

الجمهورية الجزائرية الديمقراطية الشعبية

République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université 8 Mai 1945 - Guelma -

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

Département d'Informatique



Mémoire de Fin d'Etudes Master

Filière : Informatique

Option : Sciences et Technologies de l'Information et de la Communication

Thème

**Une Nouvelle Méthode d'Apprentissage
Collaboratif En Ligne de
la Programmation Orientée Objet**

Encadré par :

Pr. LAFIFI Yacine

Présenté par :

BENYOUNES Ali

2020

Dédicaces

Je dédie ce travail à mes parents, pour tous les sacrifices, les conseils, les encouragements et le soutien. Ma mère qui a dirigé mon éducation, depuis l'enfance, avec grande responsabilité. Mon père le Professeur Dr. BENYOUNES Abdelaziz qui a dirigé mon cursus universitaire grâce à ces connaissances et son expérience. Mes frères qui m'ont aidé, avec leurs connaissances dans divers domaines, grâce à leurs différentes spécialités. Je dédie aussi ce travail, à mon enseignant Dr. BENABDALLAH AHCÈNE, qui n'a pas hésité à m'aider, à mes amis ABDERRAHIM OUSSAMA, GHRICI ABDELLAH et CHOUINI MED EL MOUNSIFF, pour leur grande aide dès la première année universitaire. Enfin, à tous mes enseignants de mon cursus éducatif.

Mr. BENYOUNES Ali

Remerciements

A Dieu Allah

L'éternel tout puissant, le Clément, le Miséricordieux.

A notre **Prophète MOHAMED** (PSSL).

A mon enseignant encadreur Monsieur **Pr. LAFIFI Yacine**,
de l'**Université, 8 mai 1945 Guelma**.

Qui a accepté de m'encadrer avec disponibilité et bienveillance.

Pour son enseignement, sa patience, et ses précieux conseils.

A mon enseignant co-encadreur Monsieur **BOUDJAHM Rochdi**.

Aux enseignants qui ont accepté d'examiner ce travail.

Je tiens aussi, à remercier toutes les personnes qui m'ont aidé à la réalisation de ce travail, en commençant par les enseignants de mon cursus universitaire, jusqu'à mes collègues.

Résumé

L'application des **Nouvelles Technologies** de l'**Information** et de la **Communication** (**NTIC**) dans le domaine de l'apprentissage a conduit à la naissance d'un nouveau domaine auto-éducatif appelé **e-Learning**. Ce style d'apprentissage a été appliqué dans diverses disciplines telles que l'Informatique avec ses différents modules pédagogiques, parmi eux nous citons l'algorithmique et la **Programmation Orientée Objet (POO)**. Ces derniers ont été jugés comme difficiles par plusieurs étudiants surtout en début de cursus universitaire. En outre, ces modules ont posé toujours un problème aux apprenants comme aux nouveaux programmeurs. Donc, il était souhaitable de proposer de nouvelles méthodes et stratégies d'apprentissage en ligne facilitant l'acquisition et la maîtrise des concepts de la **Programmation Orientée Objet**.

Une étude bibliographique sur l'**Apprentissage Collaboratif** ainsi que les différentes méthodes d'apprentissage à distance de la **POO**, nous a motivée de les adopter pour proposer une nouvelle méthode d'apprentissage pour ces concepts. Cette méthode d'apprentissage à distance nommée **LISA (Learn, Imagine, Share & Approve)**. Elle englobe un ensemble d'activités pédagogiques ordonnées. Ces activités sont adoptées à l'**Apprentissage Collaboratif** des concepts de la **Programmation Orientée Objet**. Pour chaque étape de cette méthode un ensemble de tâches pédagogiques sont proposées, comme la méthode de regroupement des apprenants ainsi que l'évaluation des contributions des apprenants.

Afin de valider et tester l'approche proposée, un système a été implémenté qui s'appelle **POOLISA**. Ce dernier est le fruit d'une conception basée sur le **Langage de Modélisation Unifiée (UML)**. Les résultats encourageants de ce travail permettent d'ouvrir de nouvelles pistes de recherche dans l'axe abordé, avec la perspective d'une amélioration au niveau de la concrétisation (l'implémentation) de la deuxième étape (**Imagine**) de notre méthode **LISA** et la possibilité de tester le système **POOLISA** avec un nombre plus important d'étudiants d'informatique et / ou des débutants programmeurs.

Mots clés : NTIC - E-learning - Apprentissage Collaboratif - Programmation Orientée Objet (POO) - LISA - POOLISA

Abstract

The application of new **Information and Communication Technologies (NTIC)** in the field of learning has led to the birth of a new self-educational field called *e-Learning*. This learning style has been applied in various disciplines such as Computer Science with its various teaching modules, among them we mention algorithmic and **Object Oriented Programming (OOP)**. The latter were considered difficult by several students especially at the beginning of the university course. In addition, these modules have always posed a problem for both learners and new programmers. It was therefore desirable to propose new *e-learning* methods and strategies that would facilitate the acquisition and mastery of the concepts of **Object-Oriented Programming**.

A bibliographic study on **Collaborative Learning** and the different methods of distance learning of the **OOP**, motivated us to adopt them to propose a new method of learning for these concepts. This method of distance learning named **LISA** (*Learn, Imagine, Share & Approve*). It encompasses a set of orderly educational activities. These activities are adopted for **Collaborative Learning** of the concepts of **Object-Oriented Programming**. For each step of this method a set of pedagogical tasks are proposed, such as the method of grouping learners and the evaluation of the contributions of learners.

In order to validate and test the proposed approach, a system has been implemented which is called **POOLISA**. The latter is the result of a design based on the **Unified Modeling Language (UML)**. The encouraging results of this work allow us to open up new research tracks in the axis discussed, with the prospect of an improvement in the concretization (implementation) of the second stage (*Imagine*) of our **LISA** method and the possibility of testing the **POOLISA** system with a larger number of computer science students and / or beginner programmers.

Keywords: NTIC - E-learning - Collaborative Learning - Object Oriented Programming (OOP) - LISA - POOLISA.

ملخص

أدى تطبيق تكنولوجيات المعلومات والاتصالات الجديدة (NTIC) في مجال التعلم إلى ولادة مجال جديد للتعلم الذاتي تحت مسمى **التعلم عن بعد**. تم تطبيق هذا النمط من التعلم في تخصصات علمية متعددة، من بينها علوم الكمبيوتر بوحدها التعليمية المختلفة، نذكر منها علوم الخوارزميات و البرمجة الشيئية (POO). هاته الاخيرة التي اعتبرها العديد من الطلاب صعبة خاصة في بداية مشوارهم الجامعي، و التي دائماً ما تطرح مشكلة للمتعلمين وكذا المبرمجين الجدد. لذلك ، كان من المستحسن اقتراح أساليب واستراتيجيات **تعلم عن بعد** جديدة، تسهل اكتساب وإتقان مفاهيم مقياس البرمجة الشيئية.

حفزتنا دراسة بيلوغرافية حول **التعلم الجماعي** وطرق **التعلم عن بعد** المختلفة لوحدة ال **POO**، على الارتكاز على هاته الاخيرة لاقتراح طريقة تعلم جديدة لهاته المفاهيم. طريقة **التعلم عن بعد** هذه تسمى **LISA** (تعلم، تخيل، شارك و اعتماد). اين تشمل مجموعة من الأنشطة التعليمية المتسلسلة. تم اعتماد هذه الأنشطة التعليمية في **التعلم الجماعي** لمفاهيم البرمجة الشيئية. لكل خطوة من خطوات طريقة التعلم هذه، مجموعة من المهام التعليمية المقترحة، كطريقة تجميع المتعلمين و كذا تقييم مساهمات المتعلم.

من أجل التحقق من صحة النهج المقترح واختباره، تم تطوير نظام يسمى **POOLISA**. هذا النظام الذي هو نتيجة تصميم معتمد على لغة النمذجة الموحدة (UML). تسمح لنا النتائج المشجعة لهذا العمل بفتح افاق بحث جديدة في هذا المجال، مع نوايا تحقيق تحسين في الافق على مستوى تجسيد (تنفيذ) الخطوة الثانية (تخيل) لطريقة **LISA** الخاصة بنا. بالإضافة إلى إمكانية اختبار نظام **POOLISA** على عدد معتبر من طلاب علوم الكمبيوتر و / أو المبرمجين المبتدئين.

كلمات مفتاحية : تكنولوجيات المعلومات والاتصالات الجديدة (NTIC) - التعلم عن بعد - التعلم الجماعي - البرمجة

الشيئية (POO) - LISA - POOLISA.

Sommaire

Introduction générale	1
------------------------------------	----------

Chapitre I. Etat de l'Art

1. Introduction	3
2. Apprentissage En ligne (E-Learning)	3
2.1. Définition	3
2.2. Plateforme du E-Learning	4
3. Apprentissage Collaboratif Assisté par Ordinateur (ACAO)	5
3.1. Définition de l'Apprentissage Collaboratif	6
3.2. Dimensions de l'Apprentissage Collaboratif	6
3.3. Définition de l'Apprentissage Collaboratif Assisté par Ordinateur	6
3.4. Avantages de l'Apprentissage Collaboratif	7
3.5. Obstacles de l'Application de l'Apprentissage Collaboratif	8
4. L'Apprentissage de l'Algorithmique et la Programmation Informatique	9
4.1. L'Apprentissage de la Programmation Orientée Objet	9
4.2. Difficultés d'Apprendre la Programmation Orientée Objet	10
4.3. Approches et Environnements d'Apprentissage de l'Algorithmique et la Programmation Informatique	12
4.4. Synthèse	18
5. Conclusion	21

Chapitre II. Conception d'un système d'apprentissage collaboratif à distance de la POO

1. Introduction	22
2. Problématique et objectifs de la recherche	22
2.1. Problématique de la recherche	22
2.2. Objectifs du travail de recherche	24
3. Etude de cas sur l'Apprentissage de la Programmation Orientée Objet	25
3.1. Interprétation des statistiques obtenues	28
4. Description de l'approche LISA : Learn, Imagine, Share and Approve	29
4.1. La structure pédagogique du module à étudier	30
4.2. L : Learn (Apprendre)	31
4.2.1. Description de l'Apprentissage Collaboratif appliqué	34
4.2.1.1. La Méthode de Regroupement	34
4.3. I : Imagine (Imaginer)	35
4.4. S : Share (Partager)	37
4.5. A : Approve (Approuver)	38
5. Conception du système	39
5.1. Identification des acteurs	40
5.2. Identification des cas d'utilisation des acteurs	40
6. Architecture du système	42
6.1. Espace Enseignant	42
6.2. Espace Etudiant / Apprenant	43
6.2. Espace Administration	44
6.2. Les Entités du Système POOLISA	44
7. Conclusion	56

Chapitre III. Implémentation du Système POOLISA

1. Introduction	57
2. Outils utilisés	57
2.1. Au niveau de la conception	57
2.2. Au niveau de l'implémentation	58
3. Interfaces et fonctionnalités du système POOLISA	61
4. Expérimentation	65
4.1. Méthodologie	65
4.2. Résultats et discussion	66
5. Conclusion	67
Conclusion générale et perspectives	68
Références bibliographiques	70

Annexes

Annexe 1. L'échelle de Likert	73
Annexe 2. Questionnaire de l'étude de cas réalisée	74
Annexe 3. Etat des résultats de réponses aux 26 questions du questionnaire	80
Annexe 4. Autres interfaces et fonctionnalités du système POOLISA	93
Annexe 5. Test t de student pour échantillons appariés	96

Liste des Tableaux

Chapitre I

Tableau 1. Etat de comparaison entre les différents environnements et systèmes d'apprentissage	20
---	----

Chapitre II

Tableau 2. Description de l'entité User	45
Tableau 3. Description de l'entité Cours	45
Tableau 4. Description de l'entité Test	46
Tableau 5. Description de l'entité Chapitre	46
Tableau 6. Description de l'entité Concept	47
Tableau 7. Description de l'entité Titre	47
Tableau 8. Description de l'entité Question	48
Tableau 9. Description de l'entité Réponse	48
Tableau 10. Description de l'entité Score	49
Tableau 11. Description de l'entité Fichier	49
Tableau 12. Description de l'entité Messages	50
Tableau 13. Description de l'entité Groupe	50
Tableau 14. Description de l'entité UserGroupe	51
Tableau 15. Description de l'entité Chat	51
Tableau 16. Description de l'entité Exemples	52
Tableau 17. Description de l'entité Etu_Termin	52
Tableau 18. Description de l'entité Projet	53
Tableau 19. Description de l'entité Solution	53
Tableau 20. Description de l'entité Rating	54
Tableau 21. Description de l'entité Solution_grp	54

Chapitre III

<u>Tableau 22.</u> Statistiques des notes obtenues	65
<u>Tableau 23.</u> Statistique du groupe avant / après utilisation du système	66
<u>Tableau 24.</u> Statistique t-test	66

Liste des Figures

Chapitre I

Figure 1. Exemple d'architecture de plateforme pour la Formation à Distance	5
Figure 2. Le travail collaboratif	6
Figure 3. La différence entre programmation orientée objet et programmation procédurale	10
Figure 4. L'environnement de programmation Pygmalion ; la figure représente la conception d'un programme calculant $n!$	13
Figure 5. Vue d'ensemble des composants de l'environnement MELBA	14
Figure 6. L'interface BlueJ	15
Figure 7. Invoquer des méthodes dans BlueJ	17
Figure 8. Interface OoP	18

Chapitre II

Figure 9. Les phases de paradigme de POO	23
Figure 10. Résultats des réponses à la question 9	25
Figure 11. Résultats des réponses à la question 13	26
Figure 12. Résultats des réponses à la question 16	26
Figure 13. Résultats des réponses à la question 20	27
Figure 14. Résultats des réponses à la question 24	27
Figure 15. Principe de l'approche LISA	30
Figure 16. La structure pédagogique du module étudié par LISA	31
Figure 17. La structure pédagogique du module de POO	33
Figure 18. Principe de l'étape I (Imagine)	35
Figure 19. Exemple de résolution du problème par Imagine	36
Figure 20. Principe de l'étape S (Share)	37
Figure 21. Principe de l'étape A (Approve)	39

Figure 22. Diagramme des cas d'utilisation du système POOLISA	41
Figure 23. Diagramme des cas d'utilisation d'espace apprenant	42
Figure 24. Base de Données du Système POOLISA	55

Chapitre III

Figure 25. Interface de logiciel StarUml	57
Figure 26. Interface Dreamweaver	58
Figure 27. Interface PhpMyAdmin	59
Figure 28. Accueil du Plateforme POOLISA	61
Figure 29. Contenu du cours POO	62
Figure 30. Learn – Liste de chapitres de cours POO	62
Figure 31. Learn – Contenu d'un chapitre	63
Figure 32. Share & Approve – Contenu d'un Projet Final	63
Figure 33. Share & Approve – Résolution Conceptuelle d'un Projet final	64
Figure 34. Share & Approve – Noter la solution d'un Co-Apprenant	64

Introduction générale

Le développement technologique et l'intégration des **TIC** (**T**echnologies de l'**I**nformation et de la **C**ommunication) ont joué un grand rôle dans le changement et l'évolution de l'enseignement et l'apprentissage classiques. Comme résultat, un nouveau domaine est apparu, qui est l'apprentissage électronique ou e-Learning. En effet, ce dernier se présente comme une nouvelle forme d'apprentissage à distance qui repose sur la mise à disposition de contenus, organisés en modules qui peuvent être assemblés en parcours de formation personnalisée, via des réseaux d'internet ou d'intranet.

Durant ces dernières années, le e-Learning a été appliqué dans plusieurs et différentes disciplines scientifiques, où il a prouvé son efficacité pour rendre l'apprentissage des apprenants plus souple et plus facile. Ceci, essentiellement durant les situations difficiles ou d'urgence, dont le monde est actuellement plus-que jamais témoin, comme les guerres ou les épidémies telles que celle de la COVID19. En effet, l'Informatique est l'une des disciplines qui ont marqué un grand avancement dans l'exploitation de cette nouvelle forme d'apprentissage, étant donné la capacité de apprenants à accepter, et de bien utiliser les outils offerts par les plateformes dédiées à l'apprentissage en ligne facilement et correctement.

L'application du e-Learning dans l'Informatique a tout d'abord commencé par l'apprentissage de l'Algorithmique vu que cette matière est le pilier principal de la formation de n'importe quel informaticien. Pour ce faire, plusieurs environnements et systèmes d'apprentissage à distance ont été conçus et développés. Ensuite, avec le développement des langages de programmation informatique de différentes générations d'une manière générale, et la **P**rogrammation **O**rientée **O**bjets (**POO**) en particulier, cette dernière a fait l'objet de quelques études et recherches. Ces dernières ont été concrétisées dans divers systèmes d'apprentissage, en mode hors ligne ou en ligne, où chacun a essayé de résoudre l'une des difficultés d'apprentissage de ce module pédagogique (la **POO**) selon sa vision.

C'est ainsi que l'objectif de notre travail de recherche, a été orienté sur le développement d'une méthode d'apprentissage à distance, adéquate à l'apprentissage de la **POO**, tout en exploitant le concept de l'apprentissage collaboratif. A notre connaissance, ce dernier n'a pas été abordé dans aucune des approches d'apprentissage développées dans ce cadre.

La nouvelle méthode d'apprentissage collaboratif proposée pour l'apprentissage des concepts de la programmation orientée objet doit tenir compte de la structure de matière à enseigner, les interactions entre les apprenants lors de la résolution collectives des problèmes et la méthode de leur regroupement. Tous ces points ont été répartis en un ensemble d'étapes composant la méthode proposée qui s'appelle : LISA pour (*Learn, Imagine, Share & Approve*). Cette méthode a été adoptée par un système d'apprentissage collaboratif des concepts de la POO baptisé POOLISA. Ce système peut être utilisé dans un cadre officiel pour les étudiants de la spécialité informatique ou autre promotion étudiant ou apprenant la programmation orientée objet, comme il peut être utilisé par les novices de la POO en forme libre.

Pour atteindre nos objectifs, notre travail a été organisé en plusieurs parties sous forme de chapitres ayant pour objet :

- Chapitre I : Etat de l'art

Il propose d'une manière synthétique une révision bibliographique sur l'apprentissage en ligne e-Learning, l'apprentissage collaboratif, l'apprentissage de l'algorithmique et la programmation informatique. Ceci, en plus de l'apprentissage de la **Programmation Orientée Objet (POO)**, avec ses difficultés ainsi que les approches et les environnements d'apprentissage de l'algorithmique, la programmation en général et la **POO** en particulier.

- Chapitre II : Conception d'un système d'apprentissage collaboratif dédié à la POO

Ce chapitre présente une étude sur la situation actuelle de l'apprentissage de la POO à travers l'analyse des résultats d'un questionnaire établi à cet effet. Ensuite, il donne une description détaillée de l'approche développée (*LISA*) avec ses différentes étapes et leurs principes de fonctionnement en détail. Enfin, il expose les différentes étapes de la conception du système adoptant cette approche ainsi que les diagrammes d'utilisation.

- Chapitre III : Implémentation du système conçu POOLISA

Il montre les outils utilisés dans le développement de ce système d'apprentissage à distance, ses interfaces et ses principales fonctionnalités.

Chapitre I. Etat de l'Art

Chapitre I. Etat de l'Art

1. Introduction

Le développement technologique et l'intégration des **TIC** (Technologies de l'Information et de la Communication) dans la vie quotidienne ont joué un grand rôle dans le changement et l'évolution de plusieurs et différents secteurs et activités. Parmi ces derniers, l'enseignement et l'apprentissage, qui ont évolué de la forme classique, vers une nouvelle forme nommée l'**EAO** (Enseignement Assisté par Ordinateur) ou le **e-learning**.

