que ;</p>

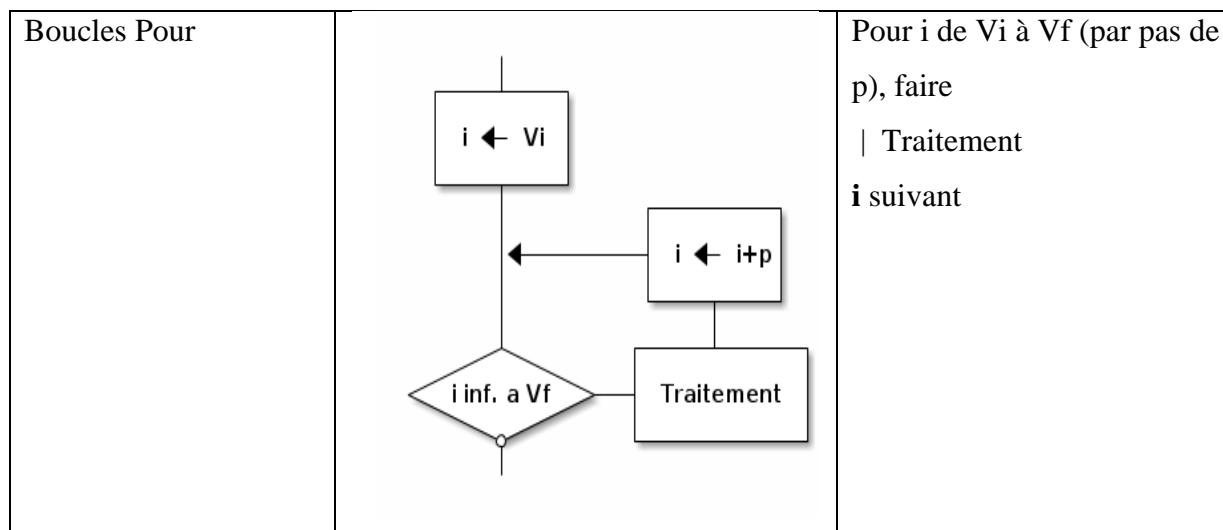


Tableau 1-3 : Organigramme Vs Pseudo-code.

9. EXEMPLES D'OUTILS QUI FAIRE LA CONVERSION ALGORITHME CODE SOURCE

Logiciel	Description	Fonctionnement
1. Algobox	- AlgoBox est un logiciel libre, et gratuit d'aide à l'élaboration et à l'exécution d'algorithmes dans l'esprit des nouveaux programmes de mathématiques du lycée. [16]	-Le code de l'algorithme se construit pas à pas de façon hiérarchique et structurée grâce à des instructions de base que l'on insère en cliquant sur les boutons de l'interface : l'utilisateur se concentre ainsi sur l'algorithme lui-même et il est fortement incité, par le fonctionnement même du programme, à faire preuve d'un minimum de rigueur.[16]
2.flowgorithm	-Flowgorithm est une libre application qui vous aide à créer des programmes à l'aide d'organigramme. [15]	les programmes sont écrits en utilisant Flowgorithm (algorithme), vous pouvez utiliser des formes pour représenter les différentes actions que vous voulez que

		votre programme à exécuter. [15]
3. AthTek flowchart to code.	- Flowchart au code est un outil de programmation d'accompagnement du Code Ath Tek à FlowChartConverter. Il peut aider les ingénieurs logiciels pour convertir organigramme au code sans effort. [14]	- la seule chose que vous devez faire est de mettre l'organigramme de programme de votre esprit dans le programme, puis vous obtiendrez le code source automatiquement. Il ne peut absolument accélérer les cycles de développement et les ingénieurs de logiciels libres du travail répétitif et mécanique. -AthTekprend en charge pour générer du code source en plusieurs langues , y compris C, C ++, C #, Java, JavaScript et Delphi. [14]
4. larp	-LARP est en fait un acronyme. Il vient de la compression de la phrase «Logiciel d'Algorithmes et de Résolution de Problèmes», conçu par Marco Lavoie. LARP est un langage de programmation permettant le prototypage rapide d'algorithmes.[17]	-LARP offre un environnement de développement d'algorithmes simple et convivial permettant à un utilisateur novice de se familiariser rapidement avec le logiciel. L'avantage de LARP est offre à la possibilité de formuler des algorithmes sous forme d'organigrammes ou pseudo code. [17]

Tableau 1-4 : Logiciels qui font la conversion.

10.SYNTHESE

Tous les outils cités précédemment offrent la traduction des algorithmes (organigramme ou pseudo-code) vers des programmes en plusieurs langages de programmation, mais ils ont tous un problème d'absence de l'apprentissage.

11.CONCLUSION

Dans ce chapitre nous avons parlé sur l'algorithmique qui intervienne dans notre vie quotidienne. Par la suite, nous avons un chapitre sur l'apprentissage, l'apprentissage assisté par ordinateur et les différentes techniques existantes.

CHAPITRE 2 : ENSEIGNEMENT ASSISTE PAR ORDINAEUR

1. INTRODUCTION

L'apprentissage assisté par ordinateur est l'avenir, et que l'avenir est maintenant. L'éducation, en tant que processus et la discipline, est principalement concerné par la transmission des connaissances, des méthodes d'enseignement, et de fournir / le maintien d'un environnement d'apprentissage conducteur, par opposition à l'éducation informelle et d'autres moyens de socialisation. Dans ce chapitre nous allons essayer de présenter l'apprentissage et l'enseignement assisté par ordinateur.

2. NOTION DE BASE

2.1.Apprentissage

2.1.1. Définition de l'apprentissage

- « Processus, activité ou ensemble d'activités qui permettent à une personne de développer des compétences ou d'acquérir des connaissances à l'intérieur d'un cours ou d'un programme de formation. » (Grand Dictionnaire Terminologique).

- En première approximation, on peut considérer l'apprentissage comme une modification stable et durable des savoirs, des savoir-faire ou des savoir-être d'un individu, modification attribuable à l'expérience, à l'entraînement, aux exercices pratiqués par cet individu.[18]

2.1.2. Les trois approches principales de l'apprentissage

Les trois modèles(Approche) décrits ci-dessous sont autant de manières d'envisager les processus d'apprentissage. [19]

A. Approche transmissive (Le savoir est transmis par l'enseignant)

C'est le modèle qui correspond à la conception empirique de l'enseignement. Il est basé sur deux présupposés:

- La neutralité conceptuelle de l'apprenant.
- La non-déformation du savoir transmis.
- **Avantages :** L'enseignement basé sur ce modèle est le plus économe en temps et en moyens.
- **Limites :** Elles dépendent de la validité des deux présupposés :

- 1) Si une conception initiale inadéquate existe elle risque de ne pas être remise en cause, et d'interférer avec la nouvelle connaissance.
- 2) Ce qui est dit par l'enseignant n'est pas toujours entendu de la même façon par tous les apprenants.

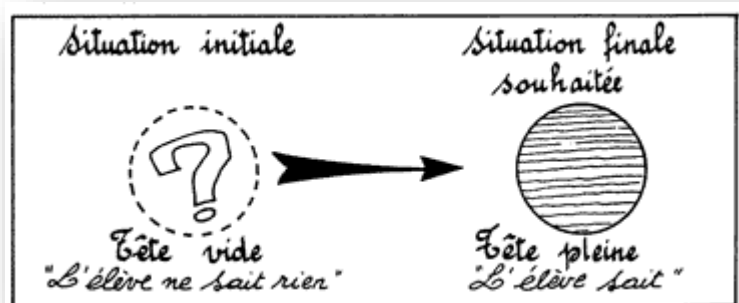


Figure 2-1 : Le modèle transmissif

B. Approche behavioriste (Le savoir est découvert par l'élève)

L'objectif d'apprentissage est décomposé en sous-objectifs formulés en termes de comportements observables (l'apprenant est capable de ... plutôt que l'apprenant connaît...).

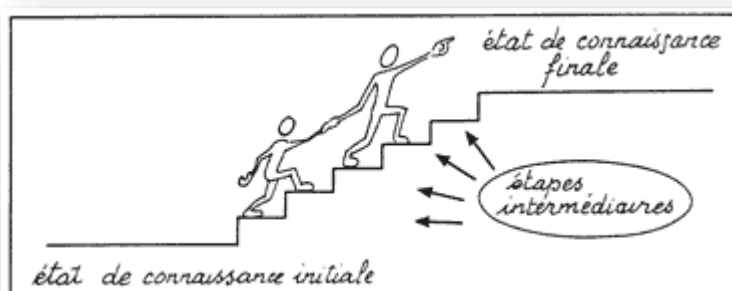


Figure 2-2 : Le modèle behavioriste

- **Avantages :** L'enseignant est attentif aux possibilités et à l'évolution individuelle de l'apprenant.
 - L'apprenant peut progresser à son rythme.
- **Limite:**
 - Les tâches découpées cachent la vision d'ensemble.
 - D'autre part, les conceptions initiales n'étant pas prises en compte, elles sont susceptibles de ressurgir lorsque l'apprenant se trouvera devant un problème plus complexe.

C. Approche socioconstructiviste (Le savoir est construit par l'élève)

Il repose sur les trois présupposés suivants :

- C'est en agissant (en résolvant des problèmes) que l'on apprend.
- Les représentations initiales s'érigent souvent en obstacle aux nouvelles connaissances.
- La connaissance ne s'acquiert pas par simple empilement.

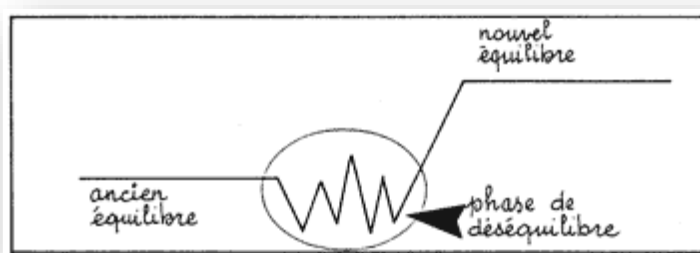


Figure 2-3 : Le modèle socioconstructiviste

- **Avantages :** L'apprenant est confronté à un problème à résoudre, ce qui lui permet démettre du sens à son apprentissage. Les conceptions initiales inadéquates ayant été détruites ou remodelées d'équilibre est durable.
- **Limite:**
 - 1) L'enseignement basé sur ce modèle est coûteux en temps.
 - 2) Il nécessite un haut niveau de compétence de l'enseignant.
 - 3) Il est parfois difficile de trouver des situations-problèmes adéquates.
 - 4) La phase de déstabilisation est délicate chez certains apprenants. [19]

2.1.3. Apprentissage assisté par ordinateur

L'apprentissage assisté par ordinateur (AAO) est un terme qui englobe généralement trois utilisations principales de l'ordinateur à des fins éducatives et formatives. [20]

2.2.L'Enseignement Assisté par Ordinateur (EAO)

2.2.1. Définition de l'enseignement

Est une pratique mise en œuvre par un enseignant visant à transmettre des connaissances (savoir, savoir-faire, compétences...) à un apprenant. Cette notion se distingue de l'apprentissage qui renvoie lui à l'activité de l'apprenant qui s'approprie les connaissances.[19]

2.2.2. Définition de l'enseignement assisté par ordinateur

Est une spécialité informatique qui regroupe les logiciels permettant l'aide à l'apprentissage dans des domaines divers, ainsi que les outils utilisés pour créer ces programmes.[22]

L'EAO constitue le sous-ensemble des applications pédagogiques de l'ordinateur comportant le recours à des **didacticiels**, c'est-à-dire à des logiciels spécifiquement conçus pour l'enseignement.[21]

2.2.3. Définition des didacticiels

Les logiciels didactiques d'EAO sont appelés didacticiels. Il en existe différents types (ex: Micro-monde, Tuteur Intelligent).

La didactique

Le regard didactique est centré sur l'apprenant confronté au savoir. Il est question de la dérivation du regard qui a prévalu longtemps en éducation et qui considérait que la relation enseignant-enseigné expliquait quasiment à elle seule les questions d'appropriation du savoir ; ce n'est plus la relation professeur- élève qui est centrale ici mais l'appropriation du savoir par l'apprenant. L'usage du substantif "apprenant", est pour marquer la volonté du regard didactique de considérer l'élève comme responsable de la conquête de ses connaissances. L'importance accordée actuellement à la didactique, peut se justifier d'une manière susceptible à travers ces trois hypothèses relevées par Michel Develay dans son ouvrage intitulé de l'apprentissage à l'enseignement :

- la didactique se développe parce que les savoirs se multiplient et la question de leur acquisition est aujourd'hui un point central dans le domaine scientifique.
- Parce qu'elle met l'accent sur l'acquisition des savoirs, la didactique a priori rebute moins les enseignants qui se disent propriétaires des contenus à enseigner.
- La didactique émergeait comme problématique en éducation après que les sciences de l'éducation ont éclaté en un grand nombre de domaines. Chacun d'eux, considérant comme son seul référent, l'explication de l'activité éducative dans son entier.[26]

Triangle didactique :

Jean Houssaye définit tout acte pédagogique comme l'espace entre trois sommets d'un triangle : l'enseignant, l'étudiant, le savoir.

-Derrière le savoir se cache le contenu de la formation : la matière, le programme à enseigner.

-L'enseignant est celui qui a quelques enjambées d'avance sur celui qui apprend et qui transmet ou fait apprendre le savoir.