Ainsi, la **diffusion d'Internet** a été le déclencheur du développement du e-learning ; et dès l'invention des premiers E-mails, des pionniers ont utilisé la toile pour échanger des savoirs.

En effet, la première école supérieure qui a proposé un cursus en ligne qui est **CALCampus**, a été créée en 1994. Peu de temps après, l'université de Virginia Beach, délivre ses premiers PhD, grâce à des cours entièrement dispensés par courrier électronique. Autre innovation technique de la décennie, la **webcam**, qui vient profondément modifier la nature de la relation entre professeurs et élèves, en permettant un suivi des cours en temps réel, et sa démocratisation progressive facilitent le dialogue et l'interactivité (**Href 1**).

L'objectif de ce chapitre est de présenter les principes de base de l'apprentissage en ligne individuel ou collaboratif, ses acteurs humains, les plateformes utilisés pour assurer ce type d'apprentissage. En plus, nous présentons quelques travaux de recherche effectués dans le cadre de l'apprentissage de l'algorithmique, et la programmation orientée objet. Enfin, nous terminons ce chapitre par la présentation d'une synthèse des travaux existants dans cet axe.

2. Apprentissage En Ligne (E-Learning)

2.1. Définition

Dans les années 2000, grâce à la baisse du coût du matériel informatique et la révolution des logiciels libres, l'e-learning se démocratise et devient mieux connu dans le monde. A cet effet, plusieurs définitions pour ce concept sont apparues.

Selon la **Commission Européenne (2001)** l'apprentissage en ligne dit « **e-learning** » est : « *l'utilisation des nouvelles technologies multimédias de l'Internet pour améliorer la qualité de l'apprentissage en facilitant d'une part l'accès à des ressources et à des services, d'autre part les échanges et la collaboration à distance* ».

Une autre définition repose sur la notion de **distance et technologies** : « *L'e-learning est l'utilisation des médias électroniques à des fins d'apprentissage variés qui vont des fonctions complémentaires dans les salles de classe conventionnelles, à la substitution complète des réunions en face-à-face par des rencontres en ligne* » (**Guri-Rosenblit, 2005; Modifiée**).

On trouve aussi une définition **orientée communication** : « *Le e-learning est une éducation qui utilise des systèmes de communication informatisés comme environnement de communication, d'échange d'informations et d'interaction entre les étudiants et les instructeurs* » (**Bermejo, 2005; Modifiée**).

Malgré les différentes visions des définitions précédentes, on peut constater que le e-learning consiste à utiliser les ressources de l'informatique et de l'Internet, pour fournir et acquérir, à distance, des connaissances.

Ainsi, pour assurer cet apprentissage en ligne ou à distance, il faut un outil qui organise et assure la communication enseignant-apprenant, et essentiellement le partage de la connaissance par ses différentes formes, ce qu'est le rôle de la plateforme e-learning.

2.2. Plateforme du E-Learning

Une **plateforme du e-learning** est un **outil** déployé sur le web, mettant à disposition d'une communauté d'apprenants des ressources pour se former dans différents domaines. Cette plateforme dont la fonction consiste à gérer un processus d'apprentissage, peut prendre l'appellation de **Learning Management System (LMS)** ou **Learning Support System (LSS)**.

Ce type de plateforme permet d'héberger des contenus pédagogiques multimédia, diffuser des modules e-learning, mais également d'accompagner les apprenants et de suivre leur progression tout au long de leur formation (suivi du parcours, enregistrement des actions, évaluation...). L'utilisation d'une plateforme de formation en ligne, rend donc possible la

gestion de l'accès aux modules de formation, l'hébergement de ces modules, ainsi qu'un suivi individualisé des utilisateurs (**Href 2**).

Aussi, l'objectif de ce type des plateformes est de faciliter les rôles et les fonctions tenus par chacun de ces acteurs : enseignant, concepteur informatique, tuteur, apprenant et administrateur, comme démontré ci-dessous dans la **Figure 1**.

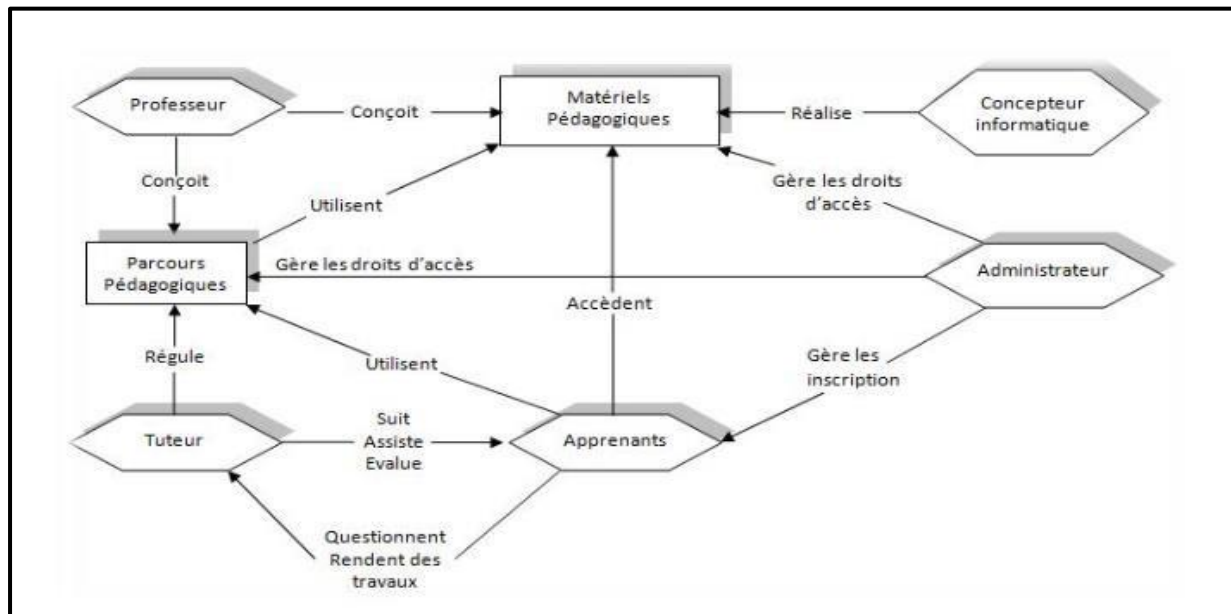


Figure 1. Exemple d'architecture de plateforme pour la Formation à Distance (**Href 3**).

3. Apprentissage Collaboratif Assisté par Ordinateur (ACAO)

Après le succès de l'apprentissage en ligne « e-learning » comme une nouvelle forme d'apprentissage, les chercheurs étaient motivés à exploiter les TIC dans l'apprentissage collaboratif classique ; et par conséquent, l'apparition d'un nouveau domaine d'apprentissage, qui est l'Apprentissage Collaboratif Assisté par Ordinateur (ACAO) ou "Computer-Supported Collaborative Learning" (CSCL).

Avant de présenter l'Apprentissage Collaboratif Assisté par Ordinateur (ACAO), il y a lieu de présenter tout d'abord, le concept d'Apprentissage Collaboratif en général.

3.1. Définition de l'Apprentissage Collaboratif

Selon **Rogalski (1998)** la collaboration en général désigne « *les situations où les acteurs partagent les mêmes buts tout au long de la réalisation d'une tâche* ». Les acteurs maintiennent alors le même but, mais également les mêmes sous buts, comme présenté dans la **Figure 2**.

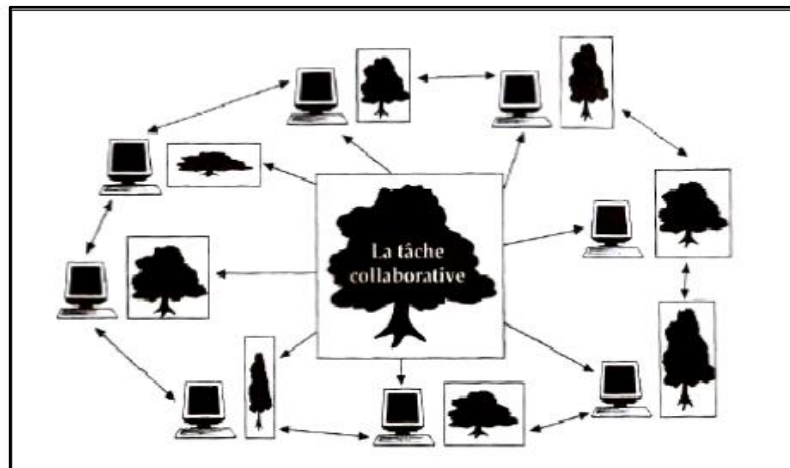


Figure 2. Le travail collaboratif (**Henri et Lundgren-Cayrol, 2001**).

Pour l'apprentissage collaboratif, **Dillenbourg (1999)** donne une définition à ce concept comme « *une situation dans laquelle deux ou plusieurs personnes apprennent ou essayent d'apprendre quelque chose ensemble* ».

3.2. Dimensions de l'Apprentissage Collaboratif

La notion de dimensions ici, veut dire les caractéristiques de base de l'apprentissage collaboratif, lesquelles commencent par le type de la tâche à accomplir, l'endroit (présentiel, à distance) des apprenants qui collaborent dans la réalisation de cette tâche, la taille de groupe d'apprenants (nombre des membres) et les critères de regroupement des apprenants (Regroupement au hasard, par affinités, par proximité, par champs d'intérêt et par enseignant).

3.3. Définition de l'Apprentissage Collaboratif Assisté par Ordinateur

L'Apprentissage Collaboratif Assisté par Ordinateur (ACAO), plus connu sous l'abréviation anglo-saxonne **CSCL** « *Computer Supported Collaborative Learning* » n'est pas strictement défini, et regroupe une multitude de recherches et d'approches ; comme :

l'apprentissage humain (Learning), l'apprentissage collaboratif (Collaborative Learning), et l'utilisation de l'ordinateur comme moyen de soutien pour l'apprentissage collaboratif (Computer Supported Collaborative Learning) (Mehnaoui, 2018).

En effet, ce concept a comme but principal, l'exploitation de l'ordinateur pour assurer une bonne collaboration des apprenants dans un apprentissage à distance.

3.4. Avantages de l'Apprentissage Collaboratif

Plusieurs chercheurs ont confirmé et insisté sur la grande importance et les différents avantages, de l'apprentissage collaboratif dans leurs travaux. Parmi eux, **Laal et Ghodsi (2012)** qui dans leur étude ont eu à résumer et à classer les avantages de l'apprentissage collaboratif, en quatre grandes catégories : social, psychologique, académique et évaluation, comme détaillé ci-après :

- **Avantages sociaux :**

- L'apprentissage collaboratif (AC) aide à développer un système de soutien social pour les apprenants
- L'AC conduit à développer la compréhension de la diversité parmi les étudiants et le personnel
- L'AC crée une atmosphère positive pour la modélisation et la pratique de la coopération
- L'AC développe des communautés d'apprentissage

- **Avantages psychologiques :**

- L'enseignement centré sur l'élève augmente l'estime de soi des élèves
- La coopération réduit l'anxiété
- L'AC développe des attitudes positives envers les enseignants

- **Avantages académiques :**

- L'AC favorise les compétences de la pensée critique
- L'AC implique activement les élèves dans le processus d'apprentissage
- Dans les situations de l'AC, Les résultats en classe sont améliorés
- Modélise les techniques appropriées de résolution de problèmes des élèves
- Les grandes conférences peuvent être personnalisées

- L'AC est particulièrement utile pour motiver les élèves dans un programme spécifique
- **Techniques alternatives d'évaluation des élèves et des enseignants :**
 - Les techniques d'enseignement collaboratif utilisent une variété d'évaluations.

Ainsi, l'apprentissage collaboratif a pour but d'améliorer la réussite des apprenants, en exploitant les compétences sociales, les relations interpersonnelles et les points forts de chaque apprenant lors des activités proposées. Ce type d'apprentissage repose sur le travail en groupes restreints, où des apprenants de capacités et de talents différents, s'efforcent d'atteindre un objectif commun ; dont le travail est structuré de façon à ce que chaque apprenant participe à l'accomplissement de la tâche proposée. De plus, les interactions sociales que l'apprentissage collaboratif favorise, incitent les apprenants à verbaliser et à reformuler leurs idées, à les confronter, à discuter et à comparer leurs façons d'apprendre. La création d'un contexte favorable à la discussion des connaissances, au sein d'un groupe de collaboration, améliore la qualité de l'apprentissage, en provoquant l'émergence de processus métacognitifs.

3.5. Obstacles de l'Application de l'Apprentissage Collaboratif

Pour arriver à profiter et à exploiter les avantages et les points forts de l'apprentissage collaboratif, il faut malheureusement confronter quelques obstacles, comme indiqué dans le travail de **Lafifi (2007)**, tels que :

- Le refus d'une demande de collaboration (on trouve des apprenants qui refusent le travail collaboratif avec les autres) ;
- La distribution des tâches entre les membres de groupe, laquelle n'est pas toujours équitable, par conséquent on trouve des apprenants qui effectuent la majorité de travail, ce qui n'est pas le but de l'apprentissage collaboratif ;
- Le regroupement des apprenants n'est pas toujours facile, surtout dans le cas où il y a des apprenants trop demandés, et d'autres non ;

- Quelques apprenants, trop demandés pour aider les autres, sentent qu'ils sont exploités et cessent de collaborer/coopérer.

4. L'Apprentissage de l'Algorithmique et la Programmation Informatique

L'algorithmique est la discipline de résolution des problèmes, ou d'une manière plus simple, c'est le pensé à une méthode ou façon systématique à procéder pour faire quelque chose. Ce pensé qui va être ensuite traduit dans un langage de programmation compréhensible par l'ordinateur; ce qui est le principe de la programmation informatique.

En outre, l'algorithmique est la base d'apprentissage de la programmation informatique selon le fameux principe qui dit « *pour écrire un bon programme, il faut écrire un bon algorithme* ». Malheureusement, il est souvent source de problème pour l'enseignant, ainsi que pour l'étudiant. En effet, l'enseignant doit trouver les méthodes adéquates pour faire assimiler des concepts assez abstraits, à des étudiants qui ne sont qu'à leur phase d'initiation (Benabbou et Hanoune, 2006).

4.1. L'Apprentissage de La Programmation Orientée Objet (POO)

L'introduction à la programmation est un module traditionnel et obligatoire dans le cursus des étudiants d'informatique. Bien que le contenu de ce module ait beaucoup évolué au fil des ans, en raison des progrès et des tendances de la technologie informatique, ainsi que de la montée et de la chute de différents langages de programmation, il y a eu l'apparition de nouveaux paradigmes de programmation, tel que la **Programmation Orientée Objet (POO)**.

Alors que la Programmation Procédurale traite les programmes comme un ensemble de données sur lesquelles agissent des procédures, la programmation orientée objet, elle, relève d'une conception définie comme des « messages » échangés par des entités appelées objets.

Ainsi, la **Figure 3** présentée ci-dessous nous renseigne sur quelques différences entre les deux styles de programmation expliqués précédemment.

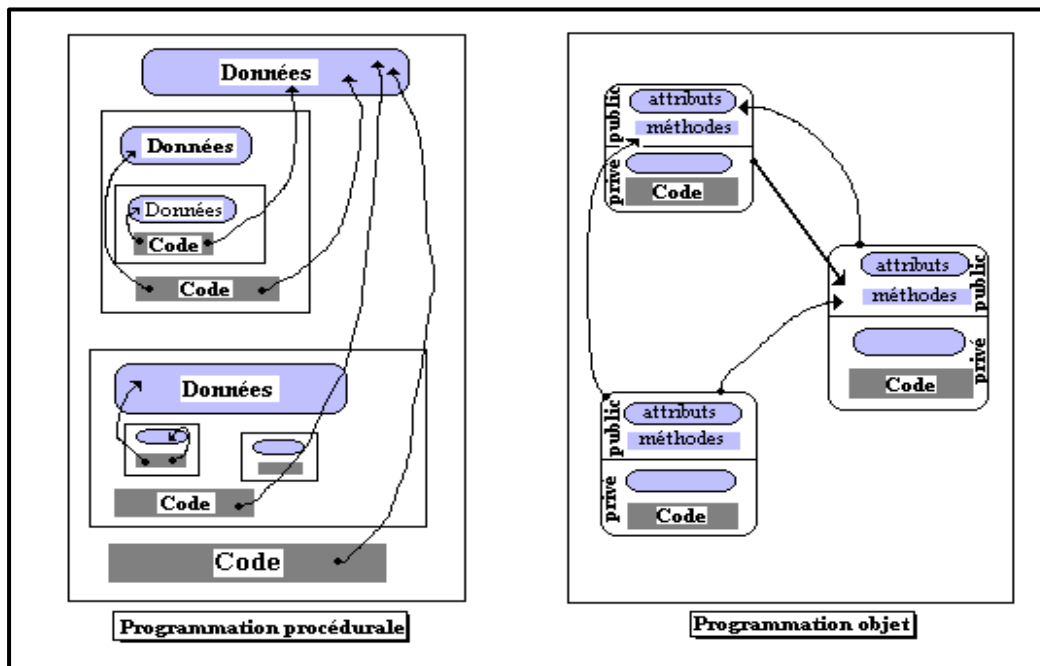


Figure 3. La différence entre programmation orientée objet et programmation procédurale ([Href 4](#)).

Malheureusement, comme dans le cas de l'apprentissage de l'algorithmique et la programmation procédurale, l'enseignement et l'apprentissage de la Programmation Orientée Objet (POO) connaissent également de nombreux problèmes et difficultés.

4.2. Difficultés d'Apprendre la Programmation Orientée Objet (POO)

Comprendre les concepts orientés objet est toujours une tâche difficile pour les étudiants, comme il est tout aussi difficile pour les enseignants de les transmettre à ces derniers. En effet, la difficulté est liée principalement au fait que les étudiants d'informatique dans leurs premiers pas sont habitués de voir et de résoudre les problèmes d'une manière itérative procédurale, laquelle est basée essentiellement sur une séquence d'instructions. Par conséquent, le passage vers une nouvelle vision, nouvelle approche, dans notre cas l'orienté objet, sera un peu difficile, spécialement avec de nouveaux concepts et de nouvelles règles comme les classes, l'invocation de constructeur, les constructeurs surchargés, l'héritage... etc.

Des recherches, comme stipulé dans le travail de **Stroustrup Bjarne (1994)**, indiquant que le programmeur prend en moyenne 6 à 18 mois pour changer son état d'esprit, à partir

d'une vue procédurale vers une vue orientée objet du monde. Ce qui dénote donc, que c'est le changement qui est difficile.

Par ailleurs, les concepts abstraits peuvent être une autre difficulté que les élèves devront affronter dans l'apprentissage de la programmation orientée objet. La classe et l'objet, sont des concepts de base dans ce module, dont il est assez difficile de trouver leurs équivalences dans la vie réelle. Ainsi, les étudiants ont donc besoin de bien allumer leurs imaginations, laquelle est malheureusement une capacité mentale non fournie pour tout le monde. Ainsi pour la compréhension des concepts de base, les étudiants ont du mal à comprendre des concepts encore plus complexes tels que l'héritage et le polymorphisme, s'ils ne peuvent même pas comprendre les concepts de base : classe et objet (**Yan Lu, 2009**).