-Quant à l'étudiant, il acquiert le savoir grâce à une situation pédagogique, mais ce savoir peut être aussi du savoir-faire, du savoir-être, du savoir agir, du faire savoir... [40]

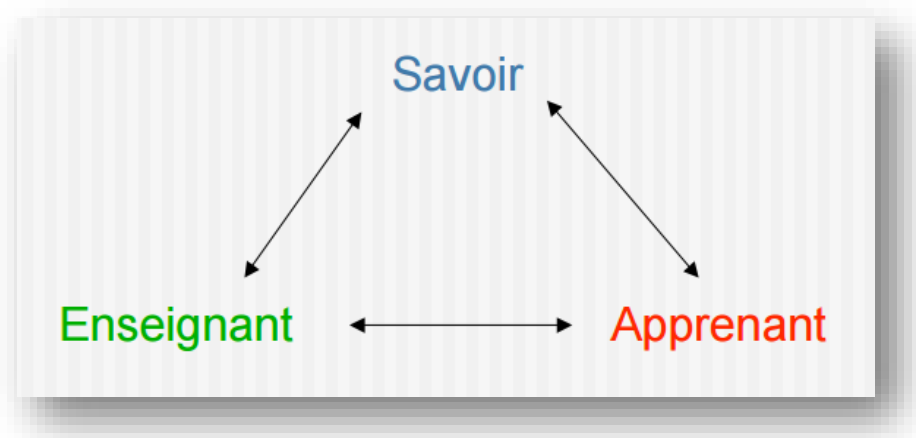


Figure 2-4 : Triangle didactique(de Houssaye) [26]

3. FONCTIONS D'ORDINATEUR DANS L'ENSEIGNEMENT

L'ordinateur possède huit fonctions dans l'enseignement (DeVries, 2001). Ces fonctions et leurs caractéristiques sont présentées au **Tableau 2-1**. [35]

Fonctions pédagogiques	Type de logiciel	Théorie	Tâches	Connaissances
Présenter de l'information	Tutoriel	Cognitivisme	Lire	Présentation ordonnée
Dispenser des exercices	Exercices répétés	Béhaviorisme	Faire des exercices	Association
Véritablement enseigner	Tuteur intelligent	Cognitivisme	Dialogue	Représentation
Captiver l'attention et la motivation de l'élève	Jeu éducatif	Principalement béhaviorisme	Jouer	
Fournir un espace d'exploration	Hypermédia	Cognitivisme Constructivisme	Explorer	Présentation en accès libre
Fournir un environnement pour la découverte de lois naturelles	Simulation	Constructivisme Cognition située	Manipuler, observer	Modélisation
Fournir un environnement pour la découverte de domaines abstraits	Micro-monde	Constructivisme	Construire	Matérialisation
Fournir un espace d'échange entre élèves	Apprentissage collaboratif	Cognition située	discuter	Construction de l'élève

Tableau 2-1 : Fonctions d'ordinateur dans l'enseignement (DeVries (2001))

L'apport de l'ordinateur à l'enseignement et à l'apprentissage peut donc prendre des formes multiples.

L'ordinateur permet de maximiser le temps de formation et de favoriser la rétention (Desgent et al. 2004).

Il s'avère que l'information présentée sous forme multimédia stimule plusieurs sens à la fois et, par conséquent, favorise une meilleure rétention (Guay, 2001).

Tous les éléments qui favorisent une meilleure rétention peuvent être exploités lorsqu'on utilise les technologies de l'information et des communications (Huffman et Vernoy (2000)). [23]

4. DIFFERENTS TYPES DE LOGICIELS E.A.O

A) Les logiciels ludo-éducatifs : Ils demandent des ordinateurs récents pour fonctionner correctement. Créés pour une utilisation dans un cadre familial, ils sont souvent présents dans les CD thèques des écoles. Leur gros inconvénient est que les notions étudiées sont difficiles à sérier et souvent engluées dans une myriade d'effets prenant beaucoup de temps. La médiation du maître est indispensable pour les utiliser dans le cadre scolaire. De plus, il est difficile d'évaluer les compétences abordées et acquises lorsque cela est possible.

B) Les éducatifs purs abordent des domaines précis (français : lecture, conjugaison, grammaire. math : tables, numération, géométrie. histoire, géo, sciences.). Il en existe de toutes sortes : du très ancien fonctionnant sur des ordinateurs obsolètes au plus récent nécessitant des ordinateurs plus performants. Ils sont plus adaptés à un usage scolaire que les précédents, les exercices sont ciblés, la possibilité d'évaluation et de regroupements de résultats est souvent présente.

C) **Les logiciels encyclopédiques** ont permis une évolution de l'enseignement assistée par ordinateur par la possibilité d'effectuer des recherches documentaires sur un support différent du livre qu'il soit sur cédérom ou "en ligne" c'est-à-dire sur Internet. La qualité inégale des logiciels présents sur le marché doit inciter l'acheteur potentiel à en prendre préalablement connaissance. [30]

5. DES EXEMPLES D'E.A.O

- **Teachmaster**: est un programme entièrement gratuit sous Windows qui sert à l'apprentissage du vocabulaire dans des cours de langues.[31]

- **Cabri-Géomètre** : est un logiciel de géométrie dynamique destiné principalement à l'apprentissage de la géométrie en milieu scolaire. Il est commercialisé par la société CabriIgo et permet d'animer des figures géométriques, au contraire de celles dessinées au tableau. Il se décline pour la géométrie plane ou pour la géométrie en 3D. C'est l'ancêtre de tous les logiciels de géométrie dynamique.[32]

6. NIVEAUX D'E.A.O

L'EAO peut être aujourd'hui considérée sur deux niveaux :

- A) Un niveau strict : Il s'applique à l'utilisation de logiciels éducatifs (que nous appellerons didacticiels) organisant des parcours d'apprentissage individuels pour les apprenants.
- B) Un niveau large : L'utilisation des didacticiels auxquels il faut ajouter les logiciels de modélisation du type DAO (Dessin Assisté par Ordinateur), CAO (Conception Assistée par Ordinateur), etc. [33]

7. PRINCIPES D'APPRENTISSAGE E.A.O

L'EAO est axé sur l'apprentissage en solitaire, il est rassurant de savoir que l'approche, loin d'être intuitive, est balisée par neuf principes bien établis que nous reprenons sommairement ici:

- 1) la contiguïté stimulus-réponse;
- 2) la richesse des sollicitations;
- 3) l'usage adéquat de répétitions;
- 4) l'à-propos de l'interaction ou du renforcement;
- 5) la pertinence des indices utilisés;
- 6) une organisation originale du contenu et un usage judicieux des techniques de rappel;
- 7) la variété dans les stratégies de présentation;

- 8) le respect des différences individuelles;
- 9) l'évaluation constante des attitudes.

Connaissant ces principes, il devrait être plus facile pour le maître d'évaluer la qualité d'un didacticiel particulier et de sélectionner pour sa classe ceux qui ont le plus de chance de succès. [24]

8. LOGICIELS D'ENSEIGNEMENT ASSISTÉ PAR ORDINATEUR

1. Ordidac

Ordidac est un logiciel de création de séquences d'apprentissage et d'évaluations. Il permet de gérer pour une même séquence plusieurs types d'activités :

- QCM (Questionnaire à Choix Multiples)
- Succession de questions ouvertes avec réponse à formuler directement à l'ordinateur
- Succession de questions ouvertes avec réponse à formuler sur copie
- Présentation de ressources multimédias qui peuvent éventuellement servir de base à une réponse sur feuille de copie.

Toutes ces activités peuvent être appuyées par une ressource multimédia de différents types : Image, photo, vidéo, son, texte, page Web, diaporama PowerPoint.

Ordidac propose également de générer des ressources multimédias, basées sur des images, réactives au passage et au clic de la souris. Ces ressources peuvent être utilisées pour illustrer les activités précédemment citées mais elles permettent également de poser à l'apprenant des questions pour lesquelles la réponse consiste en un clic correctement positionné sur l'image.

Dans les QCM, les réponses proposées peuvent être du texte, des images ou des sons.

De très nombreuses options sont disponibles ; exercice :

- noté ou non
- en temps limité ou non (pour l'exercice ou chaque question)
- retour en arrière autorisé ou non
- Aide disponible ou non suivant l'utilisation d'un même questionnaire (situation d'apprentissage ou d'évaluation)
- Correction accessible ou non pour l'élève.

Les résultats des élèves sont sauvegardés et visualisables par l'enseignant.

Ordidac fonctionne en réseau avec, entre autre, possibilité à l'enseignant de visualiser à partir de son poste les résultats des élèves.

Ordidac propose également de faire de la formation à distance.

D'autres options sont également disponibles.

Ordidac est gratuit, l'auteur n'est aucunement responsable de l'utilisation et des conséquences de l'utilisation qu'il sera faite d'Ordidac.[27]

2. Pepit : des exercices éducatifs pour tous



Figure 2-5 : Logiciel Pepit

3. Edu'Key : une clé USB pour l'école primaire



Figure 2-6 : Logiciel Edu'Key

4. Forum Retz / Sciences humaines : le numérique va-t-il bouleverser les pratiques pédagogiques ?



Figure 2-7 : Logiciel Forum Retz

5. Cahiers pédagogiques N°482 : le Web 2.0 et l'école



Figure 2-8 : Logiciel Cahiers pédagogiques

9. CATEGORIES D'ENSEIGNEMENT ASSISTE PAR ORDINATEUR

➤ Taxonomie traditionnelle

- **Exercice et pratique** (Math Munchers, Rabbit Reader, Spellavator Plus, Lecteur accéléré)

Les logiciels de forage et de pratique sont généralement utilisés de la même façon que les feuilles de travail ou les cartes éclair sont utilisées dans les salles de classe. Il offre une exposition répétée à des faits ou des informations, souvent sous forme de question ou de type de jeu. Le logiciel de forage et de pratique s'occupe principalement des habiletés de réflexion de bas ordre. Les étudiants peuvent progresser à leur propre rythme tout en utilisant le logiciel, un autre avantage.

- **Tutorial** (MavisBeacon enseigne la dactylographie, le type à apprendre, Reader Rabbit's apprendre à lire)

Le logiciel didactique présente des concepts ou des compétences et donne ensuite aux élèves la possibilité de les pratiquer. Les logiciels de forage et de pratique ne comprennent pas de composante pédagogique.

- Les étudiants qui sont absents ou qui ont besoin d'une réparation peuvent souvent bénéficier d'un tutoriel informatisé, à condition que l'un soit disponible qui couvre ce qui a été manqué.

- **Résolution de problèmes** (Carmen San Diego, l'usine, National Inspirer, Thinkin' Things, Shape Up)

- Le logiciel de résolution de problèmes permet aux apprenants de voir les résultats de leurs réactions à divers événements. Les apprenants manipulent les variables, et la rétroaction est fournie sur la base de ces manipulations.

- **Simulation** (Stand Hot Dog, Oregon Trail, Virtual Labs-Electricité)
- Une simulation est une représentation ou un modèle d'un événement réel, d'un objet ou d'un phénomène où les apprenants peuvent voir les résultats de leurs actions. Parfois, il n'est pas pratique ou faisable de faire la vraie chose, donc une simulation est utilisée pour fournir des expériences qui autrement seraient niées. La différence entre le logiciel de simulation et le logiciel de résolution de problèmes est que le logiciel de simulation s'occupe de situations réalistes.
- **Tool Software** (Inspiration, MapMachine de quartier, Print Shop, L'Encyclopédie de Grolier, Timeliner)
- Le logiciel d'outil est l'application informatique la plus répandue actuellement utilisée en éducation. Il englobe tous les logiciels qui peuvent être utilisés comme un outil pour l'apprentissage des élèves. Le logiciel lui-même n'est pas le foyer, mais plutôt l'étudiant l'utilise pour les aider à exprimer leurs pensées et montrer leur compréhension.

Programmation informatique (Logo, BASIC, Pascal)

- Il s'agissait d'une grande catégorie d'utilisation d'ordinateurs dans les écoles, mais l'accent mis sur les compétences en programmation a considérablement diminué. Beaucoup d'écoles n'offrent même plus de cours de programmation, mais plutôt d'enseigner aux élèves à utiliser des outils multimédias et d'autres outils informatiques.
- Le logo est toujours enseigné, surtout pour sa valeur en aidant les élèves à comprendre les mathématiques. C'est un langage de programmation qui n'est pas utilisé en dehors de l'éducation.[36]

➤ **Taxonomie de Maddux**

Type I: Utiliser des ordinateurs pour enseigner de façon traditionnelle

Type II: Utiliser des ordinateurs pour développer les méthodes d'enseignement et le contenu des programmes

Si nous n'utilisons que des ordinateurs comme des applications de type I, nous ne capitalisons pas pleinement sur ce que les ordinateurs peuvent ajouter à l'enseignement et à l'apprentissage.[36]

10. TECHNIQUE D'APPRENTISSAGE

- Apporter un ensemble de connaissances à l'élève.
- Établir avec lui un dialogue pédagogique



Par des moyens informatiques

Développer des programmes (didacticiels)



Permettront

- De présenter des informations à l'élève ou de fournir à celui-ci l'accès à de bases de connaissances ou de documentation.
- De le solliciter et de le faire réagir.
- De contrôler ses réactions.
- De choisir le parcours le mieux adapté



Puissance de ces fonctions dépend

- Des caractéristiques du poste de travail de l'élève.
- Des logiciels choisis pour réaliser les didacticiels.[34]

11. LES AVANTAGES DE L'E.A.O

L'enseignement assisté par ordinateur présente les avantages suivants :

- il permet une formation individuelle et une rétroaction immédiate;
- il est excellent pour les exercices et la mise en pratique;
- il peut conserver des statistiques sur la performance des apprenants;
- il fournit une formation standardisée de manière flexible;
- il favorise le travail dans une atmosphère intime et impartiale;
- il peut améliorer la vitesse d'apprentissage et le niveau d'assimilation de manière significative dans plusieurs situations. [25]

12. LES INCONVENIENTS DE L'E.A.O

Il faut, par contre, composer avec les désavantages suivants :

- l'investissement initial peut être considérable;
- il ne convient pas lorsqu'il faut développer des habiletés interpersonnelles;

- les apprenants doivent être familiarisés avec l'utilisation de l'ordinateur;
- il n'est pas approprié lorsque le contenu de l'enseignement change fréquemment;
- il n'est pas économique s'il doit être utilisé pour un seul groupe ou pour un petit nombre d'apprenants. [19]

13.CONCLUSION

On conclut que l'enseignement assisté par ordinateur est un ensemble de méthodes pédagogiques interactives qui prévoient l'usage d'un ordinateur pour dispenser un cours, suivre l'apprentissage...