Pour **Kölling Michael (1999)**, ce n'est pas l'orienté objet qui est la cause de problèmes, mais ce sont les outils (langages et environnements) disponibles pour l'enseigner. Pour cet auteur, les langages de programmation utilisés sont trop complexes, et les environnements de programmation (s'ils existent) sont trop confus, et que certains systèmes et langages utilisés pour l'enseignement ont été développés pour les professionnels. Ce qui rend la situation plus complexe pour les étudiants, lesquelles tombent dans la difficulté de les utiliser dans l'apprentissage de l'orienté objet.

Dans le même sens, d'autres auteurs voient que pour les programmeurs débutants, comme c'est le cas pour les étudiants universitaires, la tâche d'identifier les objets et de créer une conception orientée objet du programme est très difficile (**Robins et al., 2003**).

Un autre point très intéressant était présenté par **Livovský et Porubän (2014)**, indiquant que les enseignants qui commencent le cours de programmation en expliquant le paradigme de programmation à travers des exemples de petits codes sources, sont confrontés à un problème : les étudiants ne connaissent pas encore le langage de programmation orienté objet, et donc les enseignants doivent gérer l'explication de la syntaxe et les fonctionnalités du langage ainsi que les concepts et principes de la POO. C'est ainsi que **Livovský et Porubän (2014)** ont proposé de trouver une solution qui enseignerait les concepts de base du paradigme orienté objet au début du cours de programmation orientée objet (POO), sans forcer les étudiants à commencer par l'apprentissage du langage de programmation et l'écriture de code.

Aussi, une étude pionnière de **Sims-Knight et Upchurch. (1993)** montre qu'il est possible d'enseigner la conception orientée objet sans programmation (sans implémentation). L'étude était réalisée par un ensemble d'élèves qui ont joué un jeu informatique existant, puis ont regardé un expert qui montre comment concevoir ce jeu. L'expert a expliqué les considérations de conception d'une manière orientée objet et les élèves ont ensuite créés leurs propres conceptions pour ce jeu.

De tout ce qui précède, il est clairement illustré que l'apprentissage de l'orienté objet comme l'apprentissage de l'algorithmique est un ancien problème qui était traité par plusieurs chercheurs, et qui a motivé d'autres à approfondir dans cet axe jusqu'à atteindre des solutions (*approches, environnements et outils*) à ce problème.

4.3. Approches et Environnements d'Apprentissage de l'Algorithmique et la Programmation Informatique

Dans le cadre de l'amélioration du niveau d'apprentissage de l'algorithmique, la programmation en général et la Programmation Orientée Objet (POO) en particulier, plusieurs approches, environnements et systèmes étaient développés dans ce but. C'est dans ce sens, que nous allons essayer de citer et expliquer, dans la partie qui suit, quelques travaux reliés à cet axe.

EasyAlgo, un environnement informatique intelligent à des fins d'apprentissage et d'autoévaluation de l'algorithmique, réalisé dans le travail de **Benabbou et Hanoune (2006)**. Cet environnement repose sur une approche basée sur l'usage des exemples et de l'expérience, où le scénario d'apprentissage commence par la lecture des concepts théoriques. Ensuite, l'apprenant s'entraîne volontairement avec les exemples qui sont bien commentés, et se termine par une phase d'autoévaluation par des exercices. Cette dernière phase, a comme but de rendre l'apprenant capable de construire ses propres algorithmes. L'intelligence dans EasyAlgo est concrétisée par le recueil d'informations sur les styles d'apprentissage des apprenants pour l'adaptation du processus d'apprentissage, et sur leurs activités, telles que l'autoévaluation, où le système propose à l'apprenant des recommandations de correction d'exercice, qui varient selon les types d'erreurs (**Benabbou et Hanoune, 2006**).

L'approche « basé sur l'exemple » pour l'apprentissage de la programmation peut être illustrée par Pygmalion (Cypher et Halbert, 1993) qui est le premier environnement de programmation basé sur l'exemple. Pygmalion est un environnement de programmation visuelle « sur l'exemple » basé sur la métaphore du tableau blanc (ou noir), et sur le langage SmallTalk. Il est considéré comme un espace de travail graphique (Figure 4), où les différentes variables et paramètres apparaissent sous forme de cases, lesquelles contiennent la valeur courante de la variable. L'état du programme peut ainsi être affiché de façon exhaustive et continue, ce qui rétablit le principe de manipulation directe (Guibert et al., 2006).

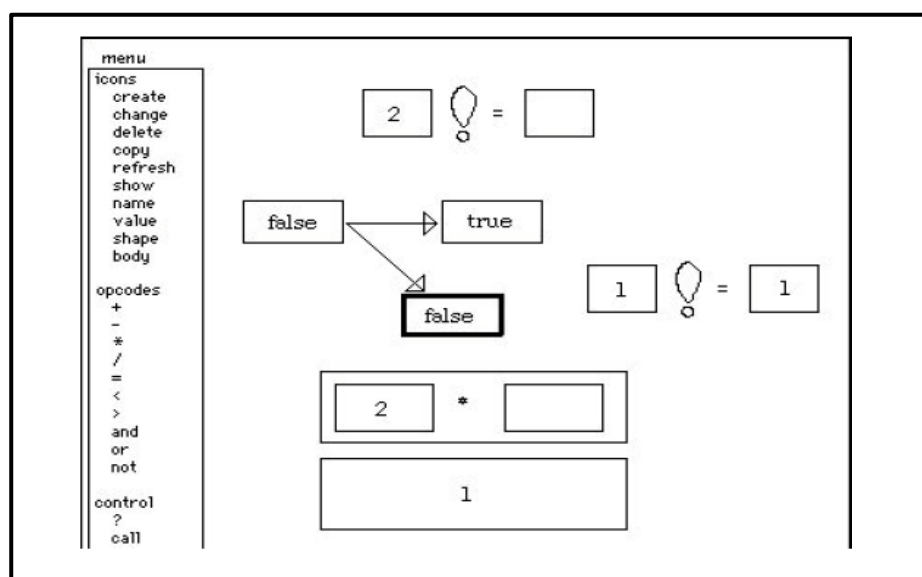


Figure 4. L'environnement de programmation Pygmalion ; la figure représente la conception d'un programme calculant $n!$ (Guibert et al., 2006).

MELBA est un environnement d'apprentissage de programmation basé sur l'approche « basé sur l'exemple » (Guibert et al., 2006). Cet environnement permet à l'étudiant d'interagir avec le système en construisant par insertions et suppressions de programme, qui est présenté sous la forme d'un arbre. A chaque exercice est associé un ensemble spécifique de commandes et de tests, accessibles par menus déroulants dans la zone opération, qui agissent sur un modèle formel du système représenté sous forme graphique dans la fenêtre d'exécution. Alternativement, il est possible d'effectuer des exercices classiques manipulant des données informatiques par l'opération d'affectation. Une fenêtre d'historique supporte une représentation de la trace d'exécution du programme. La figure ci-dessous (Figure 5) résume les composants de MELBA.

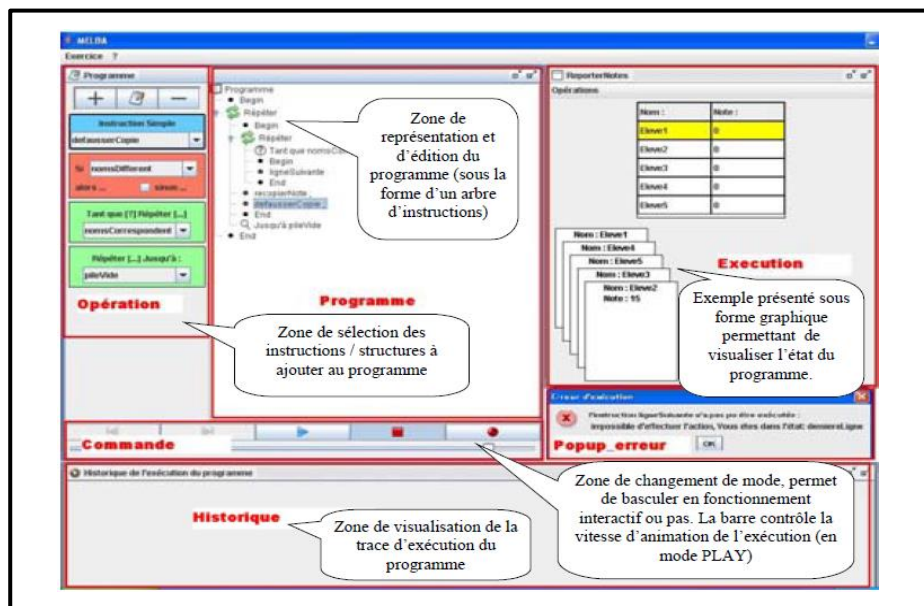


Figure 5. Vue d'ensemble des composants de l'environnement MELBA (Guibert *et al.*, 2006).

Codeacademy est une plate-forme en ligne qui a été créée en août 2011 par Ryan Bubinski et Zach Sims. Elle propose des guides et des cours gratuits dans de nombreux langages de programmation comme Python, PHP, JavaScript, Java et plusieurs autres (Href 5). Elle se considère comme une entreprise éducative qui vise à repenser la façon dont les gens éduquent. Elle contient des éléments importants tels que les forums, les glossaires; et les gens peuvent créer et publier des cours à d'autres étudiants et même les écoles intéressées peuvent accéder à du matériel pédagogique, à un système de suivi et à des activités programmées.

Rational Rose développé par IBM en 1991, est un outil puissant de modélisation visuelle pour le développement de systèmes orientés objet en utilisant le langage de modélisation unifié (UML) ; son objectif est d'unifier les différents systèmes existants en un langage de modélisation de pointe (Href 6). Outre la modélisation visuelle, Rational Rose peut également générer du code squelette en Java, C++ ou Visual Basic. Il prend en charge l'ingénierie complète aller-retour, permettant l'ingénierie inverse et la génération de diagrammes UML à partir de programmes sources.

On trouve aussi BlueJ, un environnement de développement Java qui a été développé dans le cadre d'un projet de recherche universitaire pour l'enseignement de l'orienté objet aux

débutants (Kölling, 1999). Le système est développé et maintenu par un groupe de recherche conjoint composé de l'Université Monash en Australie, de l'Institut Mærsk de l'Université du sud du Danemark et de l'Université du Kent à Canterbury au Royaume-Uni. Le projet est soutenu par Sun Microsystems. L'environnement fonctionne au-dessus du **Java Development Kit (JDK)** de Sun Microsystem en utilisant le compilateur standard et la machine virtuelle. Le but de BlueJ est de fournir un environnement facile à utiliser pour l'enseignement du langage Java, et une grande concentration a été mise sur les techniques de visualisation et d'interaction, dans le but de créer un environnement hautement interactif qui encourage l'expérimentation et l'exploration.

Les caractéristiques de **BlueJ** comprennent : un environnement entièrement intégré ; des installations d'affichage de la structure graphique (**Figure 6**) ; un éditeur intégré avec compilateur ; une machine virtuelle et débogueur ; une interface facile à utiliser ; une création d'objets interactifs et développement d'applications incrémentielles.

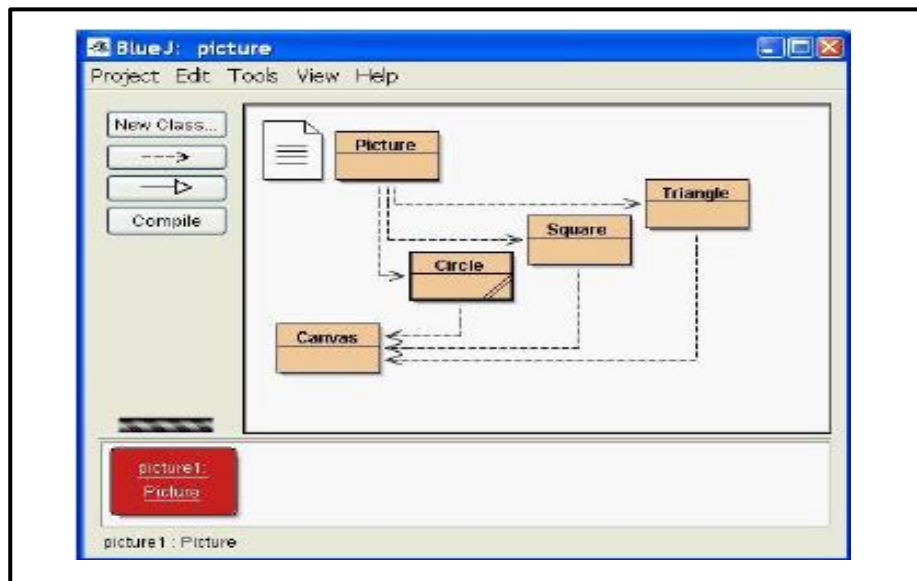


Figure 6. L'interface BlueJ (Yan Lu, 2009).

OOP-Anim est un environnement éducatif développé par **Esteves et Mendes (2004)**. Il vise à aider les étudiants à comprendre et à maîtriser les concepts de base de la programmation orientée objet. OOP-Anim est basé sur un scénario d'apprentissage qui commence par: l'entrée d'une solution (un code Java) d'un problème par l'étudiant, de la simuler et de l'animer pour visualiser comment-elle fonctionne, la corriger si nécessaire et de

la simuler à nouveau jusqu'à ce qu'une solution de ce problème soit atteinte. OOP-Anim est donc basé essentiellement sur l'approche de simulation.

WEBLOOP (A WEB-based Learning Environment for OOP) est un autre environnement d'apprentissage de la POO développé par **Cheung Ronnie (2006)** pour aider les nouveaux étudiants à apprendre la programmation orientée objet. Son apport principal par rapport aux autres environnements, c'est que les étudiants n'ont pas besoin de l'installer et de configurer différentes versions des systèmes Java sur leurs ordinateurs locaux, grâce à l'exploitation de la technologie d'Internet et du Web, où les étudiants connectés peuvent accéder aux fonctionnalités de WEBLOOP n'importe où via un navigateur Web. WEBLOOP assure la fonctionnalité de la génération des diagrammes de classes UML à partir d'un code source ou la génération inverse (code à partir d'un diagramme) comme l'outil Rational Rose. Ceci, en ajoutant la compilation en ligne et la fonctionnalité de la gestion des comptes, lesquels fournissent un espace de travail séparé pour chaque étudiant, et qui seront ensuite utilisés par les enseignants dans le suivi des progrès des élèves.

Une nouvelle approche d'enseignement de la programmation orientée objet était apparue ces dernières années, c'est l'enseignement de la POO avec les jeux, et nous allons citer certains des travaux liés à cet axe dans ce qui suit.

Greenfoot, un environnement développé par **Yan Lu (2009)**, permettant de créer des applications interactives de type simulation dans un plan bidimensionnel. Greenfoot permet l'implémentation et l'interaction avec des objets dans le contexte de scénarios. Une fois les objets créés dans Greenfoot, ils peuvent être placés de manière interactive dans un monde Greenfoot, et les utilisateurs peuvent jouer directement avec ces objets et invoquer des méthodes (**Figure 7**) dans ces objets. Le système Greenfoot est également un environnement de développement intégré (IDE) du langage Java ; il contient un éditeur, un compilateur et un débogueur (**Href 7**).

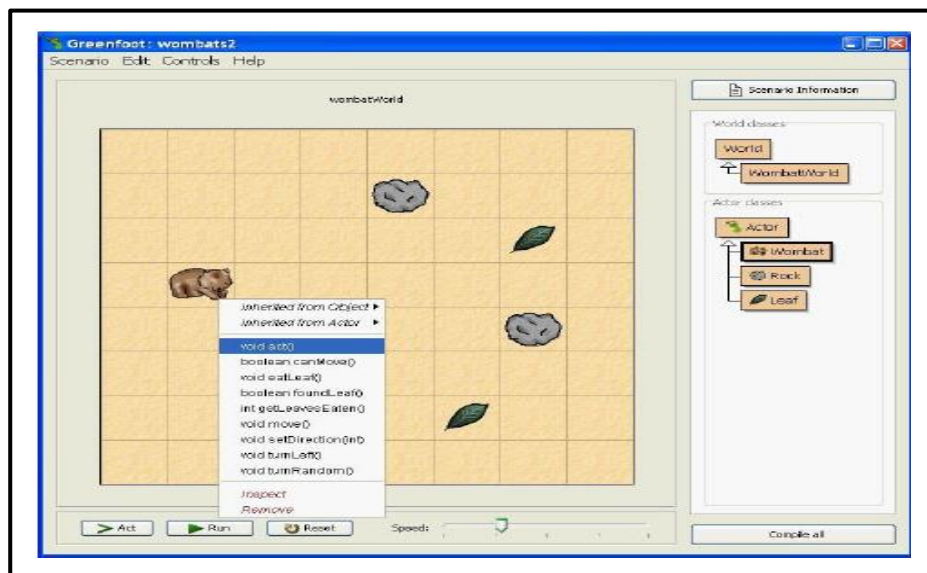


Figure 7. Invoquer des méthodes dans BlueJ (Yan Lu, 2009).

PrOgO, un jeu virtuel 3D par lequel les concepts abstraits de POO sont représentés de manière significative et visible, et qui permet aux élèves de créer des objets, de visualiser leur état et leur comportement. Les représentations de PrOgO sont très précises et l'action de chaque élève peut être interprétée dans une instruction C++. Ce qui constituera un programme C++ complet, que l'étudiant pourra le visualiser et le modifier afin de pratiquer la programmation. PrOgO est le fruit de travail de l'équipe **Djelil et al. (2015)**, qui l'ont expérimenté sur les étudiants de première année d'imagerie numérique de l'Institut de Technologie du Puy-en-Velay (Université d'Auvergne, France).

Le travail de recherche de **Seng Yoke et al. (2018)** présente une nouvelle forme de jeux d'apprentissage de la POO, où ces chercheurs ont développé Odyssey of Phoenix (OoP) qui est un jeu mobile 2D (**Figure 8**). OoP est un jeu de rôle destiné aux étudiants du premier cycle, lesquels résolvent le jeu étape par étape, ce qui leur permet d'acquérir des connaissances sur le paradigme POO à la fin de chaque tâche. Un prototype de « OoP » était expérimenté sur vingt étudiants de la première année Informatique de KDU (collège universitaire privé en Malaisie).



Figure 8. Interface OoP (Seng Yoke *et al.*, 2018).

4.4. Synthèse

De tout ce qui précède, il est bien montré que la recherche dans l'axe d'amélioration et de facilitation d'apprentissage d'Algorithmique, la Programmation et la Programmation Orientée Objet (POO) est une vieille préoccupation (depuis 1993) chez plusieurs chercheurs. En effet, chacun a présenté cette solution avec une approche et un environnement particulier, commençant par l'approche « Basé sur l'Exemple » de **Cypher et Halbert (1993)** concrétisée par l'environnement visuel de programmation procédurale Pygmalion. Cette approche qui était abordée aussi par **Benabbou et Hanoune (2006)** avec l'intégration de l'Intelligence Artificielle (IA) dans un environnement d'apprentissage de l'Algorithmique EasyAlgo, l'a été également dans le récent travail de **Guibert *et al.* (2006)** pour l'apprentissage de la programmation procédurale avec l'environnement MELBA.

Dans le même sens, **Ryan Bubinski et Zach Sims** ont préféré d'exploiter la technologie d'Internet et du Web en créant la plateforme en ligne **Codecademy en 2011**, pour l'apprentissage de plusieurs langages de programmation tels que Python, PHP, JavaScript, HTML et CSS.

Pour l'apprentissage de la Programmation Orientée Objet, la première approche était basée sur la visualisation et la modélisation graphique UML, qui était abordée par **IBM en 1991** dans l'environnement Rational Rose ; poursuivie ensuite dans BlueJ de **Kölling (1999)**.