Dans ce chapitre on a mis, l'enseignement assisté par ordinateur et leur concept de base. Par suite dans le chapitre suivant on va présenter la conception de notre projet.

PARTIE – II : NOTRE PROPOSITION

CHAPITRE 3 : CONCEPTION

1. INTRODUCTION

Après l'étude que nous avons faite dans les deux chapitres précédents sur les Algorithmes et l'Apprentissage assisté par ordinateur nous avons pris une idée sur l'apprentissage de conversion d'un algorithme vers un code java.

Dans ce chapitre nous aborderons une description générale de notre outil ensuite nous mettons en évidence le coté conceptuel de notre application qui constitue une étape fondamentale qui précède l'implémentation, en détaillant les différents diagrammes et scénarios à implémenter dans la phase suivante. Ceci permettra de mieux comprendre notre outil.

Nous avons utilisé une démarche générale basée sur le langage UML.


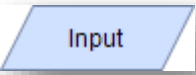
2. L'OBJECTIF DE NOTRE SYSTEME

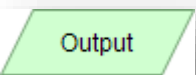
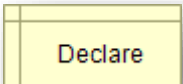
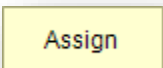
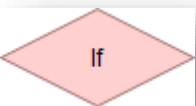
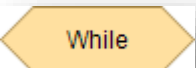
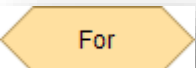
L'objectif principal de notre travail pédagogique est de réaliser un outil qui aide des apprenants débutants en langage Java pour convertir des algorithmes (organigrammes) vers un code Java et apprendre les notions de base de ce langage.

3. MECANISME DE CONVERSION

Nous avons choisi l'organigramme pour faire la conversion de ce dernier vers un code source Java.

Nous affectons chaque forme d'organigramme par un morceau de code Java, la table ci-dessous explique ce principe :

Nom	Forme	Code Java
Comment		// Tapez ici votre commentaire
Input		Scanner x = new Scanner(System.in); x.nextInt();

Output		<code>System.out.Println(x);</code>
Declare		<code>int x;</code> <code>real x ;</code> <code>string x ;</code> <code>boolean x ;</code> <code>int[][] x = new int[2][4];</code>
Assign		<code>x = y;</code>
If		<code>if (x >= y){</code> <code> }</code> <code>else {</code> <code> }</code>
While		<code>while (x >= y){</code> <code> }</code>
For		<code>for (int x = y;x <= y;x++){</code> <code> }</code>

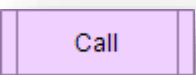
Call		<pre>public static int addition(x:int,y:int){ z = x+y; return z }</pre>
------	---	---

Tableau 3-1 : Mécanisme de conversion

4. ARCHITECTURE GLOBAL DU SYSTEME

L'architecture dessinée dans la figure suivante présente une vue approfondie sur notre système. Elle illustre les échanges et les interactions faites pendant son utilisation.

Les deux acteurs impliqués de notre modèle sont :

- **L'enseignant** : c'est le responsable de la gestion générale de notre application et il s'occupe de la création des différents cours et exercices qui aident les apprenants d'apprendre la conversion d'un organigramme vers un code source Java.

- **Apprenant** : tout simplement c'est la personne qui lit les cours et répond aux différents exercices proposés par l'enseignant.

Notre architecture est composée en deux parties, une première conçue spécialement pour l'enseignant et la deuxième pour les apprenants.

- 1. Partie 1 : interface d'enseignant** : Dans cette partie se compose essentiellement en deux interfaces :
 - Pédagogie : permet la gestion des exercices et des cours de notre application.
 - manipulation : permet la gestion des niveaux et sessions d'apprenants.
- 2. Partie 2 : interface d'apprenant** : permet à l'apprenant de suivre les cours de son niveau puis entrainer pour mieux comprendre ensuite résoudre les exercices pour passer aux niveaux supérieurs.

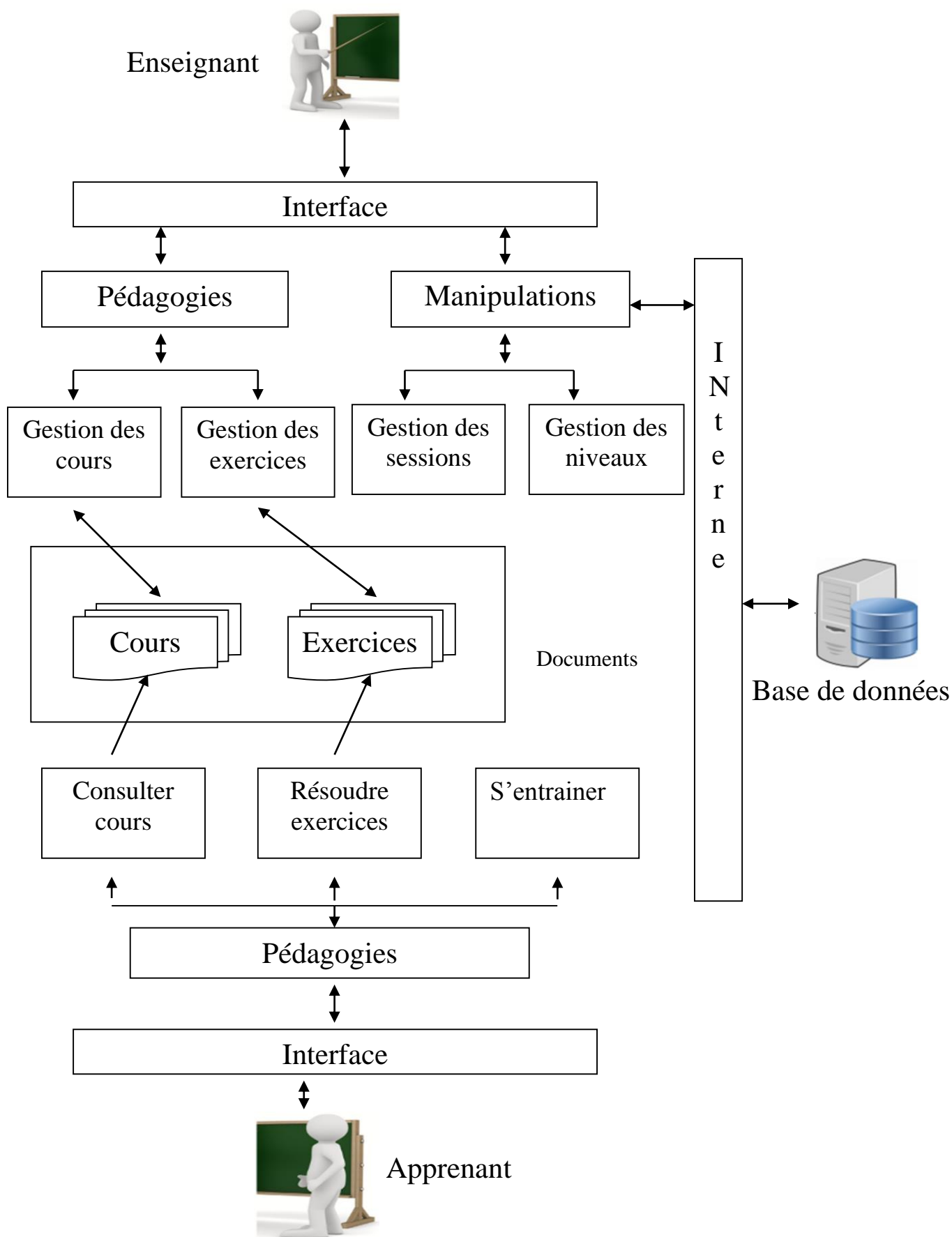


Figure 3-1 : Architecture globale du système

5. PROCESSUS D'APPRENTISSAGE

Parmi les approches d'apprentissages existantes que nous les avons citées dans le chapitre précédent nous allons choisir l'Approche behavioriste (Le savoir est découvert par l'élève) car il est adaptable pour l'apprentissage assisté par ordinateur et aussi offre quelques avantages tels que :

- la décomposition de savoir.
- L'enseignant est attentif aux possibilités et à l'évolution individuelle de l'apprenant ;
- L'apprenant peut progresser à son rythme.

Le processus d'apprentissage est découpé en 4 phases principales comme suit :

- 1 Consulter le cours** : l'objectif est de faire découvrir la conversion d'un organigramme vers un code source Java, et apprendre à chaque étape une explication sur la programmation,
- 2 S'entraîner** : L'entraînement IOI est conçu pour permettre de progresser rapidement en programmation, par un apprentissage allant de la découverte des bases de la programmation jusqu'au niveau final.
- 3 Résoudre Exercices** : De nombreux exercices de programmation existent préparés par l'enseignant dans lesquels l'apprenant a une occasion de mettre en pratique ses connaissances.
- 4 Evaluation** : Nous vérifions automatiquement et immédiatement si les programmes codés par les apprenants sont corrects. Ces derniers peuvent donc s'assurer qu'ils ont bien résolu leurs exercices avant d'en lire la correction, puis de passer à l'exercice suivant ou passer au niveau suivant.

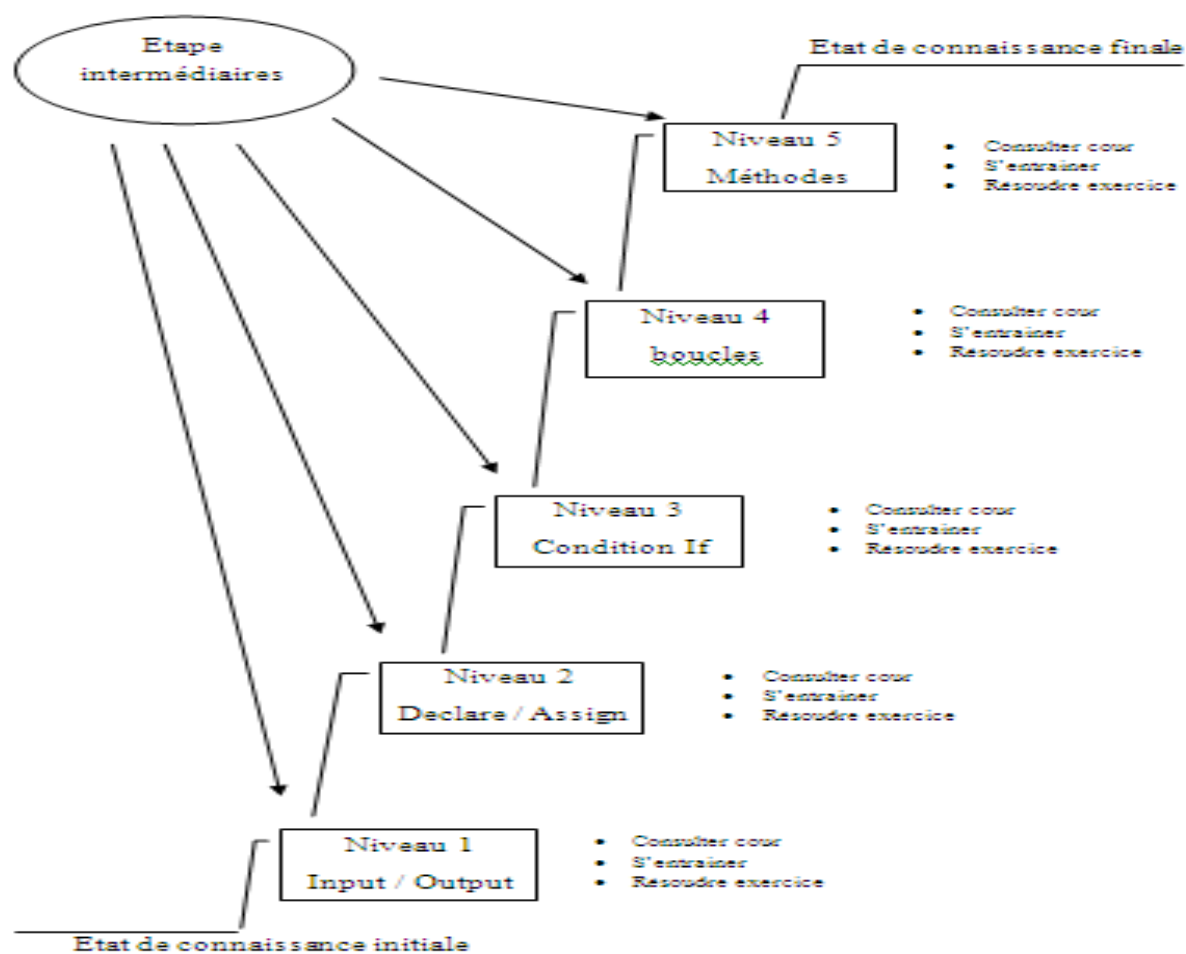


Figure 3-2 : Processus d'apprentissage

6. CONCEPTION DETAILLEE

6.1.Choix de la méthode

Le langage de modélisation unifié, de l'anglais Unified Modeling Language (UML), est un langage de modélisation graphique à base de pictogrammes conçu pour fournir une méthode normalisée pour visualiser la conception d'un système. Il est couramment utilisé en développement logiciel et en conception orientée objet.

Dans cette démarche, les diagrammes UML utilisés sont les suivant : Diagramme de cas d'utilisation, diagrammes de séquence et le diagramme de classes.

6.2.Diagrammes de cas d'utilisation

Le diagramme de cas d'utilisation nous présente les principales fonctions du système, ainsi que les acteurs qui interviennent.

Voici le diagramme de cas d'utilisation qui regroupe les fonctionnalités de notre système :

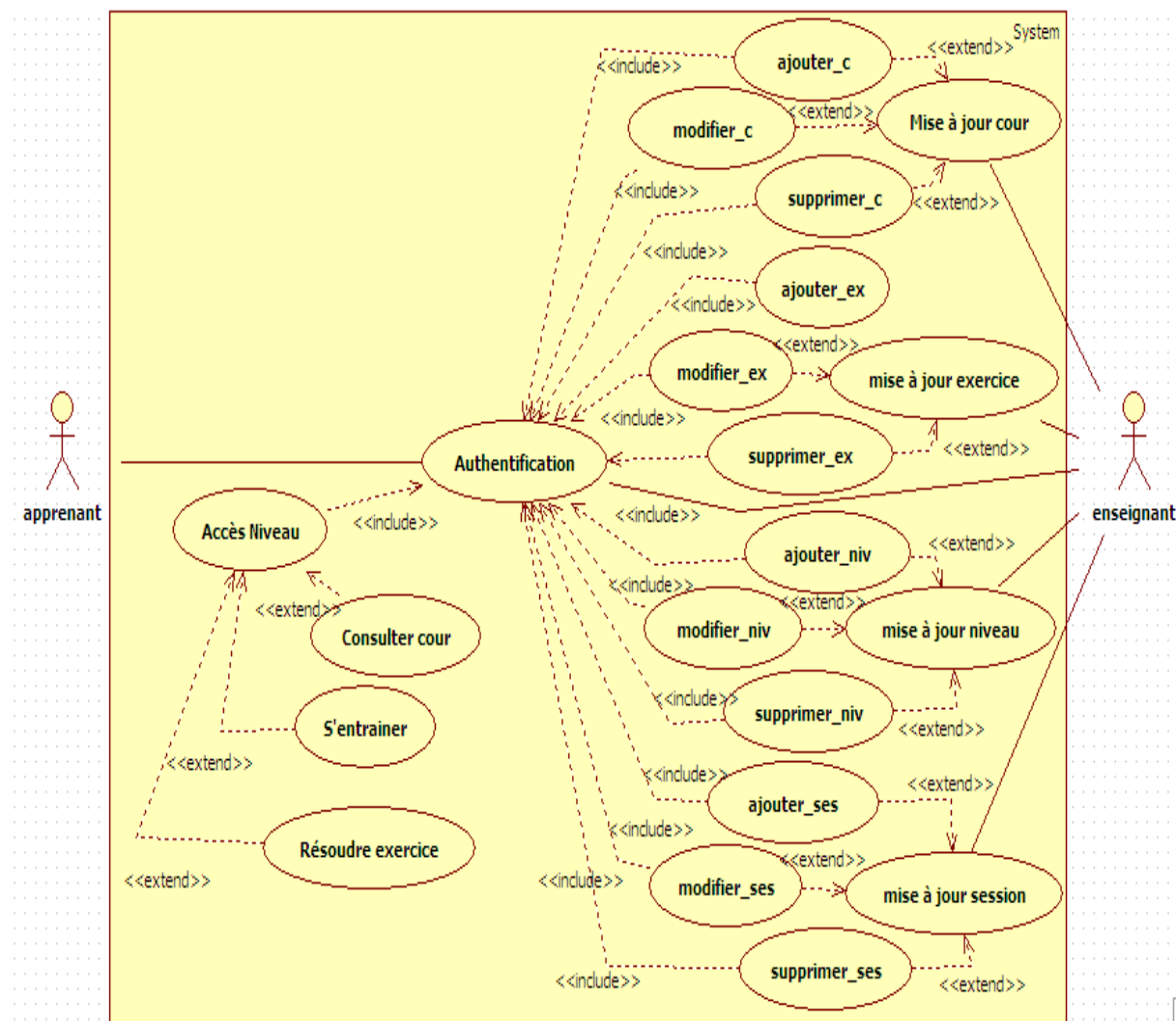


Figure 3-3 : Diagramme de cas d'utilisation des deux acteurs

Dans notre système il y a deux acteurs principaux qui sont :

➤ Acteur : enseignant

Il gère les fonctionnalités de l'application d'où il permet de faire :

- S'inscrire des apprenants : ouvrir un compte et créer un pseudo et mot passe.
- Gestion des cours : action d'ajout ou supprimer un cours.
- Gestion des exercices : action d'ajout ou supprimer un exercice.
- Gestion des apprenants : action de consulter les niveaux des apprenants.

➤ Acteur : apprenant

L'Apprenant utilise l'application d'où il peut:

- Accéder à un niveau.
- Consulter un cours ou résout un exercice.

6.3. Diagramme de séquence

Les diagrammes de séquences sont la représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique dans la formulation UML.

Dans ce qui suit, nous présentons le diagramme et nous décrivons les scénarios de chaque cas d'utilisation de séquence pour chaque cas d'utilisation dans notre système.

➤ Enseignant

Scénario : Après l'accès à l'application

Scénario : s'inscrire des apprenants.

Scénario : Gestion des sessions (apprenants).

Scénario : Gestion des cours.

Scénario : Gestion des exercices.

➤ Apprenant

- Après l'accès à l'application

Scénario : Accéder à un niveau.

Scénario : S'entraîner.

Scénario : Consulter les cours ou résoudre les exercices.

6.4. Diagramme de classe

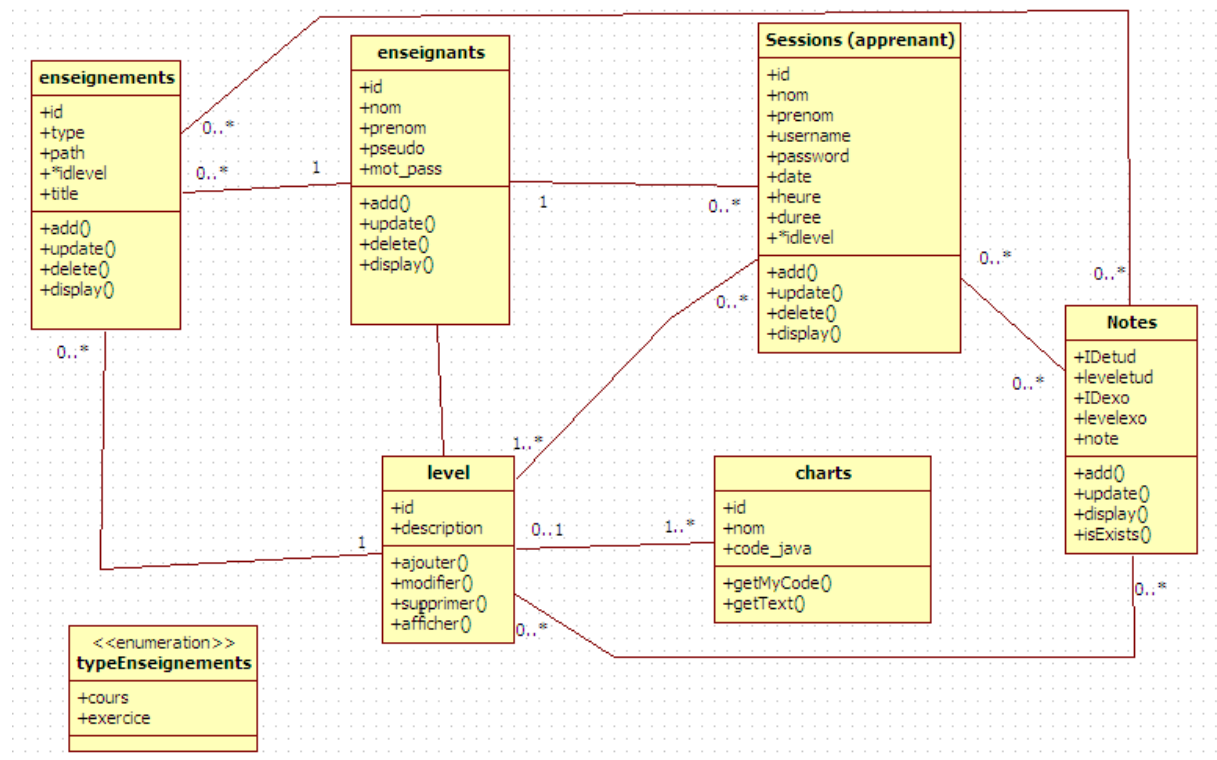


Figure 3-4 : Diagramme de classes

Un diagramme de classes dans le langage de modélisation unifié (UML) est un type de diagramme de structure statique qui décrit la structure d'un système en montrant le système de classes, leurs attributs, les opérations (ou) les méthodes et les relations entre les classes.

7. CONCLUSION

Comme nous pouvons le constater, l'activité de la conception a facilité la compréhension de notre système, qui ébauche vers l'activité d'implémentation.

CHAPITRE 4 : IMPLEMENTATION

1. INTRODUCTION

Ce chapitre vise à montrer les outils utilisés pour mettre en place notre application suivant la conception décrite dans le chapitre précédent.

Les différents modules d'analyses sont développés dans Java. Nous avons complété cet environnement par les outils de : JDOM (version 2.0.6) pour la Manipulation de documents XML et Jgraphx pour dessiner les styles d'organigramme.

Nous expliquons d'abord ces outils ensuite nous présentons les différents composants de l'application.

2. OUTILS ET LANGAGE DE DEVELOPPEMENT

2.1.Langage de programmation

Nous avons utilisé le langage JAVA, Ce choix est justifié par ses nombreux avantages dont voici quelques-unes :

- Il est orienté objet : permet l'encapsulation, le polymorphisme, et l'héritage, et qui vont nous aider à bien organiser et structurer l'application.
- la portabilité : car le compilateur java produit un code intermédiaire qui sera interprété par une JVM.
- la robustesse : bonne gestion de mémoire (nettoyage par garbagecollector, pas d'accès direct à la mémoire), et une bonne gestion d'exception.
- Il est distribué : API réseaux (socket), Applet, Servlet, RMI,...etc.
- Il dispose d'une bibliothèque de classes très riches.

Nous avons utilisé la dernière version du JDK (jdk-8u5-nb-8-windows-x64 actuellement) car elle contient plus de librairies (des packages) que de ses prédécesseurs et Eclipse néon 3[41]

2.1.1. Eclipse néon 3

Eclipse est un environnement de développement intégré (IDE) utilisé dans la programmation informatique, et il est le plus largement utilisé Java IDE. Il contient un espace de travail de base et un système de plug-in extensible pour personnaliser l'environnement.

En plus de Java, éclipse permet également de supporter différents autres langages, comme

Python, C, C++, JavaScript, XML, Ruby, PHP et HTML. Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projets multi-langage, efactoring, éditeur graphique d'interfaces et de pages Web). Conçu en Java, eclipse est disponible sous Windows, Linux ou sous une version indépendante des systèmes d'exploitation (requérant une machine virtuelle Java).

2.1.2. XAMPP

est un ensemble de logiciels permettant de mettre en place facilement un serveur Web local, un serveur FTP et un serveur de messagerie électronique. Il s'agit d'une distribution de logiciels libres (**X** (cross) **A**pache **M**ariaDB **P**erl **P**HP) offrant une bonne souplesse d'utilisation, réputée pour son installation simple et rapide. Ainsi, il est à la portée d'un grand nombre de personnes puisqu'il ne requiert pas de connaissances particulières et fonctionne, de plus, sur les systèmes d'exploitation les plus répandus.[37]

2.2. Les outils

2.2.1. JDOM

JDOM est une API du langage Java, permet de manipuler des données XML plus simplement qu'avec les API classiques. Son utilisation est pratique pour tout développeur Java.

- **Origines :**
 - **SAX :** SAX est l'acronyme de Simple API for XML (section précédent), Ce type de parseur utilise des événements pour piloter le traitement d'un fichier XML.
 - JDOM utilise des collections SAX pour parser les fichiers XML.[39]
- **Description de DOM :** Acronyme de Document Object Model, Rôle d'après le W3C :
 - fournir une représentation mémoire d'un document XML sous la forme d'un arbre d'objets et d'en permettre la manipulation (parcours, recherche et mise à jour)
 - DOM est défini pour être indépendant du langage dans lequel il sera implémenté (PHP, Java, etc.)
 - DOM n'est qu'une spécification qui, pour être utilisée, doit être implémentée par un éditeur tiers. DOM n'est donc pas spécifique à Java
 - Le parseur DOM pour JAVA le plus répandu est Xerces
- **Description de JDOM :**
 - JDOM utilise DOM pour manipuler les éléments d'un Document Object Model spécifique (créé grâce à un constructeur basé sur SAX)

- JDOM permet donc de construire des documents, de naviguer dans leur structure, d'ajouter, de modifier, ou de supprimer leur contenu
- Mais qu'est-ce que JDOM nous apporte de plus ?

- La simplicité !