L'approche de visualisation était aussi une inspiration pour l'approche de simulation des programmes orientés objet, utilisée par **Esteves et Mendes (2004)** dans **OOP-Anim**.

Par ailleurs, WEBLOOP de **Cheung Ronnie (2006)** a gardé le principe Rational Rose et BlueJ de génération de diagramme de classes UML à partir d'un code source ou la génération inverse (code à partir d'un diagramme), mais en utilisant un environnement web ni installé ni configuré, et accédé de n'importe où via le navigateur.

Toujours dans le cadre d'apprentissage de la POO, la dernière approche abordée est l'apprentissage par les jeux. Cette approche était utilisée dans Greenfoot de **Yan Lu (2009)**, dans PrOgO de **Djelil et al. (2015)** qui ont changé le langage de programmation Java par le C++ ; et enfin dans Odyssey of Phoenix (OoP) de **Seng Yoke et al. (2018)** qui ont passé vers les jeux mobiles au lieu des jeux ordinateurs.

D'une manière générale, les grandes différences observées entre tous les travaux connexes précédents, sont représentées dans le **Tableau 1**.

Compte tenu de ce qui précède, et malgré toutes les différences enregistrées entre les approches et les types d'environnements d'apprentissage développés dans les travaux précédents, on remarque qu'ils partagent tous, le choix du style d'Apprentissage individuel, et ignorent l'exploitation d'un concept pédagogique très intéressant, qui est l'Apprentissage Collaboratif. Ce dernier a été mentionné comme une base de perspective de recherche principale dans les travaux **d'Esteves et Mendes (2004)** et **Cheung Ronnie (2006)**, où les premiers auteurs ont mentionnés dans leur conclusion que leur environnement OOP-Anim sera également utilisé dans le cadre d'un projet plus vaste qui vise à explorer les aspects sociaux dans l'apprentissage de la programmation. Ceci, afin de créer un environnement qui facilite la collaboration des élèves et la supervision des enseignants dans les activités de programmation.

En effet, selon **Cheung Ronnie (2006)**, l'utilisation des environnements collaboratifs pour le développement orienté objet sera une bonne future étude.

Auteurs	Module à Apprendre	Langage de Programmation Etudié	Approche Abordée	Système Réalisé	Mode de Système	Évalué
Cypher et Halbert (1993)	Programmation Procédurale	/	Visuelle et Basé sur Exemple	Environnement Pygmalion	Hors Ligne	Non
Benabbou et Hanoune (2006)	Algorithmique	/	Basé sur Exemple et IA Recommandation	Environnement EasyAlgo	Hors Ligne	Non
Guibert et al. (2006)	Programmation Procédurale	/	Basé sur Exemple	Environnement MELBA	Hors Ligne	Oui
Ryan Bubinski et Zach Sims (2011)	Programmation	Python, PHP, JavaScript, HTML et CSS	Apprentissage en Ligne	Plateforme Codecademy	En ligne	Non
IBM (1991)	Programmation Orientée Objet (POO)	Java, C++ et Visual Basic	Visualisation et Modélisation Graphique UML	Outil Rational Rose	Hors Ligne	Non
Kölling (1999)	Programmation Orientée Objet (POO)	Java	Visualisation et Modélisation Graphique UML	Environnement BlueJ	Hors Ligne	Non
Esteves et Mendes (2004)	Programmation Orientée Objet (POO)	Java	Visualisation et Simulation	Environnement OOP-Anim	Hors Ligne	Non
Cheung Ronnie (2006)	Programmation Orientée Objet (POO)	Java	Visualisation et Modélisation Graphique UML	Environnement WEBLOOP	En Ligne	Non
Yan Lu (2009)	Programmation Orientée Objet (POO)	Java	Apprentissage par les Jeux	Environnement Greenfoot	Hors Ligne	Non
Djelil et al. (2015)	Programmation Orientée Objet (POO)	C++	Apprentissage par les Jeux	Jeu Virtuel 3D PrOgO	Hors Ligne	Oui
Seng Yoke et al. (2018)	Programmation Orientée Objet (POO)	/	Apprentissage par les Jeux	Jeu Mobile 2D Odyssey of Phoenix OoP	Hors Ligne	Oui

Tableau 1. Etat de comparaison entre les différents environnements et systèmes d'apprentissage

5. Conclusion

Dans ce chapitre, nous avons présenté le concept de l'apprentissage en ligne (*e-learning*), ses principes, et les outils nécessaires au fonctionnement de cette forme d'apprentissage. Ainsi, nous avons montré une forme très importante de ce concept, qui est la forme collaborative connue sous le nom de l'apprentissage collaboratif assisté par ordinateur, en présentant les différents avantages de cette forme d'apprentissage, ainsi que les obstacles de l'application de cette dernière.

Ensuite, nous avons passé à l'application de l'apprentissage à distance pour l'apprentissage de l'algorithmique, la programmation informatique en général, et la **Programmation Orientée Objet (POO)** en particulier. Pour cette dernière, nous avons extrait les différentes difficultés de son apprentissage, à partir de la présentation d'un ensemble de travaux de recherche traitant ce sujet, ainsi que les approches et les environnements d'apprentissage à distance développés dans ce cadre. Comme résultats, nous avons remarqué une absence de l'exploitation du concept de l'apprentissage collaboratif au sein des approches et environnements d'apprentissage favorisant ce paradigme de programmation. Ce qui nous a motivés à proposer une nouvelle méthode d'apprentissage collaboratif à distance, qui aide les apprenants à améliorer leurs niveaux de connaissances dans la Programmation Orientée Objet. La méthode d'apprentissage collaboratif à distance proposée, ainsi que le système qui la concrétise seront présentés dans le chapitre suivant.

Chapitre II. Conception

Chapitre II. Conception d'un système d'apprentissage collaboratif pour l'apprentissage de la POO

1. Introduction

Durant ces derniers temps et surtout la situation sanitaire qu'a connu le monde entier à cause de CoronaVirus, l'enseignement à distance était la seule solution existante pour continuer le processus d'enseignement. En effet, ce mode d'enseignement est devenu très populaire.

Le contenu pédagogique à enseigner dans cette nouvelle forme d'apprentissage (e-learning) était bien différencié d'un enseignant à un autre, tout en dépendant des besoins des apprenants, ainsi que les difficultés rencontrées dans un module particulier. Tout comme le contenu pédagogique, le niveau et l'âge des apprenants étaient aussi variés à partir du cycle primaire, jusqu'au cycle universitaire avec ses différentes filières (Mathématiques, Langues Etrangères, Informatique...).

L'Informatique avec ses différents modules pédagogiques, était l'une des disciplines les plus ciblées par l'apprentissage en ligne. Ceci, étant donné la capacité de ses apprenants à accepter et à bien utiliser facilement les TIC, dans cette nouvelle forme d'apprentissage. C'est en effet que dans ce cadre, divers approches et systèmes e-learning étaient développés pour différents modules informatiques, tels que la **Programmation Orientée Objet (POO)**.

Ainsi, dans ce chapitre, nous présentons notre nouvelle méthode d'apprentissage à distance, avec son principe, ses étapes et essentiellement l'exploitation utile du concept d'Apprentissage Collaboratif. Ensuite, nous allons projeter cette dernière sur l'apprentissage d'un module pédagogique particulier, dans notre cas la POO, avec une conception bien détaillée d'un système adoptant cette nouvelle méthode.

2. Problématique et objectifs de la recherche

2.1. Problématique de la recherche

Dans le cadre de l'utilisation des Technologies d'Information et de Communication (TIC) dans l'apprentissage universitaire, plus particulièrement dans la filière Informatique, nombreux systèmes d'apprentissage en ligne étaient créés dans cet axe. Ceci, pour différentes

matières (cours) telles que : la Logique, l'Algorithmique et la Programmation Informatique par ses différents paradigmes, y compris la **Programmation Orientée Objet (POO)**.

La modélisation objet est la première phase de ce paradigme de programmation. Elle consiste à créer un modèle du système informatique à réaliser, lequel permet de représenter et de qualifier les objets du monde réel. Ceci, sous forme de types ou de classes sous l'angle de la compréhension des besoins et indépendamment de la manière dont ces classes seront mises en œuvre. C'est ce que l'on appelle, l'**Analyse Orientée Objet (AOO)**. Ces éléments sont alors enrichis et adaptés, pour représenter les éléments de la solution technique nécessaire à la réalisation du système informatique. C'est ce que l'on appelle **la Conception Orientée Objet (COO)**. Dans ce sens, à un modèle d'analyse, peuvent correspondre plusieurs modèles de conception. En effet, l'analyse et la conception étant fortement interdépendantes, et on parle alors **d'Analyse et de Conception Orientée Objet (ACOO)**. Une fois un modèle de conception établi, il est possible aux développeurs de lui donner corps dans un langage de programmation particulier. C'est ce que l'on appelle **la Programmation Orientée Objet (POO)** ou **OOP** (en anglais **Object-Oriented Programming**) (**Href 8**). La figure ci-dessous (**Figure 9**) résume les phases de ce paradigme.

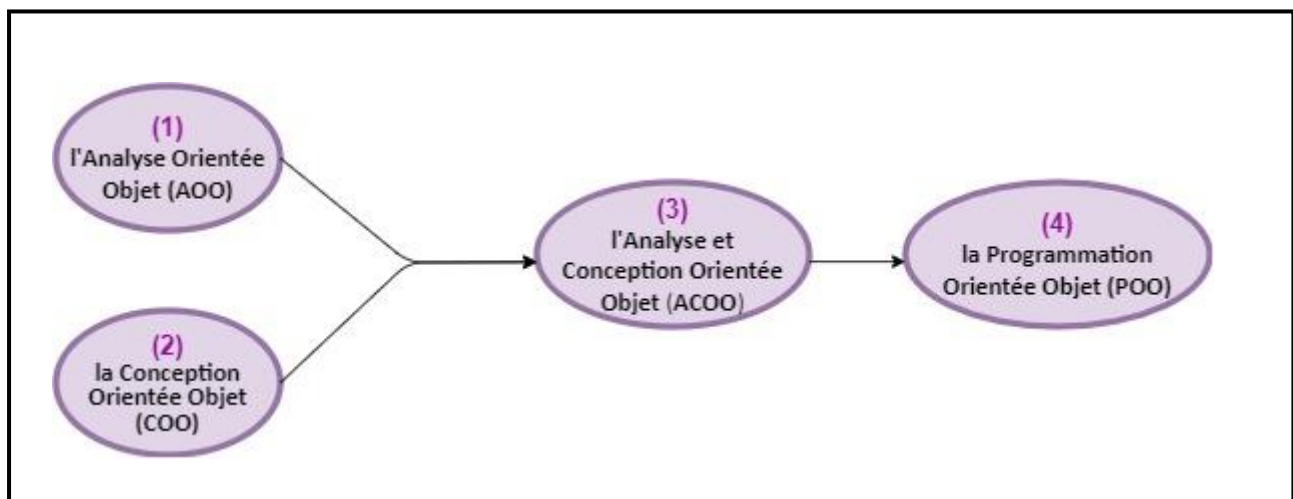


Figure 9. Les principales phases du paradigme de POO

Ce paradigme (l'**OO**) a été adopté comme un module à étudier dans le programme pédagogique des étudiants universitaires en Informatique; et comme plusieurs autres modules, les étudiants au fil des ans ont rencontré des difficultés dans l'apprentissage de ce dernier. En effet, de nombreux chercheurs ont bien mentionné ces difficultés dans leurs apprentissage

(voir chapitre I), où certains voient que le problème est au niveau du changement de la vision de modélisation des problèmes de la procédurale habituelle aux étudiants, vers une autre orientée objet. D'autres chercheurs voient que ce n'est pas l'orienté objet qui est la cause des problèmes, mais c'est les outils (langages et environnements) disponibles pour l'enseigner. Ainsi, l'un des difficultés les plus mentionnées est comment arriver à une bonne conception orientée objet d'un problème posé.

Donc, notre souci dans cette situation est d'arriver à montrer comment exploiter les **TIC** dans l'apprentissage en général, et dans celui du module de la **Programmation Orientée Objet (POO)** en particulier. Et comment arriver à assurer une collaboration utile à cet apprentissage? En effet, ces deux questions constituent la problématique principale de ce travail de recherche.

2.2. Objectifs du travail de recherche

De nombreux travaux d'investigation ont montré que les nouveaux programmeurs ainsi que les étudiants universitaires en Informatique ont affronté beaucoup de difficultés au niveau de l'apprentissage du paradigme de la programmation Orientée Objet. Ceci est dû à plusieurs raisons techniques et pédagogiques, mentionnées et différenciées d'un travail à un autre.

Le but de ce projet est de proposer une nouvelle méthode d'apprentissage collaboratif à distance, adéquate à l'apprentissage du module de la POO, ainsi que d'autres modules ayant une structure pédagogique similaire. Ensuite, de mettre au point un système d'apprentissage à distance capable de concrétiser le principe de cette méthode.

Après avoir effectué une étude bibliographique sur l'apprentissage collaboratif et ses avantages, une étude sur les différentes difficultés d'apprentissage de la POO chez les nouveaux programmeurs et les étudiants universitaires, ainsi que les systèmes développés dans le cadre de l'amélioration du niveau d'apprentissage de ce module (voir chapitre I), nous comptons sur le plan suivant pour atteindre notre objectif principal :

1. Effectuer une étude de cas sur les étudiants universitaires en Informatique, afin d'extraire les vraies difficultés rencontrées dans l'apprentissage de la POO.
2. Proposition d'une nouvelle approche d'apprentissage collaboratif à distance qui permet de faciliter l'apprentissage en général, et l'apprentissage de la POO en particulier.

3. Conception et implémentation d'un système adoptant cette nouvelle méthode, qui doit offrir une interface simple, et qu'utilise un outil d'apprentissage collaboratif.
4. Tester le prototype développé avec les étudiants du département d'informatique.

Donc, l'objectif principal de ce travail, est d'arriver à concevoir et implémenter une nouvelle approche d'apprentissage collaboratif à distance, qui facilite le processus d'apprentissage du module de **POO** chez les étudiants universitaires (**Benyounes et al., 2020**).

3. Etude de cas sur l'apprentissage de la Programmation Orientée Objet

Dans le but d'extraire les problèmes réels rencontrés par les étudiants d'Informatique lors de l'apprentissage de la Programmation Orientée Objet (POO), nous avons établi un questionnaire (en ligne et manuscrit, voir **Annexe 2**) comme une méthode de collection de données.

Lors de l'établissement de ce questionnaire, nous avons adopté les principes de l'échelle de **Likert** (voir **Annexe 1**) dans la majorité de nos questions, en mettant des réponses précises et bien nuancées afin d'obtenir des réponses aussi fiables que possible (**Benyounes et al., 2020**).

Nous présentons dans cette section, les résultats de réponses de quelques questions intéressantes posées dans ce questionnaire.

Question 9. Est-ce que vous avez rencontré des difficultés avec le module de Programmation Orientée Objet (POO) ?

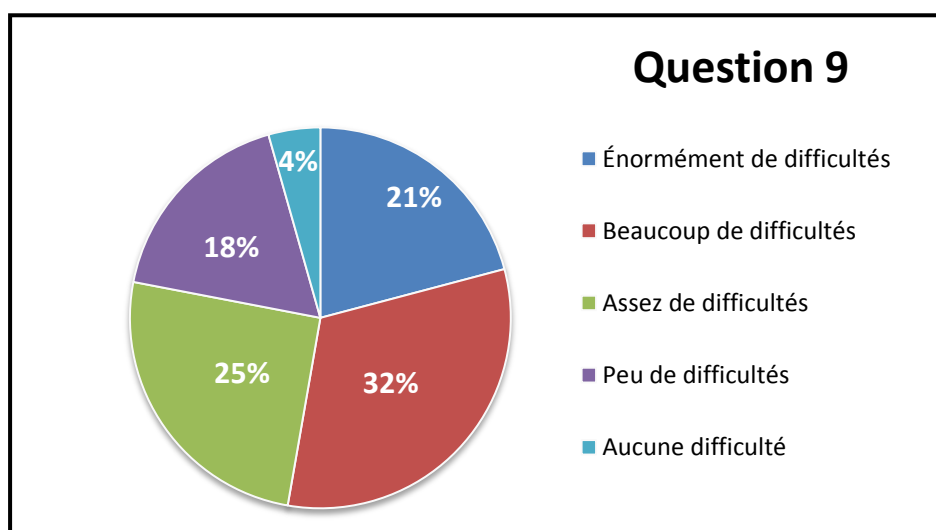


Figure 10. Résultats des réponses à la question 9

Question 13. Que pensez-vous du rythme de l'enseignant (s) dans ce module (POO) ?

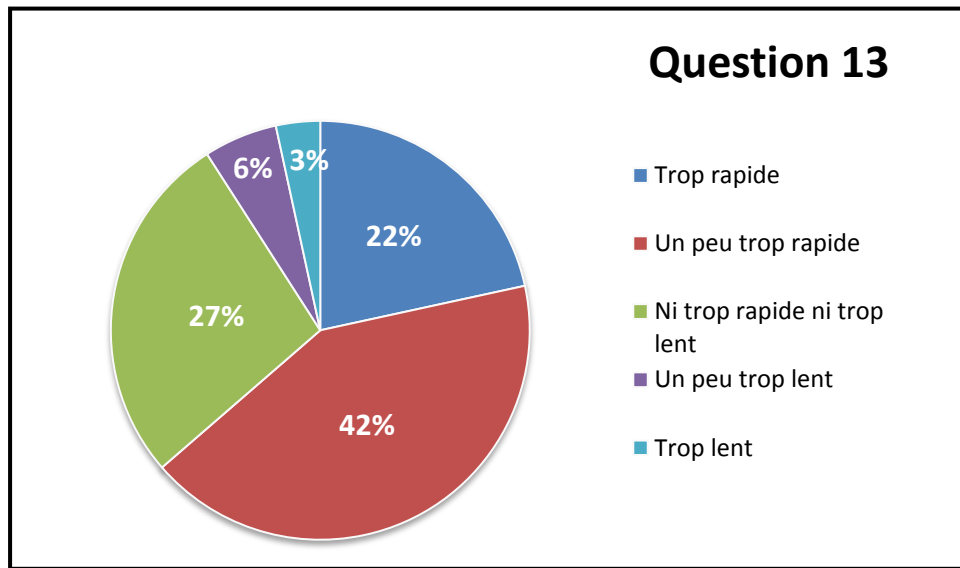


Figure 11. Résultats des réponses à la question 13

Question 16. Selon vous, la difficulté principale dans ce module (POO), était dans :

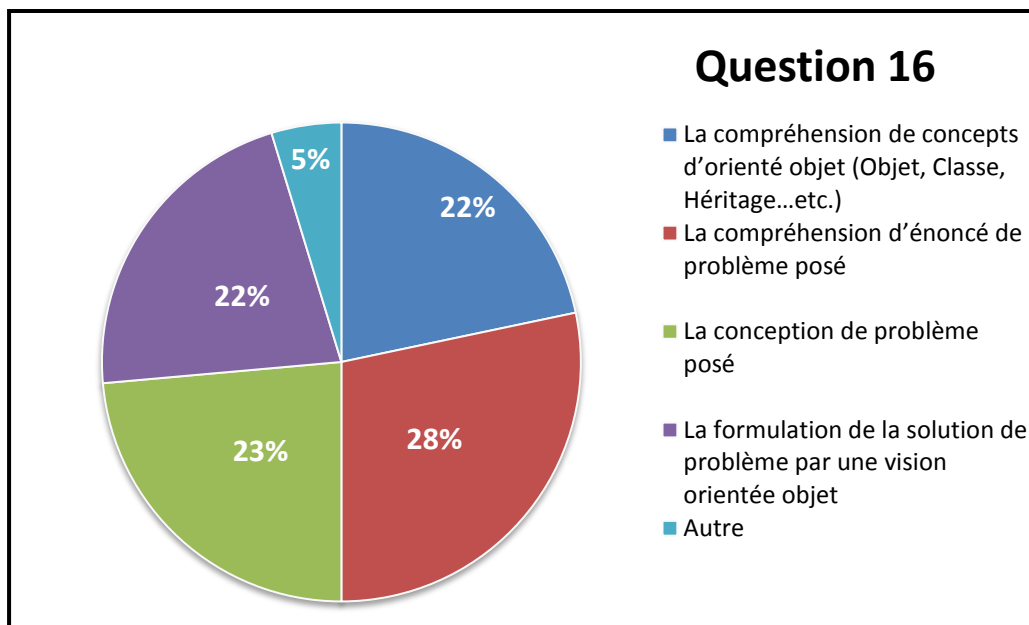


Figure 12. Résultats des réponses à la question 16

Question 20. Que pensez-vous de l'apprentissage collaboratif (Travailler en groupe) dans le module de POO ?