Il est en vérité très laborieux de développer des applications complexes autour de XML avec DOM, qui rappelons le, n'a pas été développé spécifiquement pour Java

2.2.2. JGraphX - Bibliothèque Java Swing

JGraphX est une bibliothèque Java Swing schématisation (visualisation graphique) sous licence sous la licence BSD. Bien que, les noms de paquets utilisent celui de « mxGraph », cette bibliothèque n'est pas appelée mxGraph. mxGraph est la bibliothèque JavaScript schématisant - <https://github.com/jgraph/mxgraph>.

- **Quelles sont les applications peuvent JGraphX être utilisé?**

Exemples d'applications pour une bibliothèque de visualisation graphique comprennent: des diagrammes de processus, flux de travail et la visualisation de BPM, des diagrammes, le trafic ou le débit d'eau, base de données et la visualisation Web, les réseaux et les écrans de télécommunications, les applications de cartographie et SIG, des diagrammes UML, circuits électroniques, VLSI, CAD, financière et les réseaux sociaux, data mining, biochimie, cycles écologiques, des entités et des relations de cause à effet et les organigrammes.

- **Design**

La conception originale pour JGraph était d'en faire une extension architecturale de JTree pour le langage Java.[38]

3. DESCRIPTION DU SYSTEME NOTRE SYSTEME

Il est composé de deux parties : administrative pour l'enseignant et pédagogique pour l'étudiant, la première partie est réservée seulement aux enseignants administrateurs de note outil d'apprentissage, et la deuxième partie est réservée pour tous les apprenants qui ont une session dans l'outil. Dans ce qui suit nous allons présenter quelques fenêtres de l'interface de notre application.

3.1.Interface de connexion

Cette interface permet à l'utilisateur (l'enseignant ou apprenant) qui a déjà un compte dans notre application d'accéder à son interface.

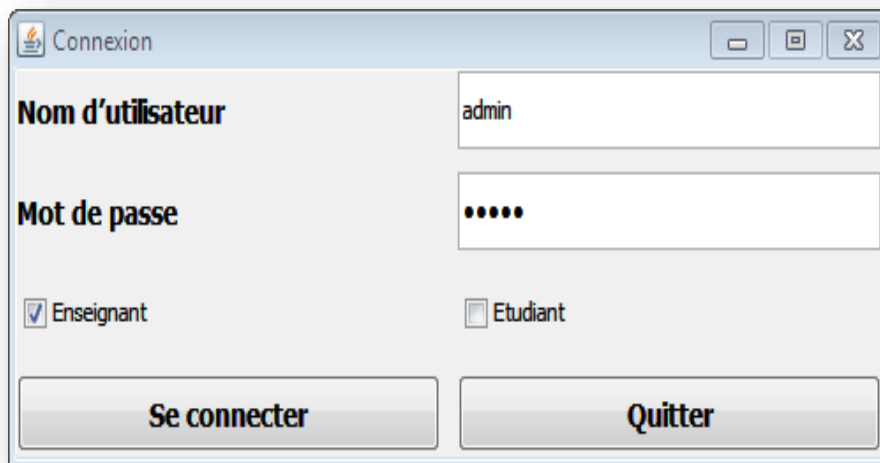


Figure 4-1 : Interface d'accès

3.2. Interface des objets d'organigramme

Nous cliquons sur le lien entre Begin et End, puis nous choisisons 'mode modification' pour afficher l'interface des objets 'add to object'.

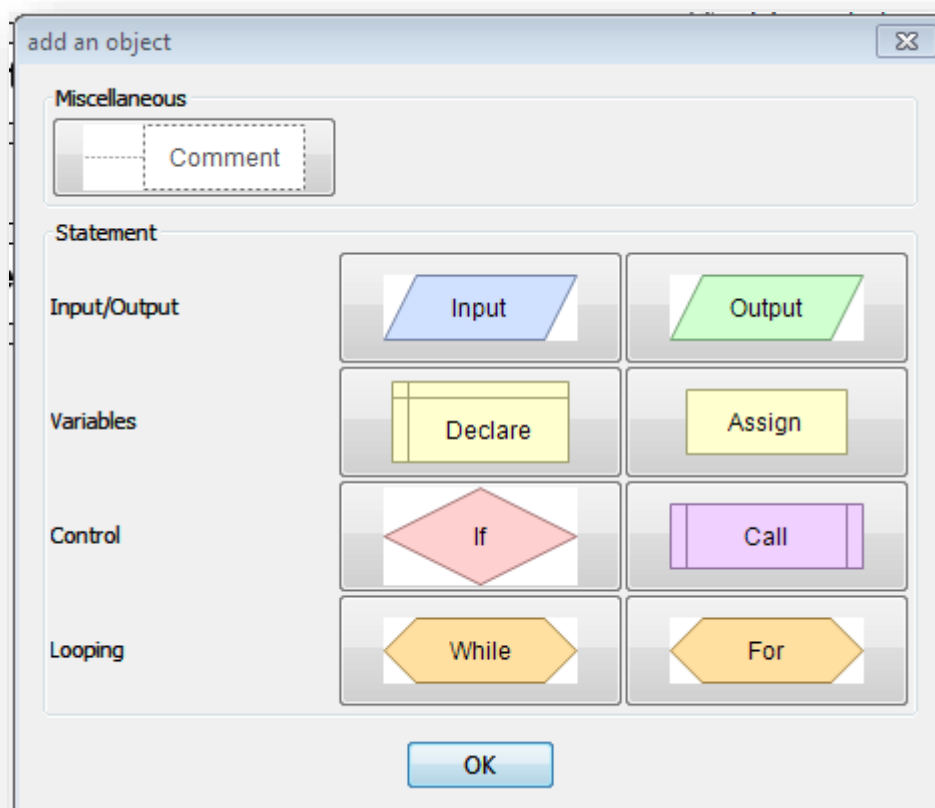


Figure 4-2 : Interface des objets d'organigramme

3.2.1. Comment

Le bouton « comment » permet l’affichage d’une nouvelle fenêtre pour saisir un commentaire.

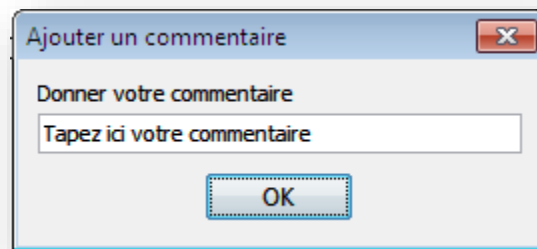


Figure 4-3 : Interface de comment

3.2.2. Input

Le bouton « input » affiche une nouvelle fenêtre pour saisir les entrées, ainsi leurs types.

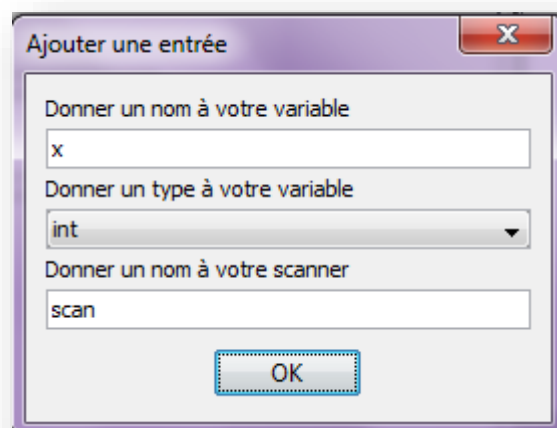


Figure 4-4 : Interface d'Input

3.2.3. Output

Le bouton « output » affiche une nouvelle fenêtre pour saisir les sorties (de type variable ou bien un message).

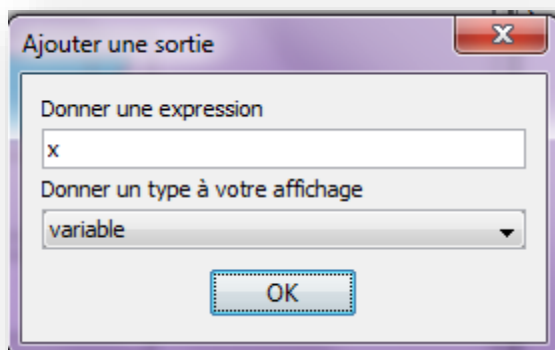


Figure 4-5 : Interface d'output

3.2.4. Declare

Le bouton « Declare » affiche une nouvelle fenêtre pour déclarer des variables ou des tableaux avec ses mentions.

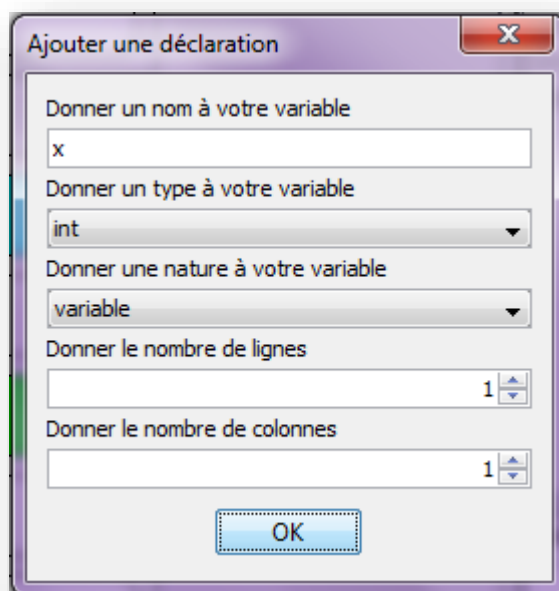


Figure 4-6 : Interface de Déclarer

3.2.5. Assign

Le bouton « Assign » affiche une nouvelle fenêtre pour faire une affectation.

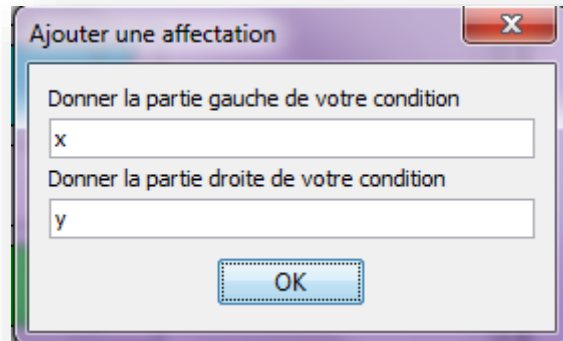


Figure 4-7 : Interface d'assigner

3.2.6. If

Le bouton « if » affiche une nouvelle fenêtre pour ajouter une condition à un bloc ou 2 blocs (else).

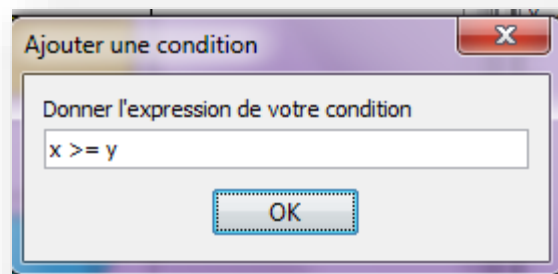


Figure 4-8 : Interface d'If

3.2.7. Call

Le bouton « Call » affiche une nouvelle fenêtre pour ajouter des méthodes prédéfinies.

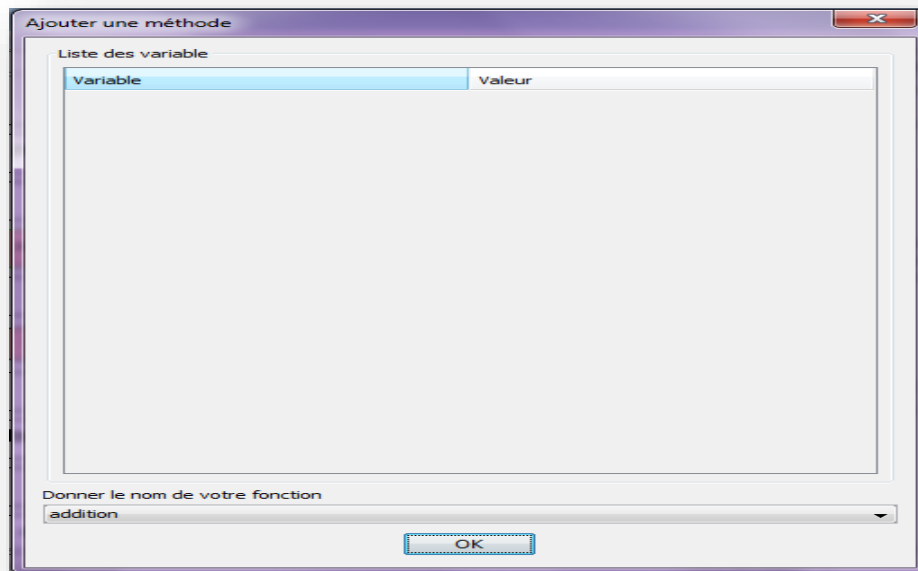


Figure 4-9 : Interface de call

3.2.8. While

Le bouton « While » affiche une nouvelle fenêtre pour ajouter une condition d'arrêt.

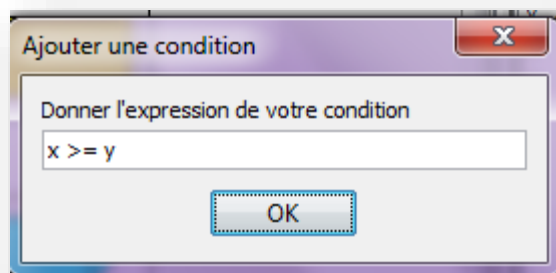


Figure 4-10 : Interface de condition de while

3.2.9. For

Le bouton « For » affiche une nouvelle fenêtre pour saisir le nom de compteur et ses valeurs : initiale et finale.