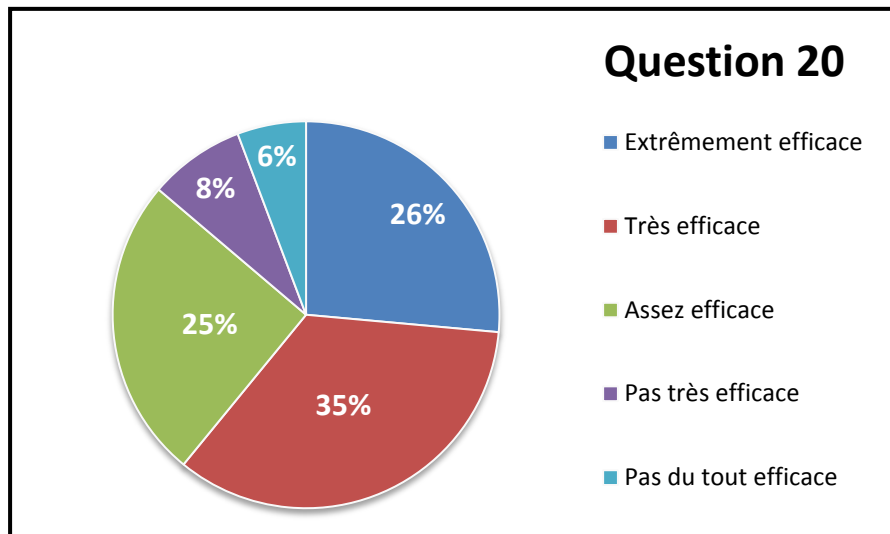


Figure 13. Résultats des réponses à la question 20

Question 24. Que pensez-vous des plateformes d'apprentissage en ligne de la POO ?

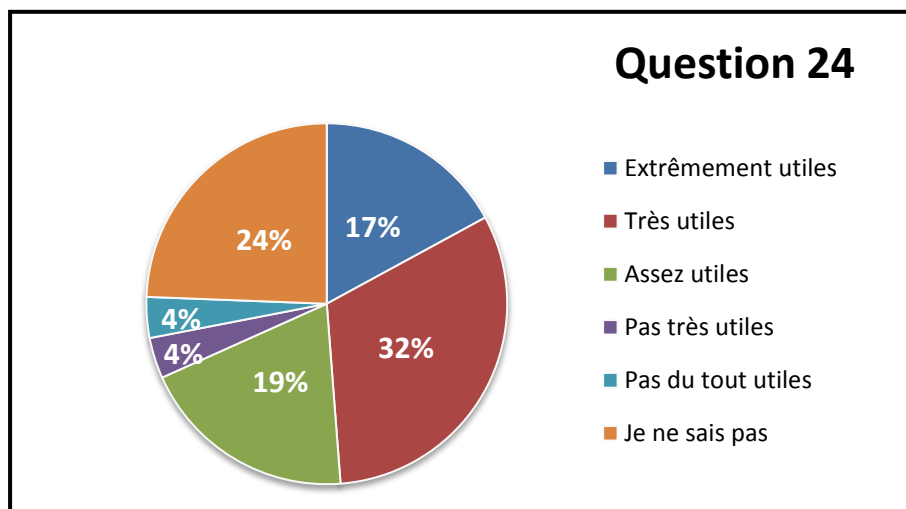


Figure 14. Résultats des réponses à la question 24

3.1. Interprétation des statistiques obtenues

En analysant les résultats obtenus de ce questionnaire chez 91 étudiants universitaires en Informatique (**Benyounes et al., 2020**) nous pouvons constater les points suivants :

- En général, la majorité des étudiants ont la difficulté d'apprendre les modules qui demandent des nouveaux paradigmes et langages de programmation.
- En particulier, la majorité des étudiants ont la difficulté d'apprendre le paradigme de la Programmation Orientée Objet (POO).
- 68% des étudiants n'arrivent pas à comprendre les explications données dans le module de POO.
- 41% des étudiants, pensent qu'il y'a un problème dans le choix des outils d'enseignement du module de POO.
- Concernant le rythme d'enseignement du module de POO, 64% des étudiants confirment qu'il est rapide par rapport à eux.
- Pour le problème majeur dans l'apprentissage de la POO, 51% des étudiants voient que le problème n'est pas d'apprendre les concepts de base de la POO, mais c'est la compréhension d'énoncés des problèmes posés dans ce module, ainsi que la manière d'arriver à une conception orientée objet à ces problèmes.
- A propos de l'Apprentissage Collaboratif (travail en groupe), la plupart des étudiants (61%) pensent qu'il est efficace dans le module de POO. En effet, 49% ont vu que la bonne méthode de regroupement dans cet apprentissage est le regroupement décidé par l'enseignant.
- Concernant une question sur les plateformes d'apprentissage en ligne de la POO, 49% des étudiants pensent qu'elles sont utiles, où ils ont insisté sur la collaboration entre les apprenants, et la simplicité des outils d'apprentissage comme deux fonctionnalités principales à aborder dans ces plateformes.

4. Description de l'approche LISA : Learn, Imagine, Share and Approve

La Programmation Orientée Objet (POO) est un module pédagogique très intéressant et indispensable dans le cursus universitaire de l'étudiant d'Informatique. C'est l'un des paradigmes de programmation les plus utilisés, et demandés dans les géants de développement technologique.

Malheureusement, l'apprentissage des concepts et principes de cette approche de programmation n'est pas toujours facile. C'est ce qui a attiré les chercheurs à penser d'une approche, méthode et d'outil d'amélioration de niveau d'apprentissage de ce paradigme.

Dans ce sens, plusieurs approches étaient concrétisées dans différents systèmes informatiques pour la résolution de divers problèmes et obstacles d'apprentissage de la POO (voir chapitre I). A nos connaissances, tous ces travaux avec leurs approches, n'ont pas abordé le concept pédagogique de l'Apprentissage Collaboratif. C'est ce qui nous a motivé, à penser d'une nouvelle approche d'apprentissage en général, et de la POO en particulier. En effet, cette dernière est une méthode adéquate pour les étudiants universitaires. Elle se base sur l'enseignement de l'Analyse et Conception Orientée Objet (ACOO) laquelle est le cœur de la POO au lieu d'enseigner l'implémentation orientée objet sur un langage de programmation particulier, tout en exploitant le principe de l'Apprentissage Collaboratif. Cette nouvelle approche, appelée LISA (Benyounes *et al.*, 2020) où :

- **L (Learn)** : c'est d'apprendre en collaboration avec les co-apprenants, les concepts de base du module étudié (dans notre cas la POO), grâce à un contenu pédagogique sous forme de chapitres composés de concepts, dans le cadre d'un Apprentissage Collaboratif.
- **I (Imagine)** : c'est de maîtriser l'utilisation de l'imagination dans la réalisation d'une tâche particulière dans le module étudié. Dans le cas de la POO, cette étape sera effectuée lors de la phase de l'Analyse et Conception. Elle consiste à utiliser l'imagination dans l'extraction des objets constituant un problème posé et de déterminer les relations entre eux, afin de les représenter par les bons concepts orientés objet.
- **S (Share)** : une étape de concevoir de la résolution personnelle d'un problème posé et de la partager par la suite avec les membres de groupe (les co-apprenants).

- **A (Approve)** : c'est de sélectionner et d'approuver la meilleure solution du problème posé parmi toutes les solutions proposées par les membres de groupe (les co-apprenants).

La figure ci-dessous (**Figure 15**) résume les différentes étapes de l'approche proposée. Ces dernières, seront appliquées lors de l'apprentissage de tout le module pédagogique concerné.

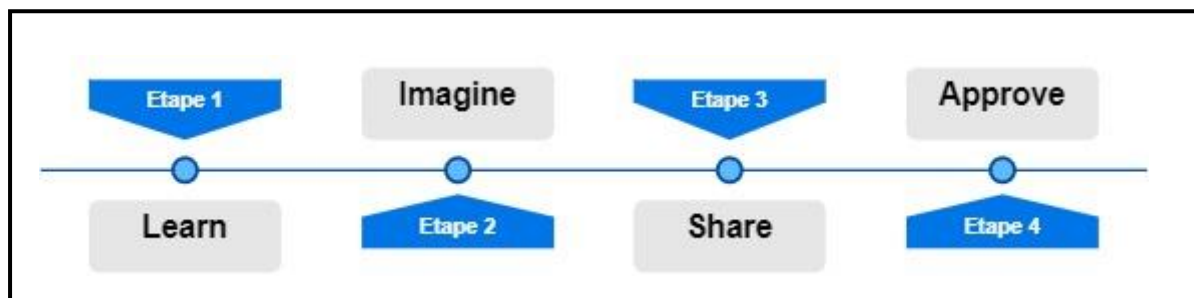


Figure 15. Principe de l'approche LISA

Ainsi, pour que cette nouvelle méthode soit bien adéquate à l'apprentissage d'autres modules, ces derniers doivent avoir la structure pédagogique ci-après détaillée.

4.1. La structure pédagogique du module à étudier

Le module à étudier par cette nouvelle méthode **LISA**, doit être organisé en deux phases principales. La première est une phase d'apprentissage des notions de base du module, laquelle est organisée sous forme de **Chapitres** constitués d'un ensemble de **Concepts**. La deuxième, est une phase de pratique et d'exploitation des connaissances apprises dans la première phase, laquelle sera concrétisée sous forme de **Projets finaux**.

La figure ci-dessous (**Figure 16**) représente la structure pédagogique du module étudié par la nouvelle méthode **LISA**.

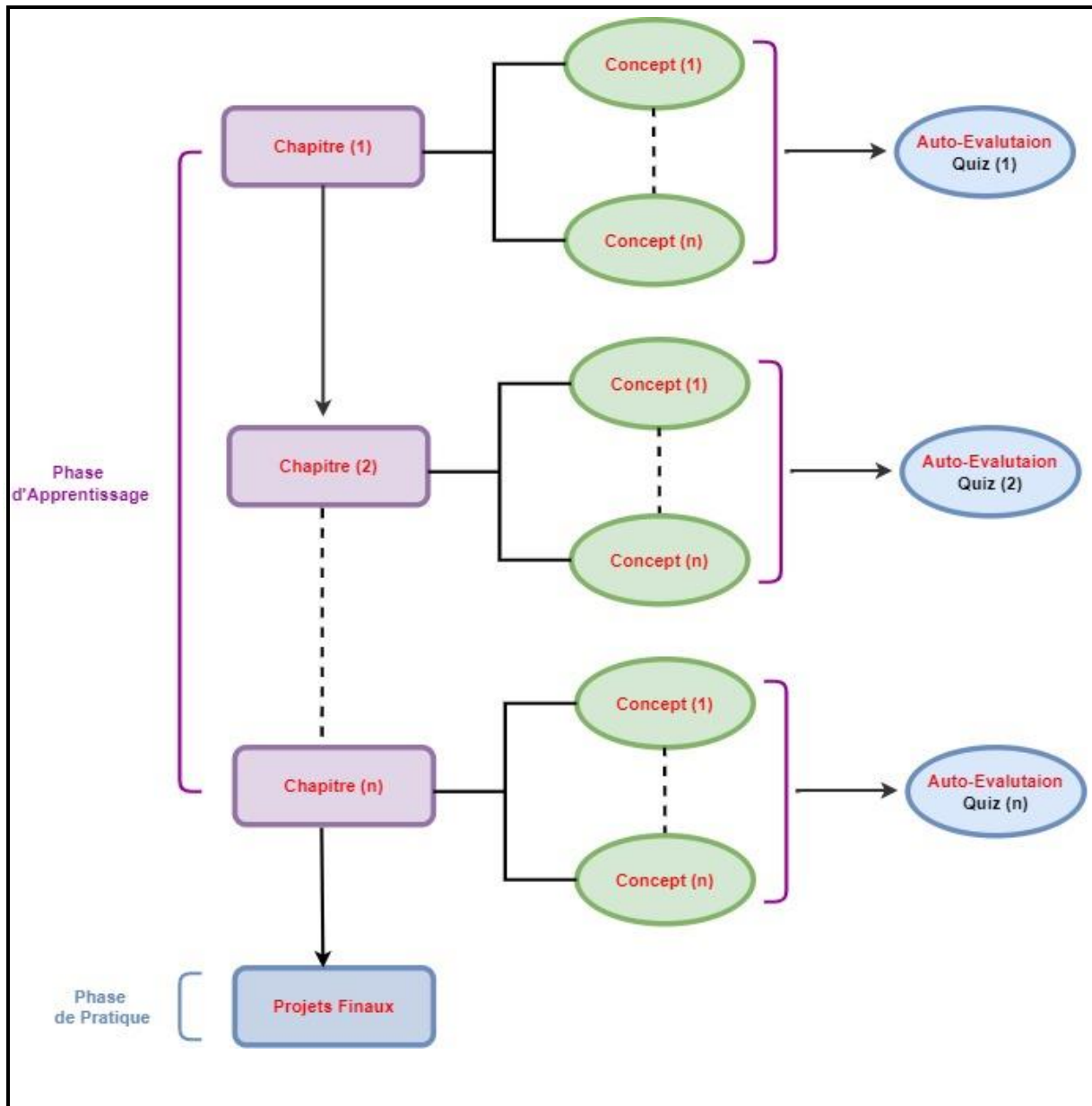


Figure 16. La structure pédagogique du module étudié par LISA

4.2. L : Learn (Apprendre)

Cette première étape, est généralement commune et indispensable dans la majorité des modules éducatifs. Elle repose sur le principe d'apprentissage des concepts de base, lois et règles de la discipline ; et placée à la première phase pédagogique (phase d'apprentissage) du module à étudier, dans notre cas la POO.

Donc, en projetant le principe de l'étape **Learn** sur le processus d'apprentissage du module de la POO, l'apprenant commence à apprendre les concepts de base (classes, objet,

héritage...) de la POO à l'aide d'un contenu écrit (chapitres) bien préparé et géré par l'enseignant. Ce dernier qui joue le rôle du concepteur de la leçon, doit répondre aux critères suivants : la fiabilité, la simplicité ainsi qu'un rythme de difficulté et d'avancement convenable aux capacités d'apprentissage des apprenants.

- Le premier chapitre (1) est une introduction à la POO, définition et avantages de ce paradigme de programmation, afin de répondre aux questions fréquemment posées dans l'esprit de chaque apprenant, qui sont : Qu'est-ce que la POO ? Et pour quoi on apprenne ce module ?
- Le deuxième chapitre (2) présente les notions de base de la POO, les Classes et les Objets, les définitions de ces concepts, leurs rôles, les relations entre eux et quand et comment on utilise ces deux concepts ?
- Le troisième chapitre (3) présente les concepts un peu complexes de la POO, en commençant par l'Encapsulation et la Surchage, jusqu'à arriver aux notions d'Héritage et Polymorphisme. Ce chapitre est dans le but de bien comprendre le rôle de ces concepts dans la création des systèmes sécurisés, réutilisables et faciles à maintenir, qui sont les grands avantages de ce paradigme de programmation.
- Le quatrième chapitre (4) traite les notions des Classes Abstraites et les Interfaces, les définitions et leurs intérêts pratiques.
- Le dernier chapitre (5), consiste à introduire la modélisation UML, en concentrant sur la notion des diagrammes des classes qui est la base de la modélisation orientée objet.

Chaque chapitre parmi ces cinq, peut être composé d'un ensemble de **concepts** selon la vision de l'enseignant modérateur du cours.

Dans ces chapitres, l'apprenant peut donc collaborer avec ses co-apprenants dans la compréhension de ces nouveaux concepts et règles orientées objet, avec toujours la possibilité d'évaluer son niveau d'apprentissage en passant des **Quiz** dans le cadre d'une étape d'auto-évaluation.

Donc, ces cinq chapitres avec leurs phases d'auto-évaluations, forment la phase d'apprentissage du module de la POO, et avec la phase de la pratique, on obtient la structure générale de ce module, qui est bien adéquate à notre méthode **LISA**. Cette structure est résumée dans la figure ci-dessous (**Figure 17**).

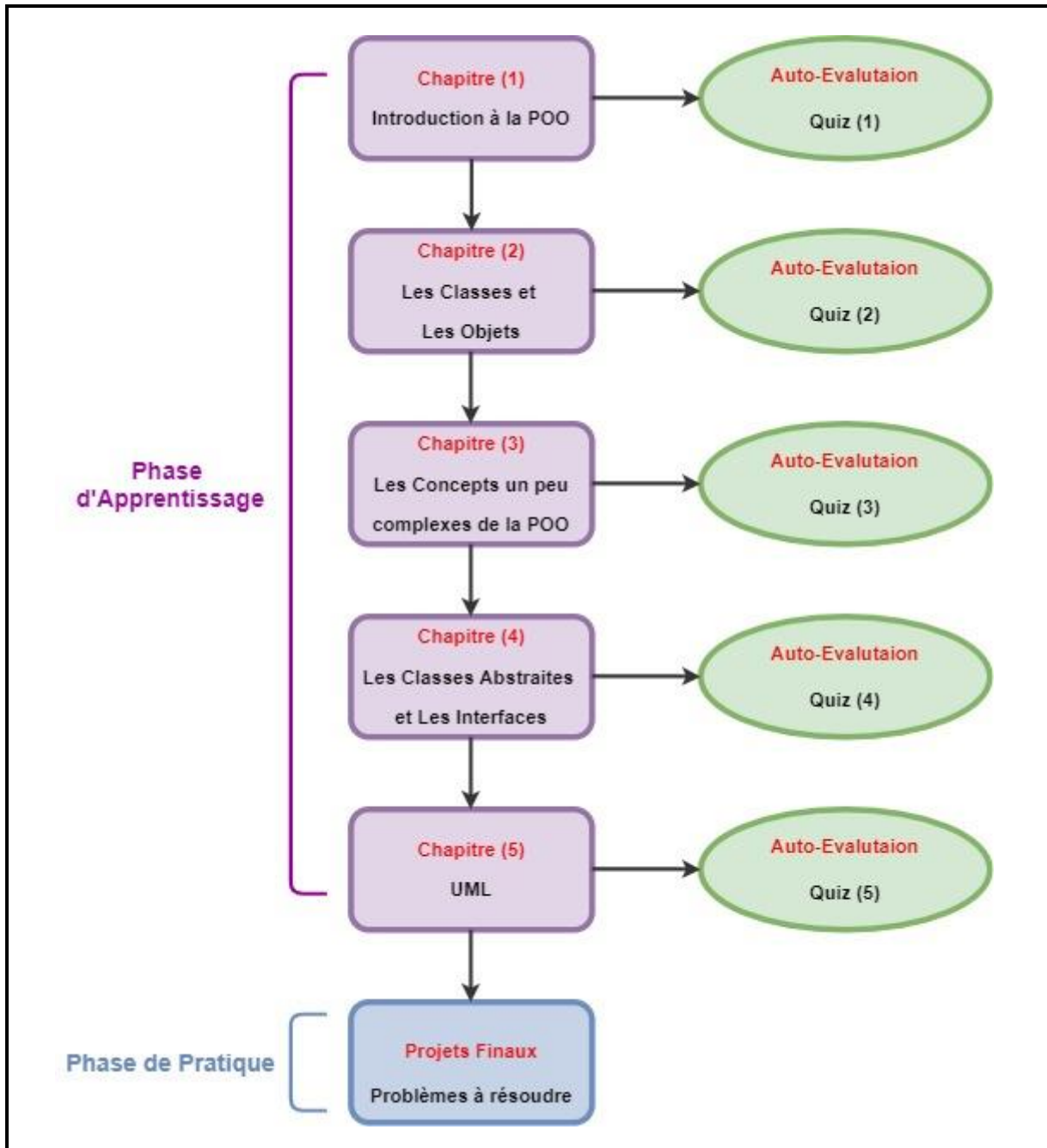


Figure 17. La structure pédagogique du module de POO

Learn nous permet d'introduire et d'exploiter le concept de l'apprentissage collaboratif. Celui-ci permet à l'apprenant de discuter avec ses co-apprenants lors de la phase d'apprentissage, dans le cadre d'échange de connaissances et de correction de fausses compréhensions. Cet apprentissage collaboratif peut être différencié d'un système à un autre selon la vision du créateur et les besoins pédagogiques.