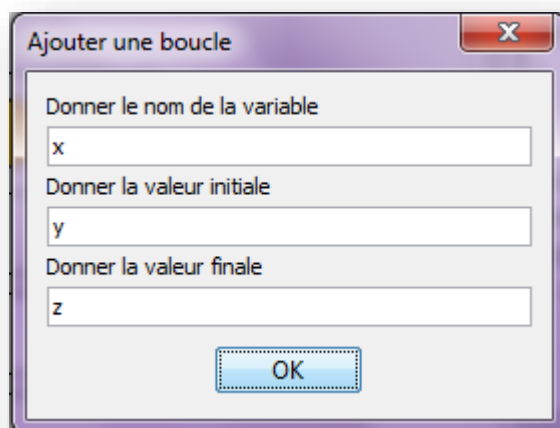


Figure 4-11 : Interface de condition de for

4. PARTIE ENSEIGNANT

L'outil offre à l'enseignant une interface de gestion comme suit :



Figure 4-12 : Interface de partie d'enseignant

Nous avons deux menus : Manipulations et pédagogies dans cette espace :

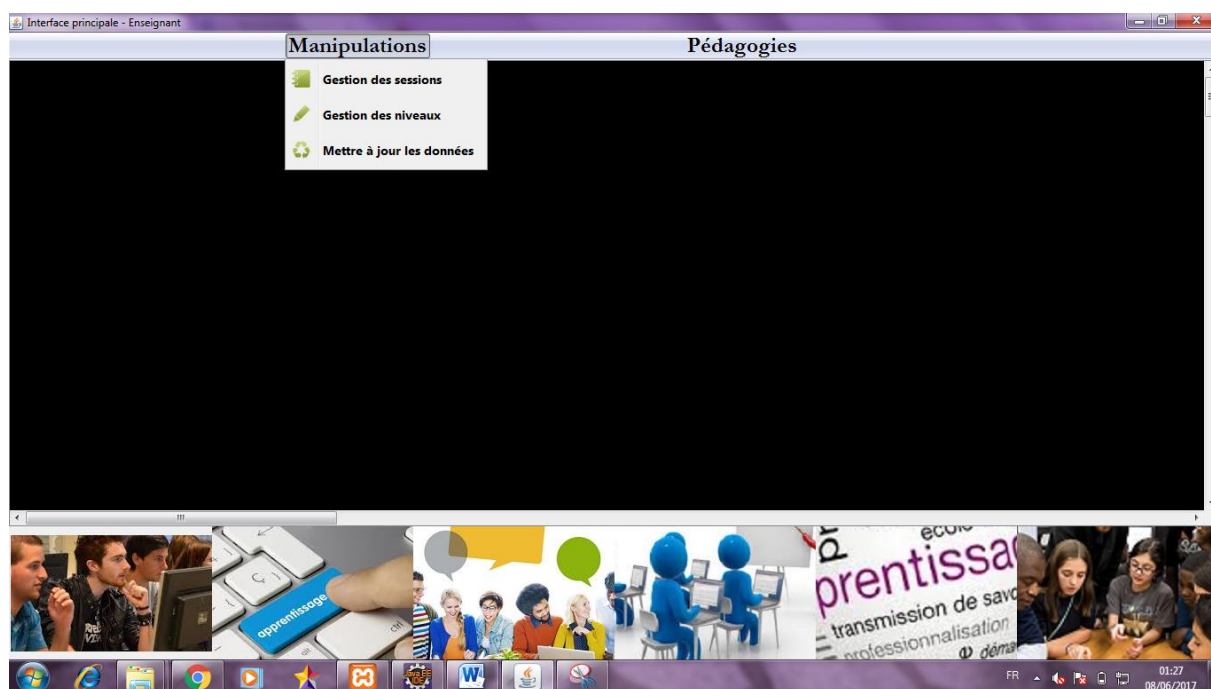


Figure 4-13 : Interface de menu de manipulations

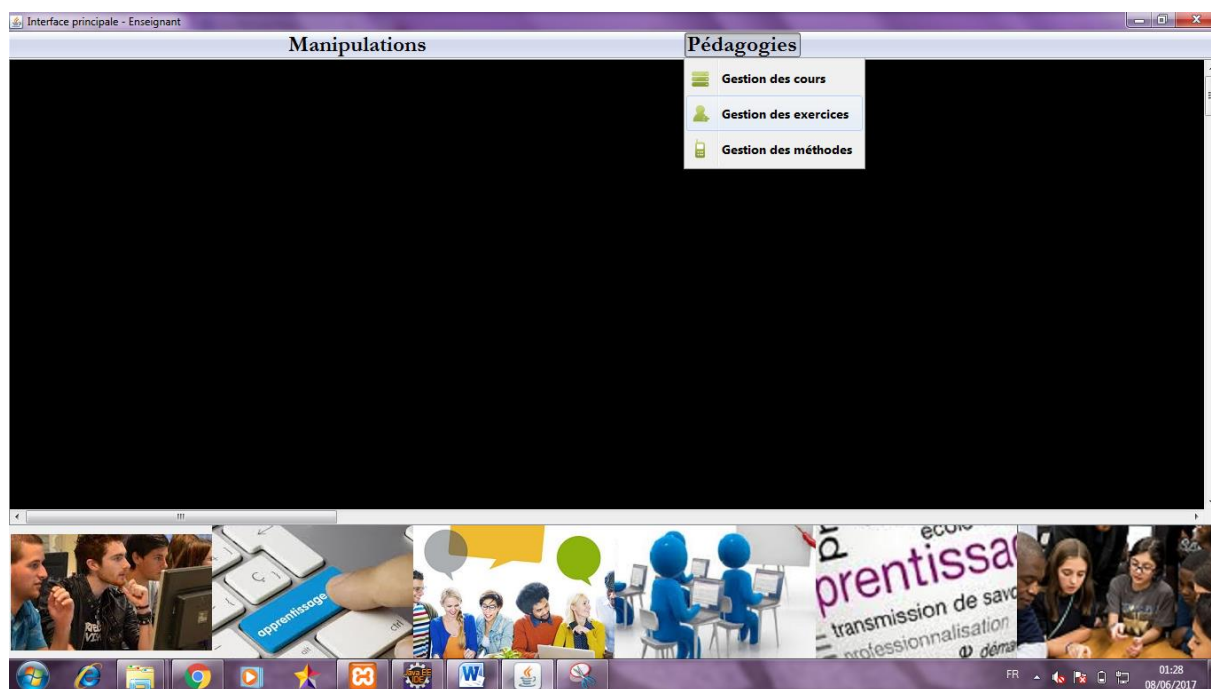


Figure 4-14 : Interface de menu de pédagogies

4.1.Manipulations

4.1.1. Interface de gestion des sessions

L'enseignant gère les sessions des apprenants à partir de la fenêtre de gestion des sessions :

- ❖ Bouton Ajouter : il peut ajouter une nouvelle session pour un nouvel apprenant et remplir le formulaire.
- ❖ Bouton Modifier : après avoir sélectionné une ligne (apprenant existe déjà) dans le tableau, l'enseignant peut modifier ses informations.
- ❖ Bouton Supprimer : après avoir sélectionné une ligne (apprenant existe déjà) dans le tableau, l'enseignant peut supprimer un apprenant.

Le bouton « Valider » permet d'enregistrer les modifications des sessions.

The screenshot shows a window titled "Gestion des sessions". At the top, there is a section "Informations sur les sessions actuelles" containing a table with the following data:

ID	Nom	Prénom	Username	Password	Date	Heure	Durée	Niveau
1	nom	prenom	username	password	2017-05-27	07:20	toujours	niveau 1 (...)

Below the table is a section "Gestions des sessions" with a form for adding or modifying sessions. The form fields are:

- ID :
- Nom :
- Prénom :
- Username :
- Password :
- Date/Heure : ...
- Durée (jours) :
- Niveau :

At the bottom of the form are three buttons: "Ajouter", "Modifier", and "Supprimer". Below these buttons is a "Valider" button.

Figure 4-15 : Interface de gestion des sessions

4.1.2. Interface de gestion des niveaux

L'enseignant gère les niveaux des apprenants à partir de la fenêtre de gestion des niveaux :

- ❖ Bouton Ajouter : il peut ajouter un nouveau niveau en cochant dans la fiche d'ajout les styles de diagrammes qui désignent ce niveau.
- ❖ Bouton Modifier : l'enseignant peut modifier un niveau en lui sélectionnant dans le tableau des niveaux.
- ❖ Bouton Supprimer : l'enseignant peut aussi supprimer un niveau existe déjà en lui sélectionnant dans le tableau.

Le bouton « Valider » permet d'enregistrer les modifications des niveaux.

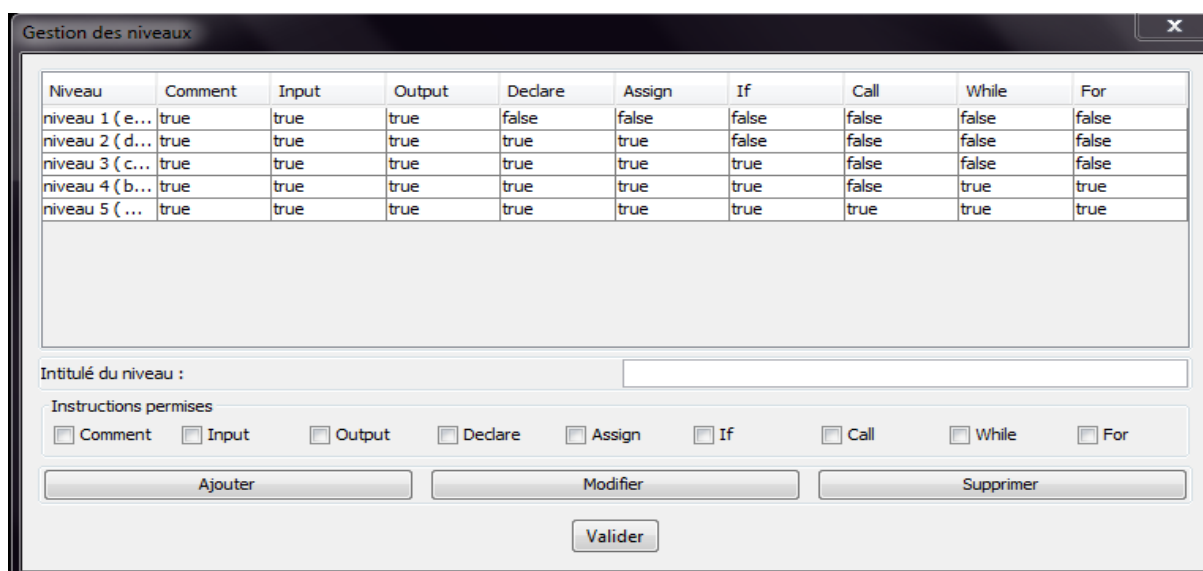


Figure 4-16 : Interface de gestion des niveaux

4.2. Pédagogies

4.2.1. Interface de gestion des cours

Elle se décompose en 3 parties :

- ❖ Dessiner le diagramme.
- ❖ Afficher le code en temps réel et gérer les méthodes.
- ❖ Faire une explication sur ce cours.

Puis donner un titre à ce cours (la case de titre) et affecter leurs niveaux à partir d'un clic sur le bouton « affecter niveau ».

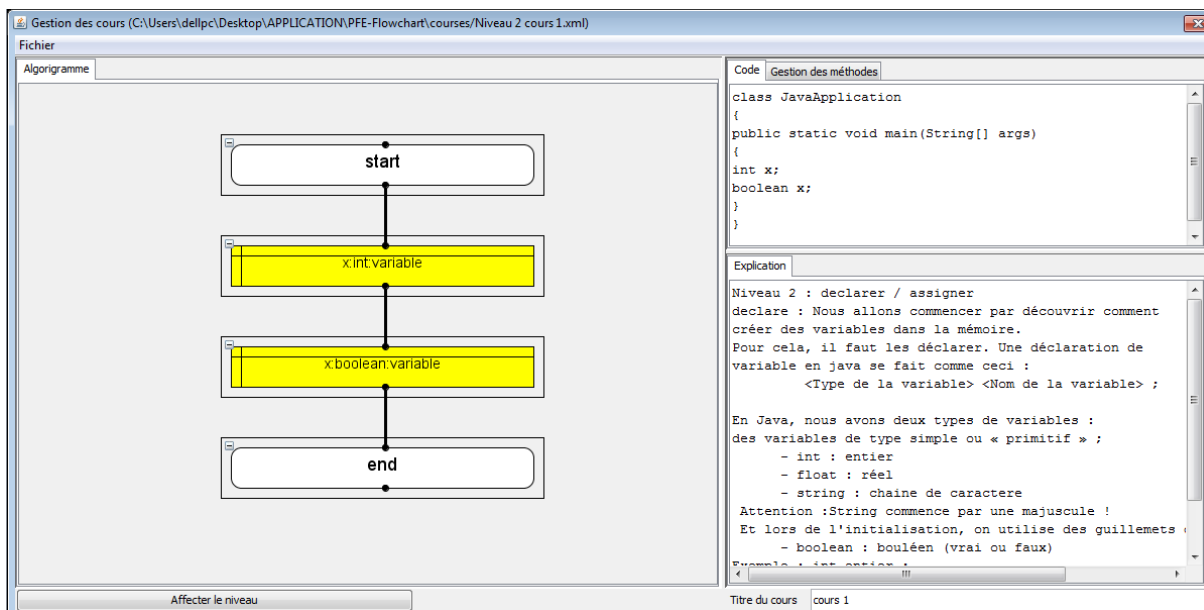


Figure 4-17 : Interface de gestion des cours

Le bouton « enregistrer » dans le menu « Fichier » permet d’enregistrer les cours.