Dans notre système, l'apprentissage collaboratif abordé est bien présenté dans la section suivante.

4.2.1. Description de l'Apprentissage Collaboratif appliqué

Dans cet apprentissage collaboratif, l'apprenant sera affecté à son groupe d'apprentissage, selon la méthode de regroupement **le premier arrivé le premier servi (affecté à un groupe)**.

4.2.1.1. La Méthode de Regroupement

Etant donnée l'absence d'informations utiles (préférences, mentalités, capacités sociales...) sur les apprenants dans notre système, ainsi que l'absence d'une pré-connaissance sur leurs vrais niveaux dans la POO, qui est généralement le cas réel de l'enseignant de n'importe quel module pédagogique avec ses étudiants, on a choisi d'appliquer une méthode de regroupement aléatoire, basée seulement sur le temps d'arrivée (d'inscription au cours) de l'apprenant. Aussi, le marché du travail impose souvent aux nouveaux diplômés de travailler à côté de nouvelles personnes, avec des mentalités totalement différentes et inattendues. C'est ce qui nous a inspiré à compter sur la caractéristique de l'aléatoire dans le regroupement de nos apprenants, pour les habituer à adapter dans toutes les conditions de travail et d'accepter l'autre avis, afin de les bien préparer à la vie professionnelle.

Donc, notre méthode de regroupement est caractérisée par :

- Un regroupement aléatoire, basé sur le temps d'arrivée (d'inscription au cours) d'un nouvel apprenant.
- Comme le regroupement est aléatoire, il y'a une possibilité d'avoir des groupes mixtes (des membres masculins et féminins).
- La taille des groupes est de quatre (4) membres, lequel choix a été renforcé par un ensemble de travaux qui ont confirmé que 4, est la taille idéale d'un groupe d'apprentissage.

4.3. I : Imagine (Imaginer)

Après la compréhension des concepts de base de la POO dans l'étape **L (Learn)**, on passe à la deuxième étape **I (Imagine)**. Cette dernière repose sur l'utilisation correcte de ces concepts dans la phase de l'Analyse et Conception des problèmes qui est le grand défi des apprenants de la POO.

Dans cette étape, les illustrations peuvent être de bons outils d'explication de la procédure d'arrivée à une bonne conception d'un problème posé. Ceci en présentant à l'apprenant un ensemble d'exemples illustratifs qui clarifient bien l'utilisation de l'imagination dans l'extraction des objets réels qui constituent un problème posé. Ensuite de bien imaginer les relations possibles entre eux, et enfin, l'utilisation correcte des concepts de base de la POO dans la représentation de ces objets avec leurs relations, dans un monde orienté objet.

La figure ci-dessous (**Figure 18**) résume le principe de l'étape **Imagine**.

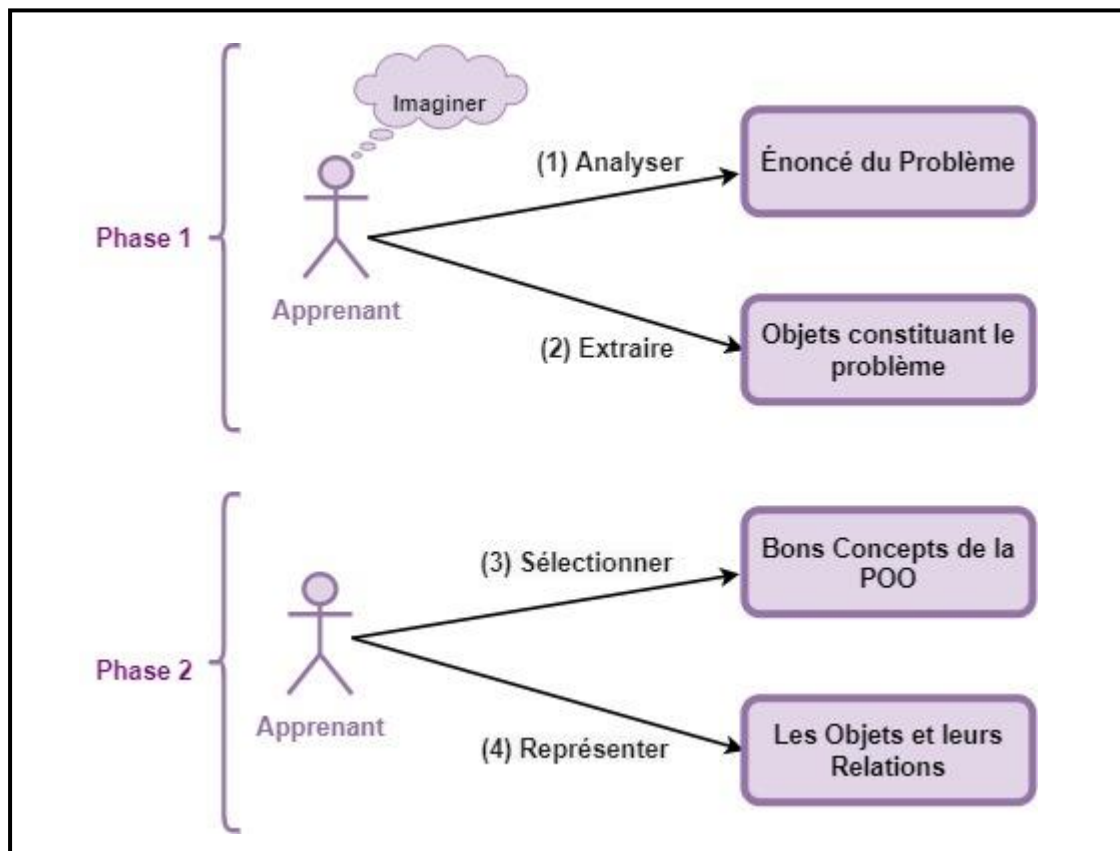


Figure 18. Principe de l'étape **I (Imagine)**

Pour la maîtrise de l'utilisation de l'imagination dans ce défi conceptuel, l'enseignant prépare un ensemble de problèmes résolus sous forme de conceptions orientées objet illustratives bien détaillées avec toutes les étapes de la résolution. Ceci en expliquant à l'apprenant comment extraire les objets cachés dans un énoncé d'un problème posé, de bien imaginer les relations entre elles, et de choisir correctement les bons concepts de la POO pour présenter ces objets.

La figure ci-dessous (**Figure 19**) présente l'un des exemples de résolution des problèmes par l'étape **Imagine**.

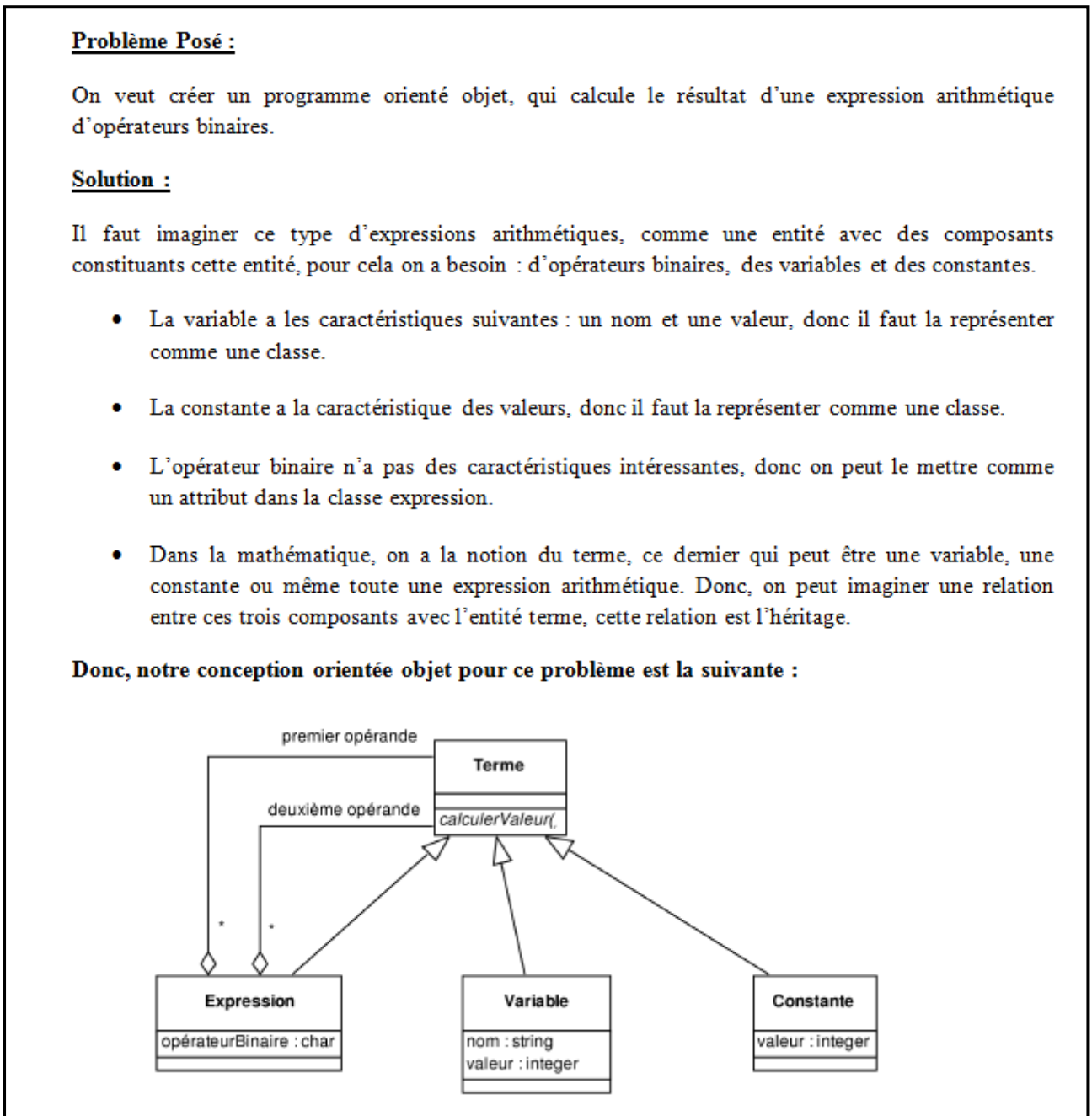


Figure 19. Exemple de résolution du problème par **Imagine**

Après avoir compris le rôle et l'utilisation de l'imagination dans la construction d'une bonne conception, l'apprenant passe à une pratique et application de cette compréhension, dans le cadre d'une opération de collaboration.

4.4. S : Share (Partager)

La première étape dans cette opération de collaboration commence par l'essai de chaque membre du groupe à appliquer ce qu'il a appris dans l'étape **I** (Imagine). Ceci, en réalisant individuellement les conceptions orientées objet d'un ensemble de problèmes posés, en utilisant un outil de conception graphique en temps réel. Ce dernier a comme but d'expliquer et de concrétiser mieux ces conceptions, et de les partager en temps réel avec ses co-apprenants. Ensuite, l'apprenant commence à expliquer et à argumenter ses choix conceptuels à ses co-apprenants dans un groupe de discussion (chat). Tout cela, dans le but de corriger les erreurs de conception de chaque co-apprenant et de partager les connaissances. La figure ci-dessous (**Figure 20**) présente le principe de l'étape **S** (Share).

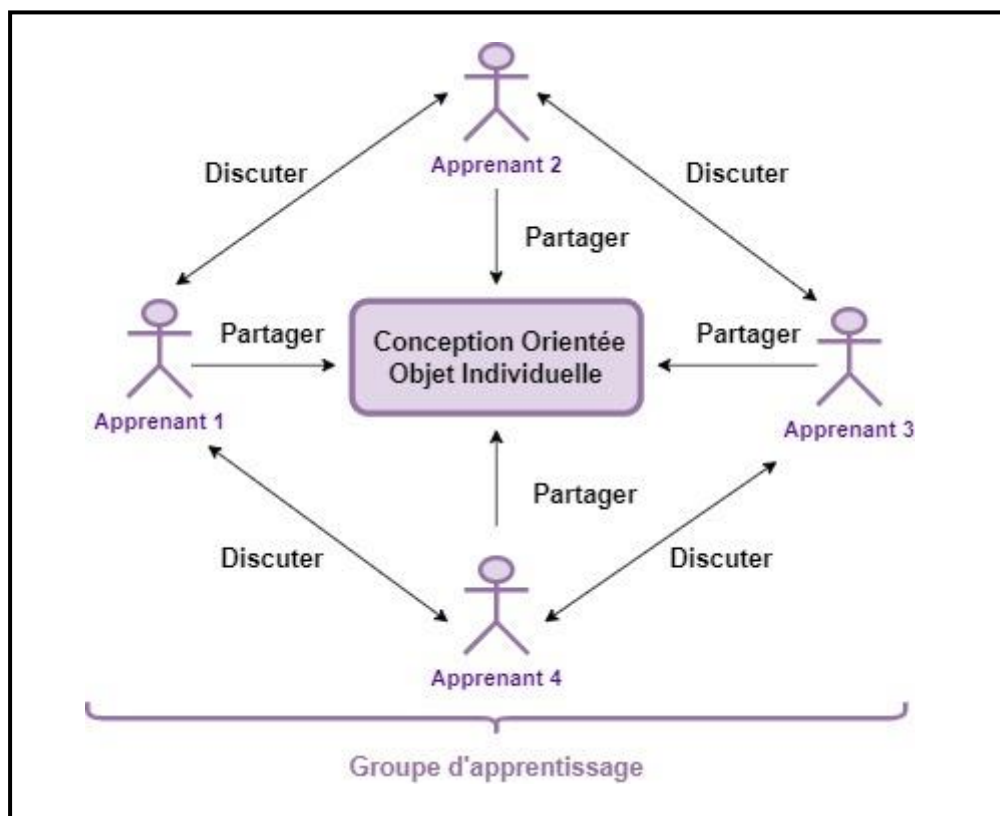


Figure 20. Principe de l'étape S (Share)

4.5. A : Approve (Approuver)

Après la réalisation individuelle de la conception orientée objet par chaque membre du groupe, c'est le temps de sélectionner la meilleure conception, laquelle sera qualifiée comme la solution finale du groupe. Pour cela, chaque membre doit noter les solutions (conceptions) de ses co-apprenants par un système de vote par étoile (star rating system), où l'opération de vote doit se dérouler selon les règles suivantes :

- l'opération de vote commence si, et seulement si, la solution est marquée comme terminée par l'apprenant propriétaire de la solution.
- L'apprenant ne peut pas noter sa solution.
- L'opération de vote doit être effectuée et terminée avant le délai de remise du projet final.
- Dans le cas où le délai de remise de projet est terminé, et les membres du groupe n'ont pas voté sur aucune solution, la meilleure solution sera déterminée par un tirage au sort entre toutes les solutions marquées comme terminées.
- La plage de la note est de 1 à 5.
- La note finale d'une solution, est égale à la somme des notes des électeurs sur le nombre des électeurs (la note finale sera égale à la note moyenne).
- Le résultat du vote est déterminé par la solution qui a la meilleure note finale (moyenne).
- Dans le cas d'une égalité entre les notes moyennes de deux solutions ou plus, la meilleure solution sera déterminée par un tirage au sort.

Enfin, Lorsque la solution du groupe est déterminée, elle sera envoyée à l'enseignant dirigeant. Ce dernier va lui attribuer une note, et va diriger ses apprenants avec des conseils et des remarques sur la solution réalisée.

La figure ci-dessous (**Figure 21**) présente le principe de l'étape A (**Approve**).

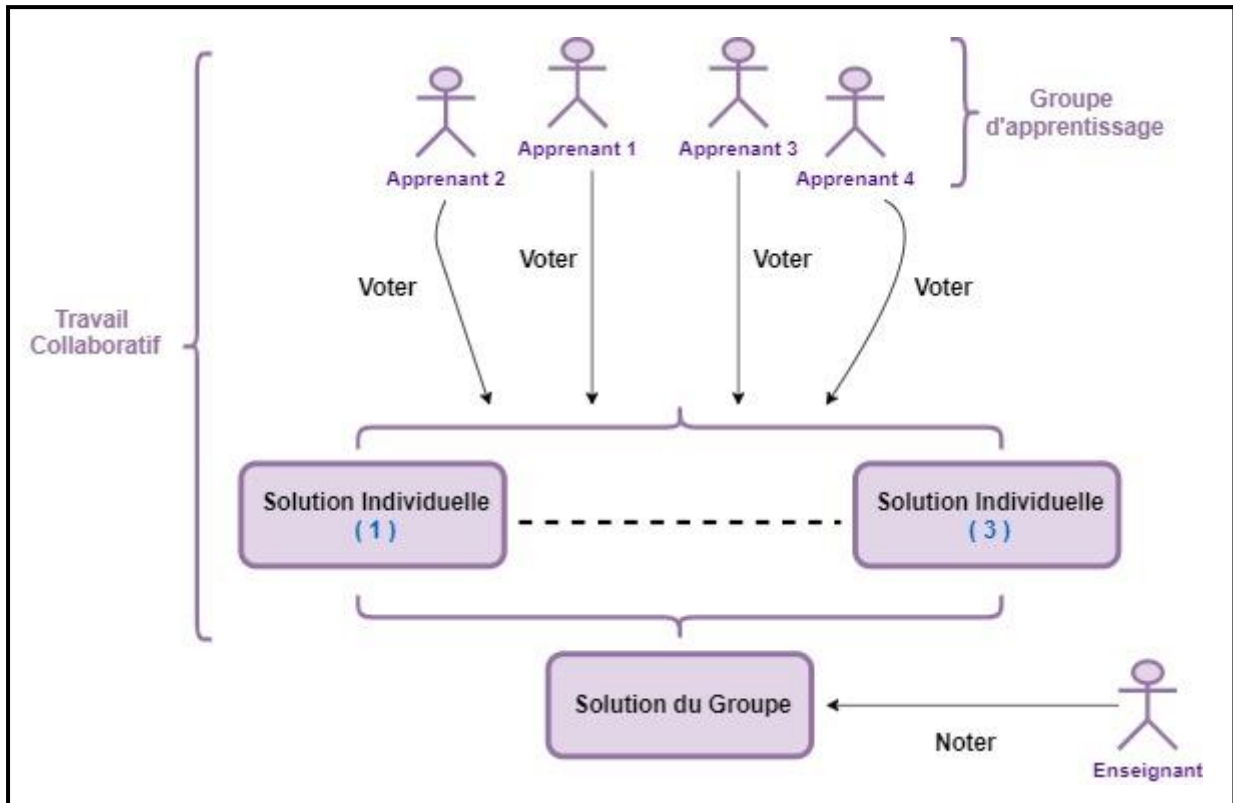


Figure 21. Principe de l'étape A (**Approve**)

5. Conception du système

Etant donné la difficulté d'intégrer notre méthode LISA dans des systèmes plateformes d'apprentissage en ligne existants, nous avons réalisé notre propre système.