4.2.2. Interface de gestion des exercices

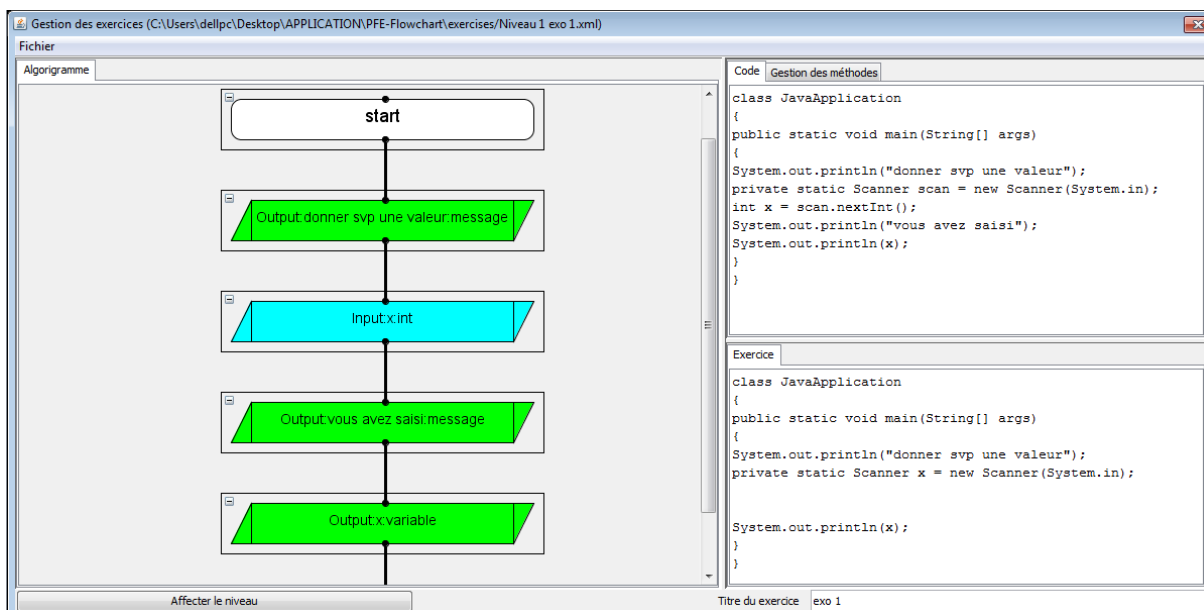


Figure 4-18 : Interface de gestion des exercices

Elle se décompose en 3 parties :

- ❖ Dessiner le diagramme.
- ❖ Afficher le code en temps réel et gérer les méthodes.

- ❖ La partie « exercice » : l'enseignant peut modifier le code de l'exercice pour que l'étudiant reçoit des lignes de code vides et il doit trouver sa solution. Puis, il peut donner un titre à cet exercice (la case de titre) et affecter leur niveau à partir d'un clic sur le bouton « affecter niveau ».

Le bouton « enregistrer » dans le menu « Fichier » permet d'enregistrer l'exercice.

4.2.3. Interface de gestion des méthodes

Elle se décompose en 3 parties :

- ❖ Dessiner l'organigramme de la méthode.
- ❖ Afficher le code en temps réel.
- ❖ La partie « paramètres » : l'enseignant peut définir la Signature de la méthode, son nom et ses paramètres. Puis, il peut donner un titre à ce cours (la case de titre) et affecter leur niveau à partir d'un clic sur le bouton « affecter niveau ».

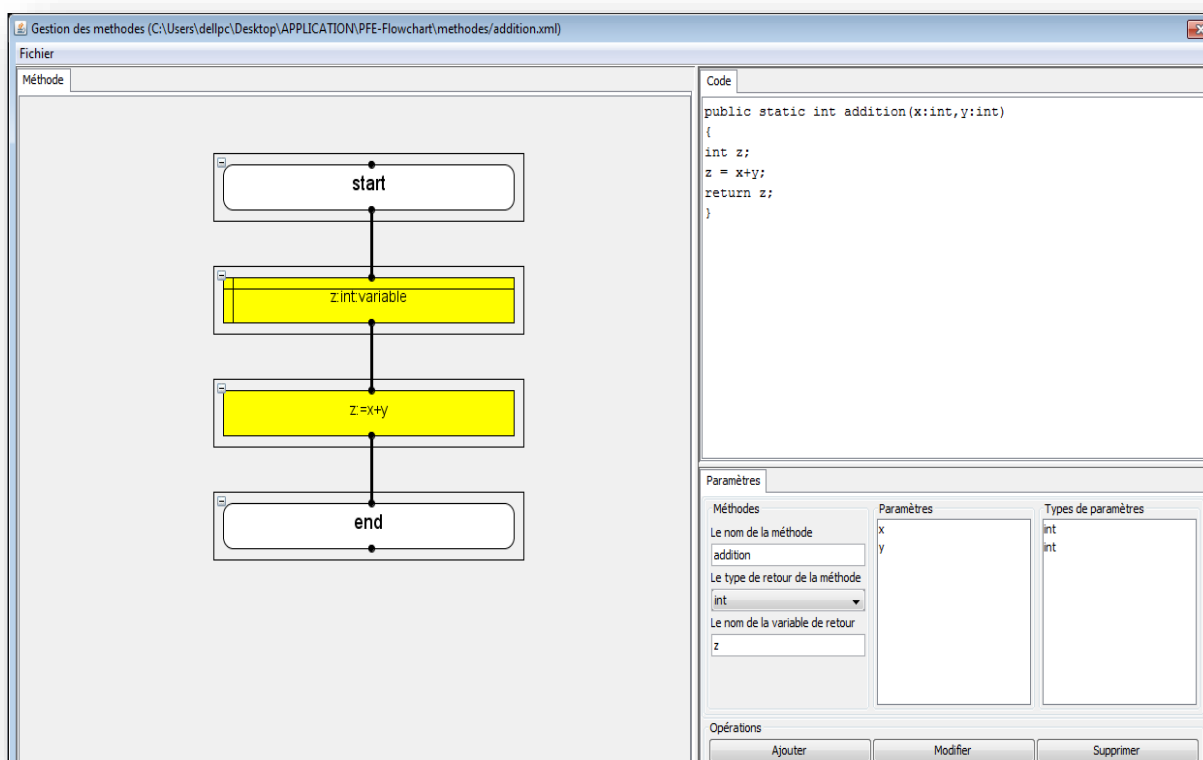


Figure 4-19 : Interface de gestion des méthodes

Le bouton « enregistrer » dans le menu « Fichier » permet d'enregistrer la méthode.

5. PARTIE ETUDIANT

L'étudiant peut ouvrir sa session via notre outil. Nous lui affichant une fenêtre qui contient les boutons qui sont en premier dans l'interface : « S'entraîner », « consulter cours » et « résoudre un exercice » pour manipuler son niveau et le 4ème bouton « choisir un niveau » pour ré-accéder à un niveau précédent.

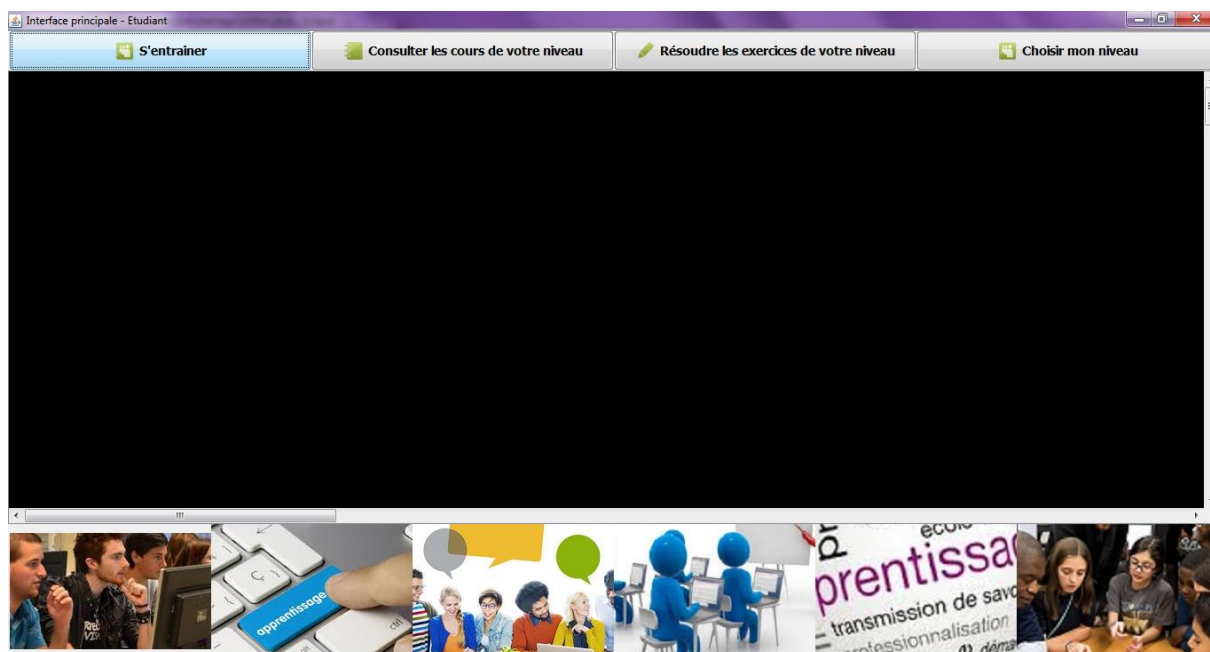


Figure 4-20 : Interface de partie d'étudiant

5.1. Interface de S'entraîner

L'accès à un niveau permet à l'apprenant de faire des entrainements (choisir des organigrammes disponibles et voir le code source en Java de ces derniers en temps réel).

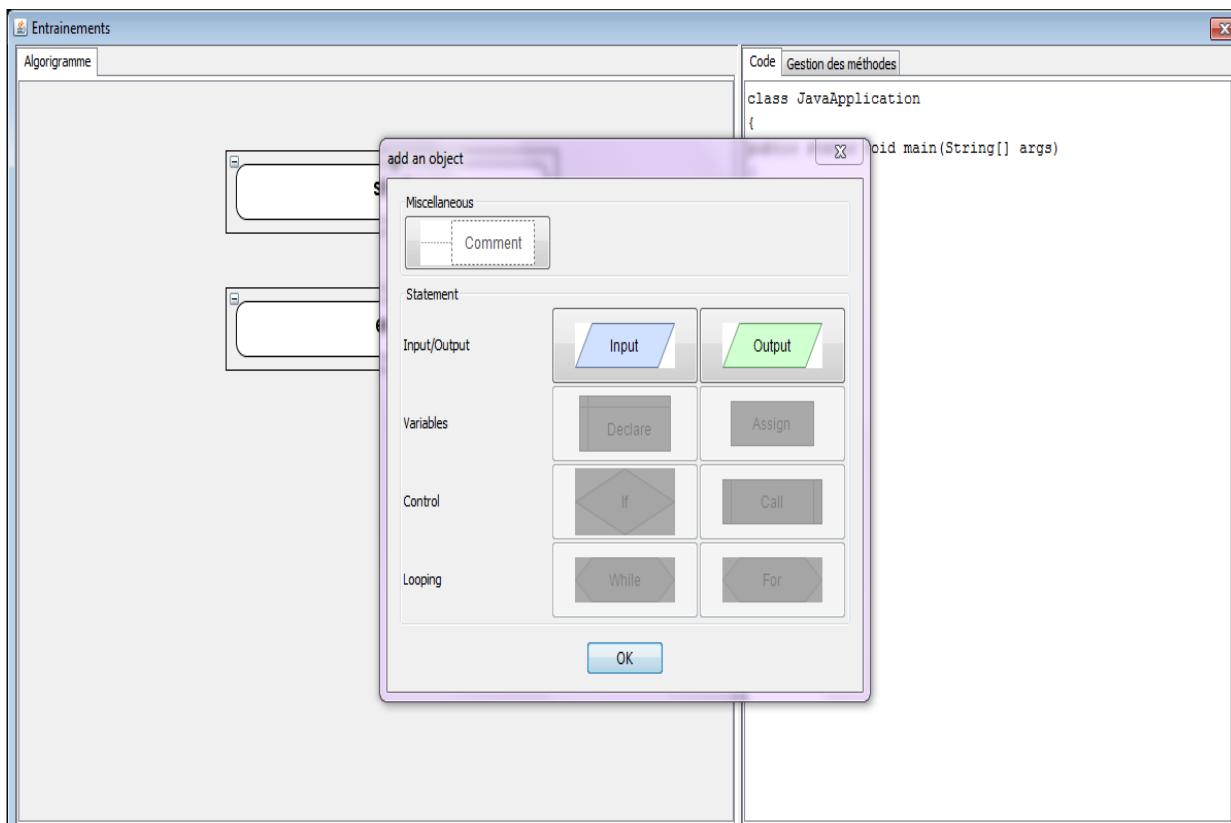


Figure 4-21 : Interface d’entraînement

5.2.Interface de consultation des cours d’un niveau

- **Choix d’un cours**

Pour consulter un cours il faut choisir et cocher un des cours à partir la fenêtre de choix comme suit :

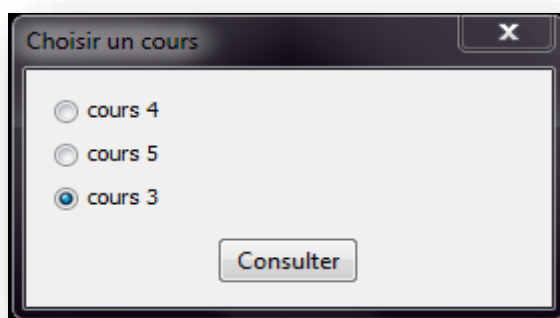


Figure 4-22 : Interface de choix d’un cours.