Le système à concevoir est appelé **POOLISA** (une abréviation à la procédure d'apprentissage de la **POO** par notre nouvelle méthode **LISA**). Ce système est sous forme d'une plateforme d'apprentissage en ligne.

On va présenter la conception de notre plateforme d'apprentissage par l'un des diagrammes de langage de modélisation unifié (**UML**), qui est le **diagramme des cas d'utilisation (UseCase)**. Ceci, dans le but d'identifier les différents acteurs en interaction avec notre plateforme, de présenter leurs rôles, ainsi que la nature des relations entre eux.

5.1. Identification des acteurs

Dans cette plateforme d'apprentissage, on distingue trois types d'acteurs :

- Enseignant.
- Apprenant.
- Administrateur.

5.2. Identification des cas d'utilisation des acteurs

Les cas d'utilisation des acteurs de notre plateforme, sont :

• Enseignant

- Gérer le contenu pédagogique à transmettre aux apprenants.
- Proposer des exercices d'auto-évaluation et les projets pratiques à la fin de la formation (QCM ou Quiz).
- Communiquer avec les autres acteurs humains (autres enseignants et autres apprenants).
- Evaluer le travail collaboratif (projet final) demandé aux groupes des apprenants.

• Etudiant

- Consulter le contenu pédagogique ajouté par les enseignants concepteurs de la matière à enseigner.
- S'auto-évaluer en répondant aux quiz proposés par le concepteur du cours.
- Collaborer avec les apprenants du même groupe dans le cadre de l'apprentissage et de la résolution collaborative des problèmes proposés
- Communiquer avec les autres acteurs humains (autres enseignants et autres apprenants).

• Administrateur

- Gérer les comptes d'utilisateurs (enseignants et apprenants).
- Veiller à la sécurité du système.

Après l'identification des acteurs et leurs cas d'utilisation dans notre plateforme, on va les résumer dans le diagramme de cas d'utilisation, présenté dans la **figure 22**.

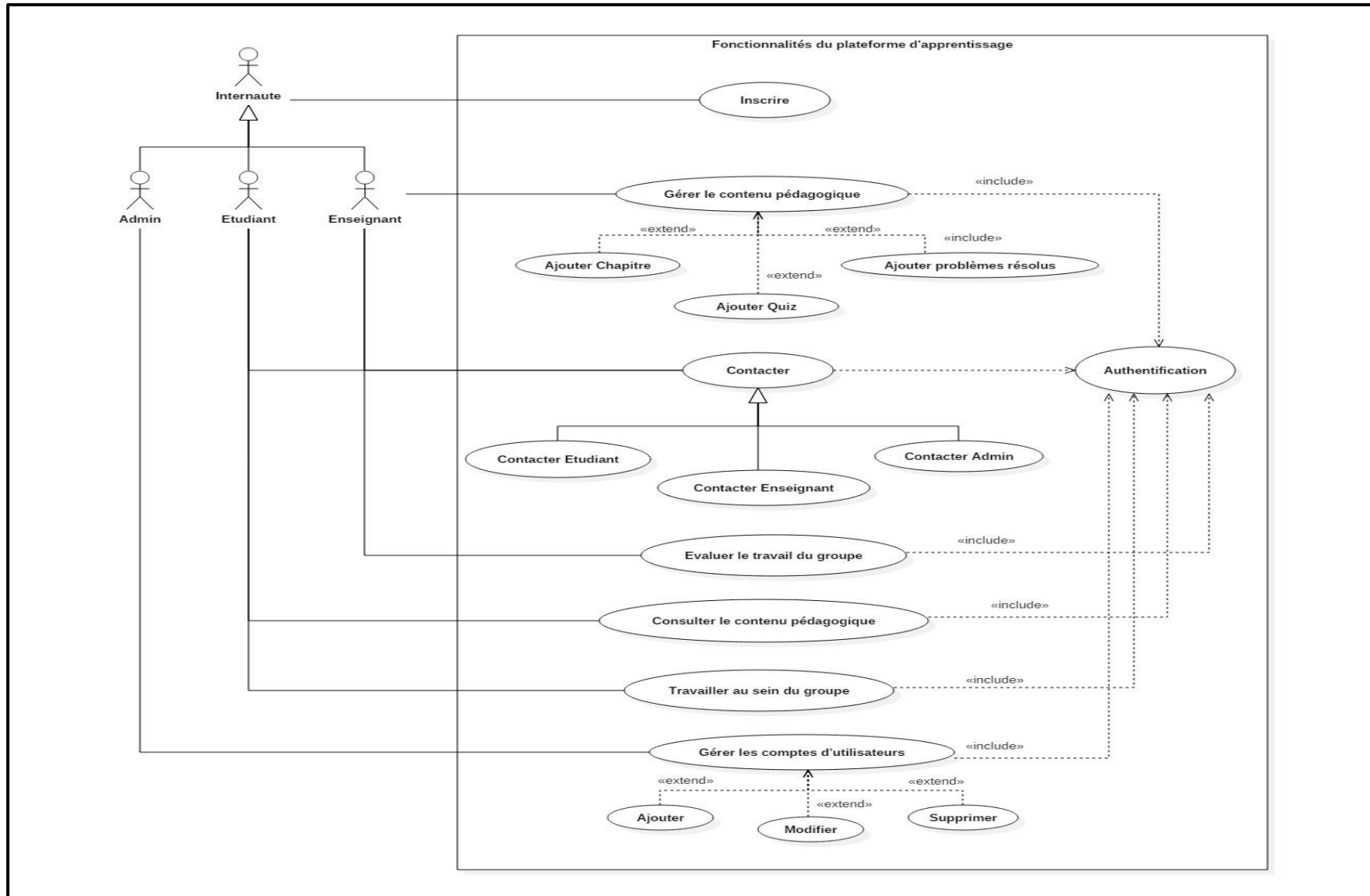


Figure 22. Diagramme des cas d'utilisation du système POOLISA.

La figure ci-dessous (**Figure 23**) présente le diagramme de cas d'utilisation d'espace apprenant.

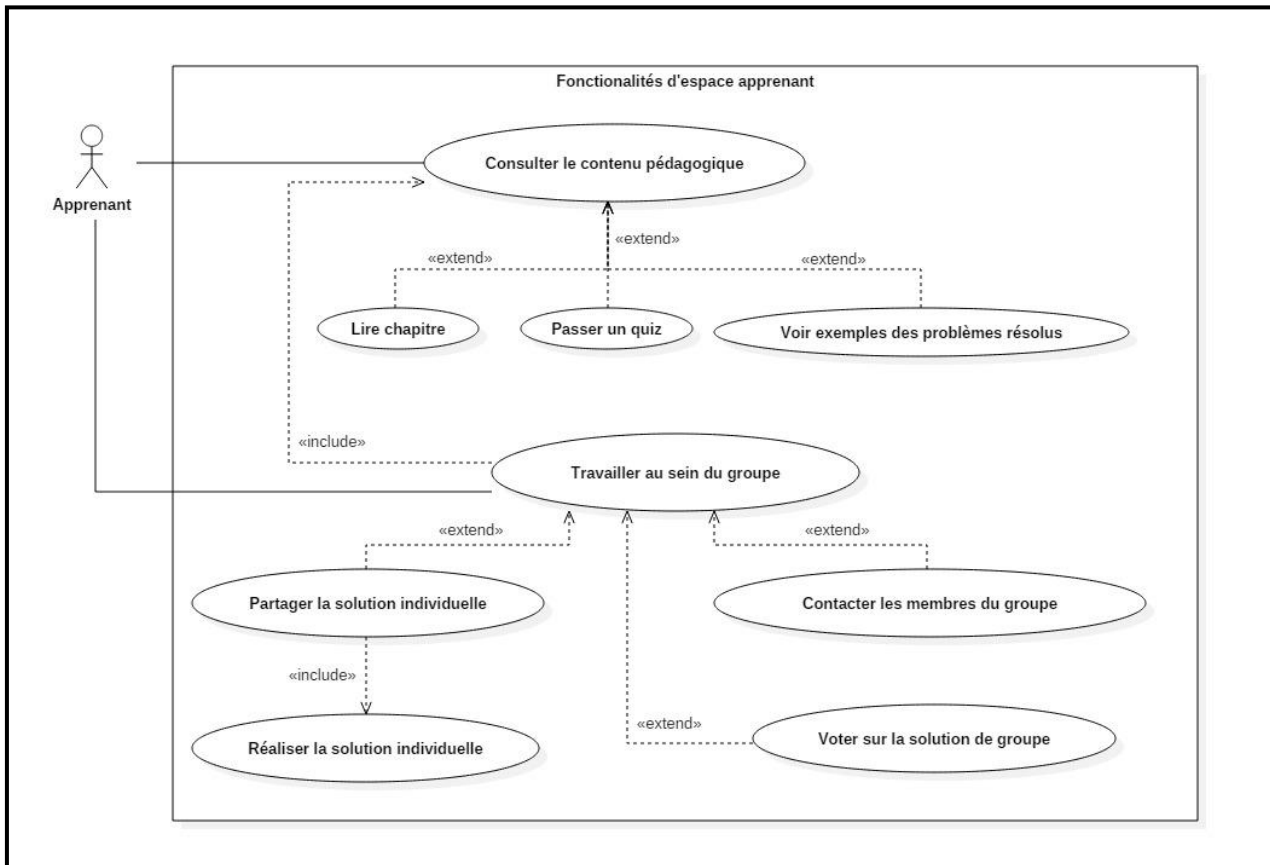


Figure 23. Diagramme des cas d'utilisation d'espace apprenant

6. Architecture du système

La plateforme **POOLISA** est composée de trois espaces : espace Enseignant, espace Etudiant, et espace Administration.

6.1. Espace Enseignant

L'accès à cet espace demande une inscription à la plateforme en tant qu'enseignant. Avant de s'inscrire, l'enseignant doit d'abord contacter l'administration pour bien confirmer de sa qualification scientifique à donner des cours dans cette plateforme. Après cette confirmation, l'administration envoie un code à l'enseignant, lequel lui permet d'entamer l'inscription dans la plateforme **POOLISA**.

Après l'inscription, l'enseignant accède à son espace Enseignant pour créer ses cours. L'enseignant peut donc ajouter et gérer plus d'un cours selon les besoins pédagogiques. Ceci, dans le but de rendre notre méthode **LISA** ainsi que notre plateforme d'apprentissage en ligne supportable à l'enseignement et à l'apprentissage d'autres cours, et non seulement à la **POO**.

Chaque cours (formation) dans la plateforme est caractérisé par une période d'inscription bien définie, ainsi qu'un test d'entrée. Ce test a pour but d'assurer que l'accès à un cours, et par conséquent l'appartenance à l'un de ses groupes d'apprentissage, est limité seulement aux apprenants qui ont un niveau de connaissances acceptable dans le domaine général de ce cours. Ceci afin d'assurer que chaque apprenant inscrit (donc a bien passé le test d'entrée) à un cours particulier, a réellement une capacité à aider ses co-apprenants dans l'apprentissage et le travail collaboratif demandé dans ce cours. Dans notre cas avec la **POO**, l'enseignant prépare donc un test d'entrée sur l'Informatique en général, l'Algorithmique, et la Programmation en particulier.

Après la création du cours, l'enseignant commence à ajouter le contenu pédagogique selon le principe de la méthode **LISA**, en ajoutant : des chapitres, des quiz et ressources (différents types de fichiers) associées à ces chapitres, un ensemble de problèmes résolus comme des exemples pédagogiques, et enfin des travaux finaux à résoudre par groupes d'apprenants dans un délai bien défini, avec toujours la possibilité d'éditer (modifier, supprimer) le contenu pédagogique ajouté.

En plus de l'ajout et la modification du contenu pédagogique, l'espace enseignant offre aussi un outil de messagerie entre l'enseignant et les trois acteurs de notre système : d'autres enseignants, les étudiants et l'administrateur.

6.2. Espace Etudiant / Apprenant

L'accès à cet espace demande une inscription à la plateforme en tant qu'étudiant. Ensuite, pour assister à un cours (ou une formation) particulier (e) dans la plateforme, l'étudiant doit d'abord réussir à passer le test d'entrée à ce cours durant la période d'inscription. Après cette étape, l'étudiant sera affecté directement et automatiquement à son groupe d'apprentissage.

La première étape pédagogique dans cet espace est l'étape **L (Learn)**. L'étudiant dans cette étape consulte les chapitres de cours, lesquels sont composés de concepts. Ensuite, il peut évaluer son niveau d'apprentissage des concepts de ces chapitres, en passant des quiz dans le cadre d'une auto-évaluation.

Ensuite, avant de passer à la deuxième étape **I (Imagine)** afin de consulter l'ensemble des problèmes résolus, l'étudiant doit d'abord achever l'étape **L (Learn)** en terminant tous les chapitres obligatoires dans ce cours.

Enfin, l'étudiant passe à l'étape pratique. Ceci, en utilisant un éditeur de conception orientée objet graphique partagée en temps réel avec ses co-apprenants du groupe d'apprentissage. En plus, l'apprenant peut utiliser un outil de chat pour collaborer entre eux dans la réalisation des travaux finaux à rendre dans un délai défini, dans le cadre de l'étape **S (Share)** et l'étape **A (Approuve)**.

L'espace étudiant offre aussi un outil de messagerie entre l'étudiant et les trois acteurs de notre système : d'autres étudiants, les enseignants et l'administrateur.

6.3. Espace Administration

Cet espace a un accès limité seulement aux personnes autorisées par l'administration de la plateforme. L'administrateur peut donc ajouter des nouveaux utilisateurs (enseignants ou étudiants), de modifier les données des utilisateurs inscrits et de supprimer les comptes enseignants et étudiants.

L'espace Administration offre aussi un outil de messagerie entre l'administrateur et les deux acteurs principaux de notre système : les étudiants et les enseignants.

6.4. Les Entités du Système POOLISA

Les différentes entités déterminantes dans le fonctionnement de notre système, sont détaillées dans les tableaux ci-dessous indiqués, et peuvent être distinguées comme suit :

User ; Cours ; Test ; Chapitre ; C o n c e p t ; Titre ; Questions ; Réponse ; Score ; Fichier ; Messages ; Groupe ; UserGroupe ; Chat ; Exemples ; Etu_Termin ; Projet ; Solution ; Rating ; Solution_Grp.

User : cette entité définit les trois acteurs (utilisateurs) du système (**tableau 2**).

Tableau 2. Description de l'entité User

Nom de l'entité	Définition de l'entité	Définition des attributs de l'entité	Type (longueur)
User	Utilisateur de système	Id : identificateur de l'utilisateur Pseudo : pseudonyme de l'utilisateur Pass : mot de passe de l'utilisateur Statut : le rôle de l'utilisateur dans le système (Etudiant, Enseignant ou Admin) Nom : nom de l'utilisateur Prénom : prénom de l'utilisateur Date : date d'inscription LastAct : dernière activité Adresse : adresse-mail de l'utilisateur	Entier (3) Alphanumérique (20) Alphanumérique (20) Alphanumérique (30) Alphabétique (20) Alphabétique (20) DateTemps Temps Alphanumérique (45)

Cours : est cette entité qui définit les cours de la plateforme (**Tableau 3**).

Tableau 3. Description de l'entité Cours

Nom de l'entité	Définition de l'entité	Définition des attributs de l'entité	Type (longueur)
Cours	Les Cours Ajoutés	Id_cours : identificateur de cours Id : identificateur d'enseignant responsable de cours Nom_c : titre de cours Desc_c : description de cours Date : date de création de cours DateD : date début d'inscription DateF : date fin d'inscription	Entier (3) Entier (3) Alphanumérique (100) Alphanumérique (200) DateTemps DateTemps DateTemps

Test : cette entité définit le test d'entrée (d'inscription) à un cours particulier dans la plateforme (**tableau 4**).

Tableau 4. Description de l'entité **Test**

Nom de l'entité	Définition de l'entité	Définition des attributs de l'entité	Type (longueur)
Test	Test d'entrée au cours	Id: identificateur de test Id_etu : identificateur d'apprenant id_c : identificateur de cours Date : date de création de cours	Entier (6) Entier (3) Entier (3) DateTemps

Chapitre : est cette entité qui définit les chapitres associés aux cours (**Tableau 5**).

Tableau 5. Description de l'entité **Chapitre**

Nom de l'entité	Définition de l'entité	Définition des attributs de l'entité	Type (longueur)
Chapitre	Les Chapitre associées au cours	Id_chapitre : identificateur du chapitre Id : identificateur du cours de ce chapitre Nom_chap : nom du chapitre Desc_c : description du chapitre Requis : chapitre obligatoire ou non Date : date d'ajout de chapitre	Entier (3) Entier (3) Alphanumérique (100) Alphanumérique (200) Entier (1) DateTemps

Concept : est cette entité qui définit les concepts du chapitre (**Tableau 6**).

Tableau 6. Description de l'entité **Concept**

Nom de l'entité	Définition de l'entité	Définition des attributs de l'entité	Type (longueur)
Concept	Les Concepts du Chapitre	Id : identificateur de concept Id_chap : identificateur de chapitre de concepts Nom_concept : nom de concept Num : numéro de concept Contenu : contenu de concept Date : date d'ajout de concept	Entier (3) Entier (3) Alphanumérique (200) Entier (1) LongBlob DateTemps

Titre : est cette entité qui définit les quiz associés au chapitre (**Tableau 7**).

Tableau 7. Description de l'entité **Titre**

Nom de l'entité	Définition de l'entité	Définition des attributs de l'entité	Type (longueur)
Titre	Les Quiz de chapitre	Id_chapitre : identificateur de Quiz Nom : titre de Quiz Id_chap : identificateur de chapitre de Quiz Id_cou : identificateur de cours	Entier (4) Alphanumérique (200) Entier (3) Entier (3)

Questions : est cette entité qui définit les questions associées au Quiz (**Tableau 8**).

Tableau 8. Description de l'entité **Questions**

Nom de l'entité	Définition de l'entité	Définition des attributs de l'entité	Type (longueur)
Questions	Questions de Quiz	Id : identificateur de la question	Entier (4)
		question : les question posées	Alphanumérique (350)
		Note : la note de la question	Entier (1)
		Id_titre : identificateur du Quiz de cette question	Entier (4)

Réponse : est cette entité qui définit les réponses associées aux questions du Quiz (**Tableau 9**).