- **Consultation du cours choisi**

Une interface affiche le cours choisi sous format d’un organigramme avec explication comme suit :

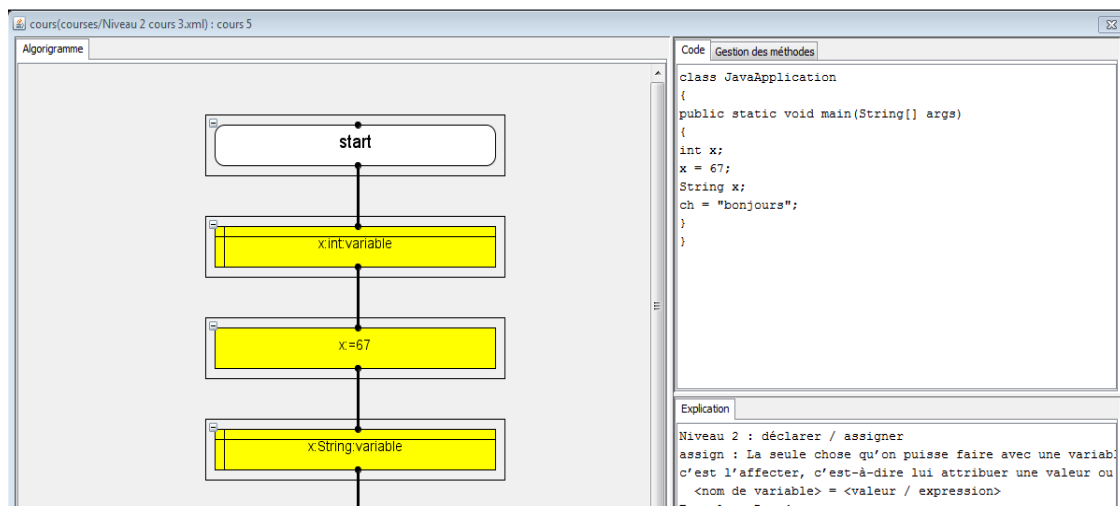


Figure 4-23 : Interface de cours choisi

5.3.Interface de résolution des exercices d'un niveau

- **Choix d'un exercice**

Pour résoudre un exercice il faut choisir et cocher un des exercices à partir la fenêtre de choix d'un exercice comme suit :

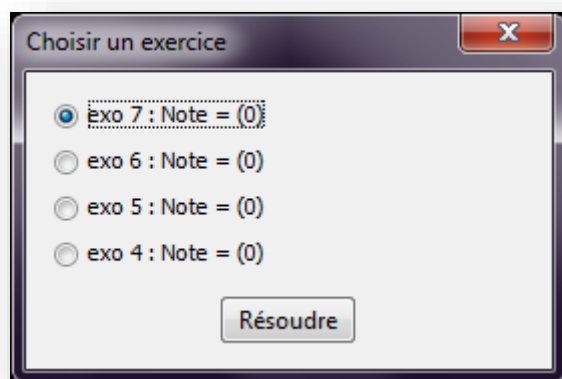


Figure 4-24 : Interface de choix d'un exercice

- **Résolution d'un exercice choisi**

Une interface affiche l'exercice choisi comme un algorithme et deux autres phases : La première pour écrire leur solution et la deuxième pour récupérer le code des méthodes prédéfinies.

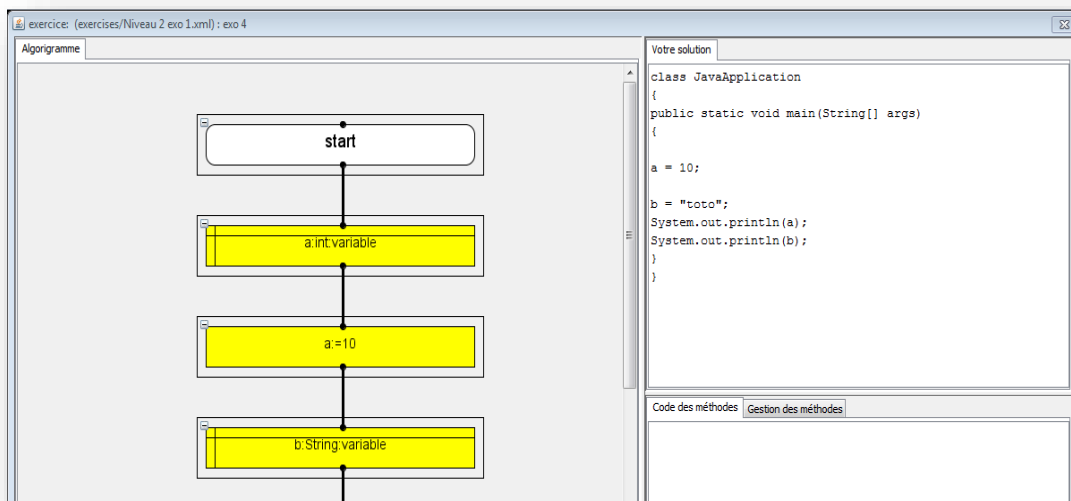


Figure 4-25 : Interface d'un exercice choisi

Le bouton « valider » permet de prendre une note sous forme de pourcentage sur 100%. Les notes permettent aux apprenants de passer à un autre niveau.

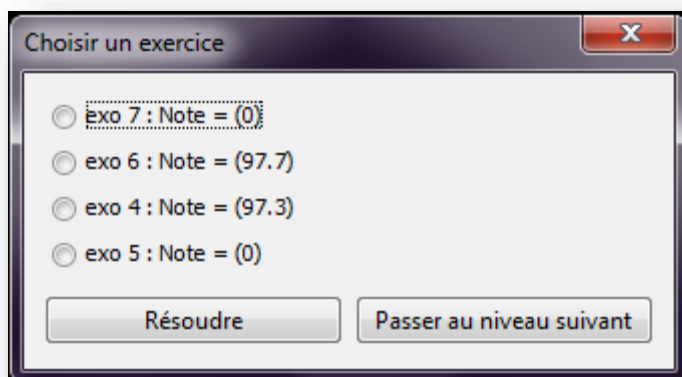


Figure 4-26 : Interface de passage vers le niveau suivant

5.4.Interface de choix d'un niveau

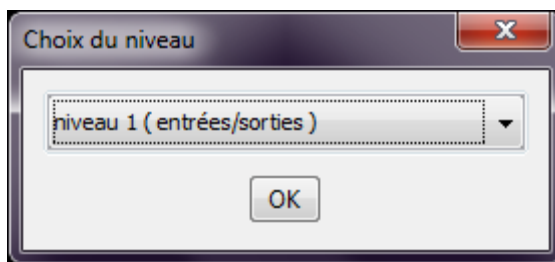


Figure 4-27 : Interface de choix d'un niveau

6. CONCLUSION

Dans ce chapitre nous avons présenté le fonctionnement de notre application ainsi que les outils utilisés dans la mise en œuvre de cette application. Nous avons commencé par un petit historique concernant les outils ensuite nous avons expliqué le fonctionnement de l'application avec des fenêtres prise de l'application elle-même.

Conclusion générale

L'objectif de notre travail est de produire un outil logiciel aux étudiants, leur permettant d'apprendre comment convertir un algorithme vers un code source Java dans le domaine de programmation.

Comme l'objectif fixé au départ, était de réaliser un simulateur de conversion, notre étude sur le domaine d'algorithmique était primordiale afin d'entamer ensuite l'aspect théorique de l'E.A.O, nous avons pris comme représentation d'algorithme la forme graphique « organigramme », et nous avons pris comme modèle d'enseignement l'approche behavioriste.

Nous avons implémenté un outil qui facilite l'apprentissage et l'enseignement pour les apprenants et pour les enseignants. Nous avons utilisé l'environnement Java comme moyen de développement.

Notre logiciel n'est que dans sa version initiale. Notre prochaine étape consiste à faire une évaluation de cet outil, faire l'analyse sémantique des solutions des algorithmes introduites par l'apprenant, ainsi, proposer d'autres améliorations en ajoutant d'autres fonctionnalités.

Bibliographie

- [1] Nicolas Guibert, Patrick Girard, Programmation sur Exemple et Enseignement assisté Par ordinateur de l'algorithmique : le projet MELBA
- [2] Saint Maximin, les Systèmes Numériques : Implantation D'un Algorithme En Langage Structure.
- [3] N. Jacon, livre INTRODUCTION À L'ALGORITHMIQUE.
- [4] <http://www.futura-sciences.com/tech/definitions/informatique-algorithmique-411> « Consulter le 4/3/2017 ».
- [5] <http://www.gladir.com/CODER/ALGO/historique.htm> « consulter le 23/2/2017 ».
- [6] <http://www.commentcamarche.net/contents/6-algorithme-definition-et-introduction> « Consulter le 23/2/2017 ».
- [7] Marie-Agnès peraldi-frati, Cours Algorithmique Et Programmation Informatique».
- [8] Cours d'informatique, Université Mohammed V, 2007/2008
- [9] dictionnaire Url : <http://dictionnaire.education/fr/pseudo-code> « Consulter le 04/03/2017 ».
- [10] <http://www.maxicours.com/soutien-scolaire/information-et-gestion/1re-stg/213613.html>
- [11] Algorithmique et algorithme.pdf
- [12] <http://troumad.developpez.com/C/algorigrammes/> « consulter le 19/01/2017 ».
- [13] <http://frederic.redonnet.free.fr/cours/INFOBTS/algo/algocours.htm>
- [14] http://www.athtek.com/flowchart-to-code.html#.WLR9sW_hDIU « consulter le 04/03/2017 ».
- [15] <http://www.flowgorithm.org/about/index.htm>. « Consulter le 25/03/2017 ».
- [16] <http://www.xmlmath.net/algobox/doc.html>. « Consulter le 04/03/2017 ».
- [17] Marco Lavoie, Guide d'utilisation de LARP Marco, 2008.
- [18] Gérard Barnier (1996), Théories de l'apprentissage et pratiques d'enseignement.
- [19] Far Hadjar, Ghilan Sabrina, Master Un simulateur graphique de protocole de Liaisons de donnée dédiée à l'apprentissage et l'enseignement, 2013.
- [20] apprentissage assiste par ordinateur, URL : <http://www.thecanadianencyclopedia.com/articles/fr/apprentissage-assiste-par-ordinateur>. « Consulter le 07/03/2017 ».
- [21] <http://www.pedagonet.com/other/STRTGIE.htm> « Consulté le 14/06/2017 ».
- [22] Jannick TRUNKENWALD, Des relations multiformes, étroites et fécondes, Conférence 2 mathématiques et informatique.
- [23] Annick Caron Ph D Programme d'aide à la recherche sur l'enseignement et l'apprentissage

- [24] EAO, URL: <http://www.sites.fse.ulaval.ca/reveduc/html/vol1/no2/eao.html>
«Consulté le 19/03/2013».
- [25] URL : <http://www.enap.ca/didactheque/htmlfra/outils/informateur/seminairenov98/methode/ordinateur.htm> «Consulté le 20/03/2013».
- [26] <http://josialekenne.over-blog.com/article-33705840.html>. Consulté le 06/03/2017
- [27] <http://www.tice-education.fr/index.php/tous-les-articles-et-ressources/articles-informatiques/355-ordidac-logiciel-denseignement-assiste-par-ordinateur> « Consulté le 03/03/2017 ».
- [28] http://imss-www.upmf-grenoble.fr/prevert/SpecialiteDEMS/Cours%202007/UE1/Theories_Apprentissage_master.pdf. «Consulté le 06/03/2017».
- [29] Dr Fatima BENDELLA USTO-MBDe l'E.A.O à l'E.I.A.H, définitions et fonctionnalités d'un EIAH, 2011.
- [30] <http://www.tice1d.84.ac-aix-marseille.fr/spip/spip.php?article26> « Consulté le 03/03/2017 ».
- [31] Teachmaster, URL: <http://www.babylon.com/definition/Teachmaster/French>
«Consulter le 18/03/2017».
- [32] Cabri-Géomètre , URL : http://fr.wikipedia.org/wiki/Cabri_G%C3%A9om%C3%A8tre
«Consulté le 18/03/2017».
- [33] François FOUACHE (Promotion 2009-2011), La Formation Musicale Assistée par Ordinateur (FMAO) – Atouts et limites
- [34] Dr KHOLLADI Mohamed-Khair-Eddine, Cours de Magister de EIAO , 2001.
- [35] Erica de Vries, Les logiciels d'apprentissage : panoplie ou éventail
- [36] Categories of Computer-Assisted <https://www.cs.csustan.edu/~lamie/sed590/CSUS%20Categories%20of%20CAI.htm>
« Consulté le 07/02/2017 ».
- [37] <https://www.supinfo.com/articles/single/3829-connexion-reseau-local-avec-wamp-mamp-xampp> « Consulté le 15/02/2017 ».
- [38] manuel de l'utilisation JGraphx.
- [39] <http://cynober.developpez.com/tutoriel/java/xml/jdom/> « Consulté le 15/03/2017 ».
- [40] <http://www.gallika.net/ressources-outils/examens/a%CE%A3%CE%95%CE%A0/a%CE%A3%CE%95%CE%A0/didactique-specifique/article/le-triangle-pedagogique-de-jean/> « Consulté le 15/03/2017 ».

[41]<http://design-patterns.fr/introduction-a-la-programmation-orientee-objet> « Consulté le 25/5/2017 ».