Tableau 9. Description de l'entité **Réponse**

Nom de l'entité	Définition de l'entité	Définition des attributs de l'entité	Type (longueur)
Réponse	Réponse des questions de Quiz	Id : identificateur de la réponse	Entier (6)
		Réponse : la réponse choisie	Alphanumérique (350)
		Correct : état de la réponse (correcte ou incorrecte)	Entier (1)
		Id_questions : identificateur de la question associée à cette réponse	Entier (4)

Score : est cette entité qui définit la note du QCM (**Tableau 10**).

Tableau 10. Description de l'entité **Score**

Nom de l'entité	Définition de l'entité	Définition des attributs de l'entité	Type (longueur)
Score	Note de Quiz	Id : identificateur de la note du Quiz	Entier (6)
		Id_etu : identificateur de l'étudiant	Entier (3)
		Id_qcm : identificateur du Quiz	Entier (4)
		Score : la note obtenue dans le Quiz	Float (2)
		Date_Qcm : date de passage du Quiz	DateTemps

Fichier : est cette entité qui définit les différents fichiers associés aux chapitres (**Tableau 11**).

Tableau 11. Description de l'entité **Fichier**

Nom de l'entité	Définition de l'entité	Définition des attributs de l'entité	Type (longueur)
Fichier	Les fichiers associés au chapitre	Id_fichier : identificateur du fichier	Entier (3)
		Id_ch : identificateur du chapitre de ce fichier	Entier (3)
		Nom_f : nom du fichier	Alphanumérique (100)
		File_url : lien d'emplacement du fichier	Alphanumérique (255)

Messages : est cette entité qui définit les messages échangés entre les utilisateurs (Tableau 12).

Tableau 12. Description de l'entité Messages

Nom de l'entité	Définition de l'entité	Définition des attributs de	Type (longueur)
Messages	Les messages échangés entre les utilisateurs	Id : identificateur du message Id_expediteur : identificateur de l'expéditeur du message Id_destinataire : identificateur du destinataire du message Message : le contenu du message Date : date d'envoi du message Vu : message vu ou pas	Entier (4) Entier (3) Entier (3) Text DateTemps Entier (1)

Groupe : est cette entité qui définit les groupes d'apprentissage dans un cours particulier dans la plateforme (Tableau 13).

Tableau 13. Description de l'entité Groupe

Nom de l'entité	Définition de l'entité	Définition des attributs de l'entité	Type (longueur)
Groupe	Les Groupes d'apprentissage	Id_grp : identificateur du groupe Id_c : identificateur du cours	Entier (3) Entier (3)

UserGroupe : est cette entité qui définit la relation d'appartenance de l'apprenant à un groupe d'apprentissage dans un cours particulier (**Tableau 14**).

Tableau 14. Description de l'entité UserGroupe

Nom de l'entité	Définition de l'entité	Définition des attributs de l'entité	Type (longueur)
User-Groupe	Les Groupes d'apprentissage	<p>Id : identificateur de relation d'appartenance</p> <p>Id_grp : identificateur du groupe</p> <p>Id_etu : identificateur de l'apprenant</p>	<p>Entier (6)</p> <p>Entier (3)</p> <p>Entier (3)</p>

Chat : est cette entité qui définit les messages échangés entre les membres d'un groupe d'apprentissage particulier (**Tableau 15**).

Tableau 15. Description de l'entité Chat

Nom de l'entité	Définition de l'entité	Définition des attributs de l'entité	Type (longueur)
Chat	Les messages échangés dans un groupe d'apprentissage	<p>Id : identificateur de message</p> <p>Message : contenu de message</p> <p>From_pseudo : pseudonyme d'émetteur de message</p> <p>Created : temps d'envoi de message</p> <p>Id_grp : identificateur de groupe d'apprentissage d'émetteur de message</p>	<p>Entier (11)</p> <p>Alphanumérique (300)</p> <p>Alphanumérique (20)</p> <p>DateTemps</p> <p>Entier (3)</p>

Exemples : est cette entité qui définit les exemples des problèmes résolus (**Tableau 16**).

Tableau 16. Description de l'entité **Exemples**

Nom de l'entité	Définition de l'entité	Définition des attributs de l'entité	Type (longueur)
Exemples	Exemples ajoutés	Id_exp : identificateur d'exemple Id_co : identificateur de cours d'exemple Nom_exp : titre d'exemple Contenu : contenu d'exemple Date : date d'ajout d'exemple	Entier (3) Entier (3) Alphanumérique (100) LongBlob DateTemps

Etu Termin: est cette entité qui définit les traces de terminaison d'activités (chapitres et exemples) pour un apprenant particulier (**Tableau 17**).

Tableau 17. Description de l'entité **Etu Termin**

Nom de l'entité	Définition de l'entité	Définition des attributs de l'entité	Type (longueur)
Etu Termin	Les traces d'un apprenant	Id : identificateur de trace Id_etu : identificateur de l'apprenant Id_cours : identificateur de cours Id_chap : identificateur de chapitre Id_concept : identificateur de concept Id_exp : identificateur d'exemple Date : date de terminaison	Entier (11) Entier (3) Entier (3) Entier (3) Entier (3) Entier (3) DateTemps

Projet : est cette entité qui définit les projets finaux (**Tableau 18**).

Tableau 18. Description de l'entité **Projet**

Nom de l'entité	Définition de l'entité	Définition des attributs de l'entité	Type (longueur)
Projet	Projets ajoutés	Id_prj : identificateur de projet Id_co : identificateur de cours de projet Nom_prj : titre de projet Desc_prj : description de projet Contenu : contenu d'exemple Date : date d'ajout d'exemple DateRemise : date de remise de projet	Entier (3) Entier (3) Alphanumérique (100) LongBlob LongBlob DateTemps DateTemps

Solution : est cette entité qui définit les solutions d'un projet final particulier (**Tableau 19**).

Tableau 19. Description de l'entité **Solution**

Nom de l'entité	Définition de l'entité	Définition des attributs de l'entité	Type (longueur)
Solution	Solution du Projet final	Id : identificateur de la solution Id_etu : identificateur de l'apprenant propriétaire de solution Id_prj : identificateur de projet Id_grp : identificateur de groupe d'apprentissage de propriétaire de solution Json : sauvegarde de la solution sous le type Json Img : sauvegarde de la solution sous le type Image (jpeg) Date : date d'ajout d'exemple	Entier (6) Entier (3) Entier (3) Entier (3) LongText LongBlob DateTemps

Rating : est cette entité qui définit le système de vote sur les solutions (Tableau 20).

Tableau 20. Description de l'entité **Rating**

Nom de l'entité	Définition de l'entité	Définition des attributs de l'entité	Type (longueur)
Rating	Note de vote sur une solution	Id_rating : identificateur de la note Solution_id : identificateur de la solution Id_etu : identificateur de l'apprenant sélecteur Note : la note attribuée Date : date de vote	Entier (3) Entier (6) Entier (3) Entier (3) DateTemps

Solution_Grp : est cette entité qui définit la solution du groupe d'un projet final (Tableau 21).

Tableau 21. Description de l'entité **Solution_Grp**

Nom de l'entité	Définition de l'entité	Définition des attributs de l'entité	Type (longueur)
Solution_Grp	Solution du Groupe	Id : identificateur de la solution du groupe Solution_id : identificateur de la solution sélectionnée Id_grp : identificateur de groupe d'apprentissage de propriétaire de solution Id_prj : identificateur de projet Tirage : méthode de sélection de solution du groupe Note : note d'enseignant Remarques : remarques d'enseignant	Entier (3) Entier (6) Entier (3) Entier (3) Entier (1) Entier (2) Alphanumérique (300)

Ainsi, et pour plus de compréhension, la figure ci-dessous (**Figure 24**), résume les différentes tables de la base de données du système **POOLISA**, et les relations entre elles.

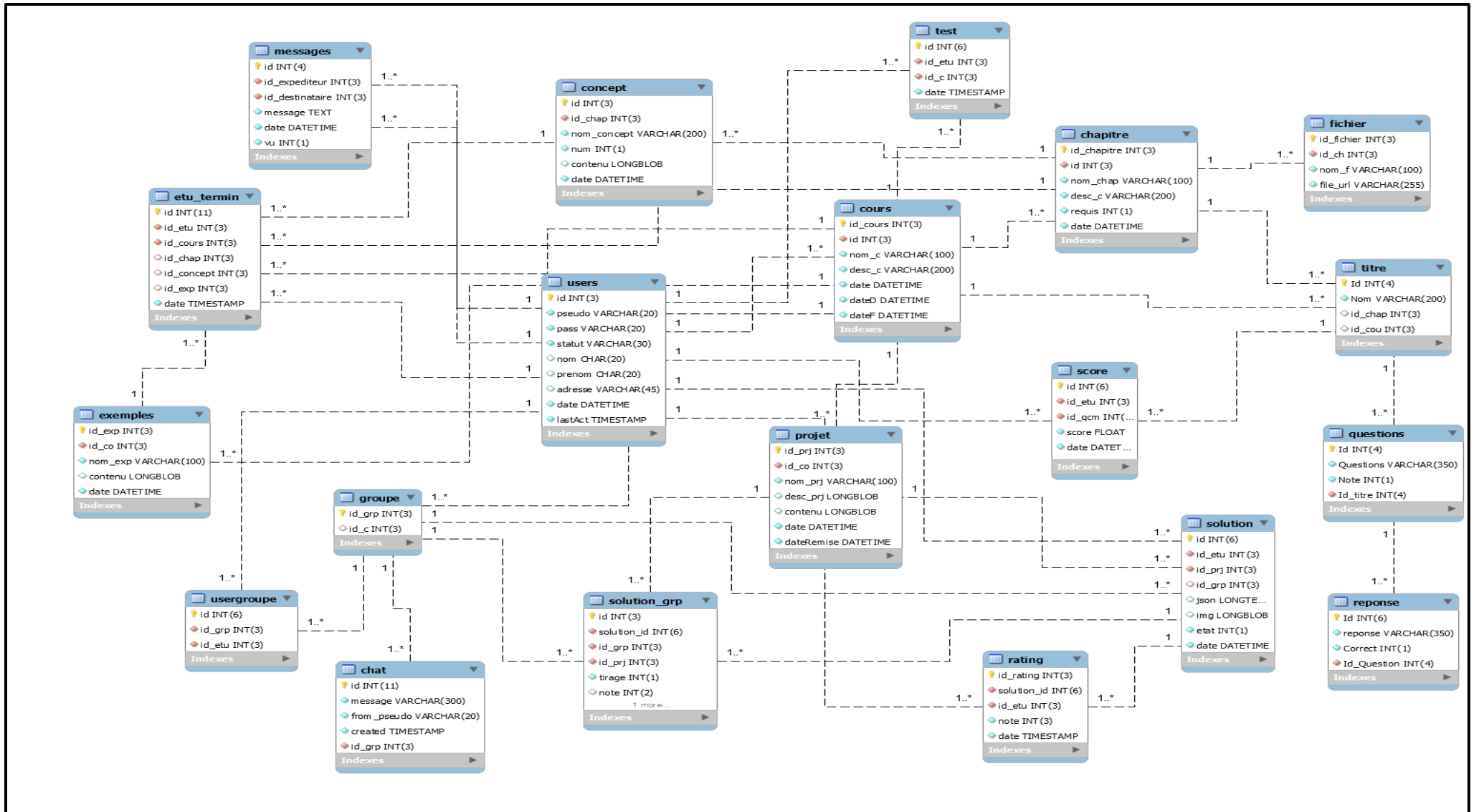


Figure 24. Base de Données du Système POOLISA.

7. Conclusion

Dans ce chapitre, nous avons montré et illustré notre vision sur la conception d'un système pour l'apprentissage des principes de la programmation orientée objet, chez les étudiants universitaires. Où nous avons constaté que la difficulté principale dans cet apprentissage, est au niveau conceptuel.

Ensuite, et en basant sur les résultats obtenus à partir de cette étude, nous avons proposé les bases de notre méthode d'apprentissage collaboratif en ligne (**LISA**) pour ce module pédagogique ainsi que d'autres modules ayant une structure pédagogique similaire. Puis, nous avons présenté en détail les bases de cette méthode ainsi que la description de l'apprentissage collaboratif appliqué au sien de ses étapes.

Enfin, nous avons présenté la conception et l'architecture d'un système nommé **POOLISA**, adoptant la méthode proposée, dans le but de la concrétiser et de la valider dans le cadre de la dernière étape de ce travail de recherche.

Chapitre III. Implémentation

Chapitre III. Implémentation du Système POOLISA

1. Introduction

Dans ce chapitre, nous allons présenter les outils utilisés dans le développement de notre système **POOLISA**, de la conception jusqu'à l'implémentation, en présentant quelques interfaces et fonctionnalités de ce système.

2. Outils utilisés

2.1. Au niveau de la conception

Afin de concevoir le diagramme de cas d'utilisation utilisé dans la conception de notre système **POOLISA**, on a utilisé le logiciel **StarUml** comme montré dans la **Figure 25**.

- **StarUml** : est un logiciel dédié à la modélisation des diagrammes UML, lequel a été employé dans notre cas, pour la réalisation du diagramme de cas d'utilisation de notre projet (**Figure 25**).

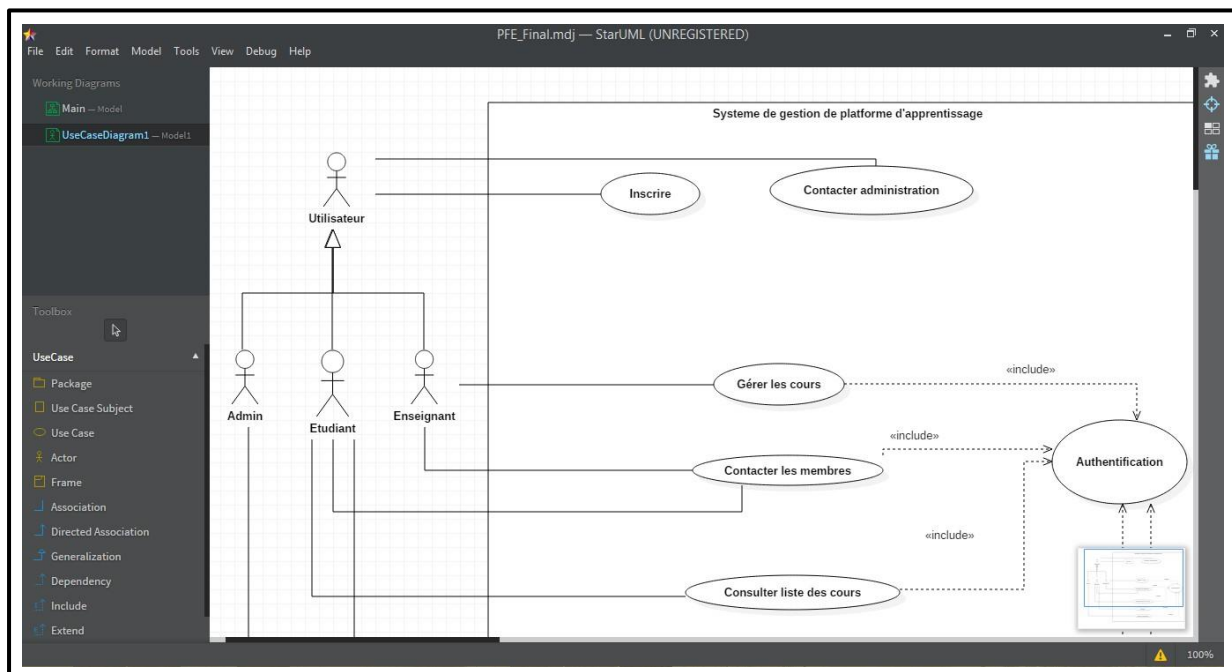


Figure 25. Interface de logiciel *StarUml*.

2.2. Au niveau de l'implémentation

Afin de développer notre Système, on a utilisé l'éditeur de texte multi-langages *Dreamweaver*.

- ***Dreamweaver*** : un éditeur de site web WYSIWYG pour Microsoft Windows, et Mac OSX créé en 1997, commercialisé par Macromedia, puis Adobe Systèmes sous licence utilisateur final (**Figure 26**) (**Href 9**).

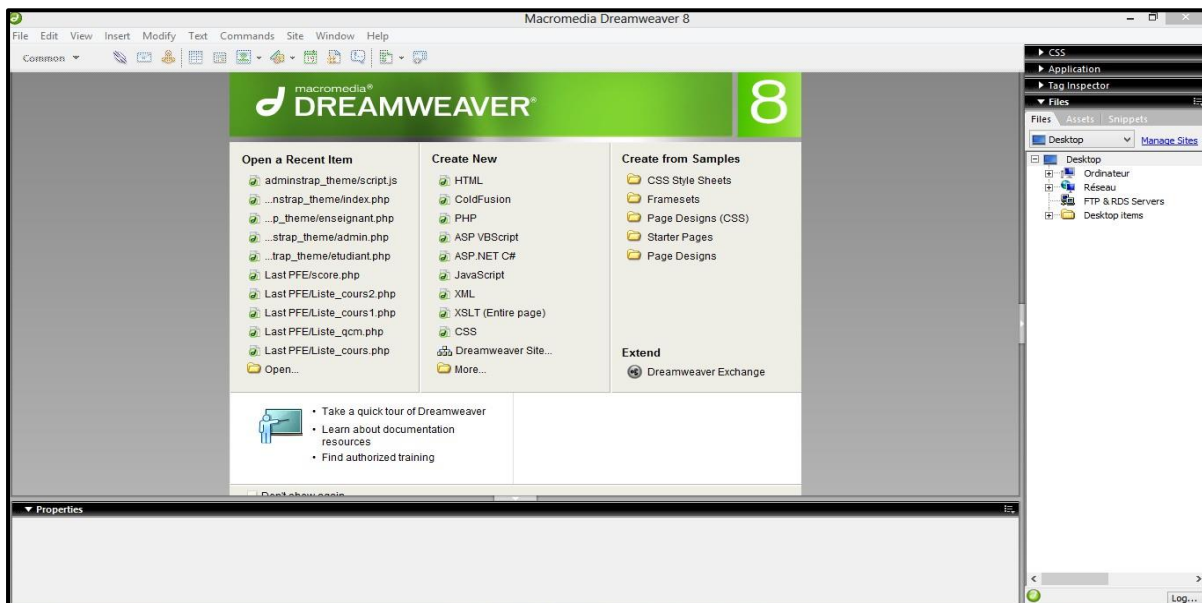


Figure 26. Interface Dreamweaver.

En parallèle, plusieurs types de *langages et Framework*, comme détaillé ci-dessous, ont été utilisés dans le développement du système **POOLISA**, à savoir :

- *MySQL* (*My Structured Query Language*) ;
- *SQL* (*Structured Query Language*) ;
- *PHP* (*Hypertext Preprocessor*) ;
- *JavaScript* ;
- *HTML* (*HyperText Markup Language*) ;
- *HTML5* (*HyperText Markup Language 5*) ;
- et *Bootstrap*.

- **MySQL (My Structured Query Language)** : est un système de gestion de base de données (SGBD), à utilisation gratuite. C'est un serveur de base de données relationnelle SQL, conçu pour la construction des sites web en association avec PHP (langage) et Apache (serveur web), pour simuler un serveur local afin de stocker et d'accéder aux bases de données (Figure 27).

En effet, le SQL (*Structured Query Language*) est un langage universellement normalisé servant à interroger et modifier le contenu d'une base de données relationnelle (Href 10).

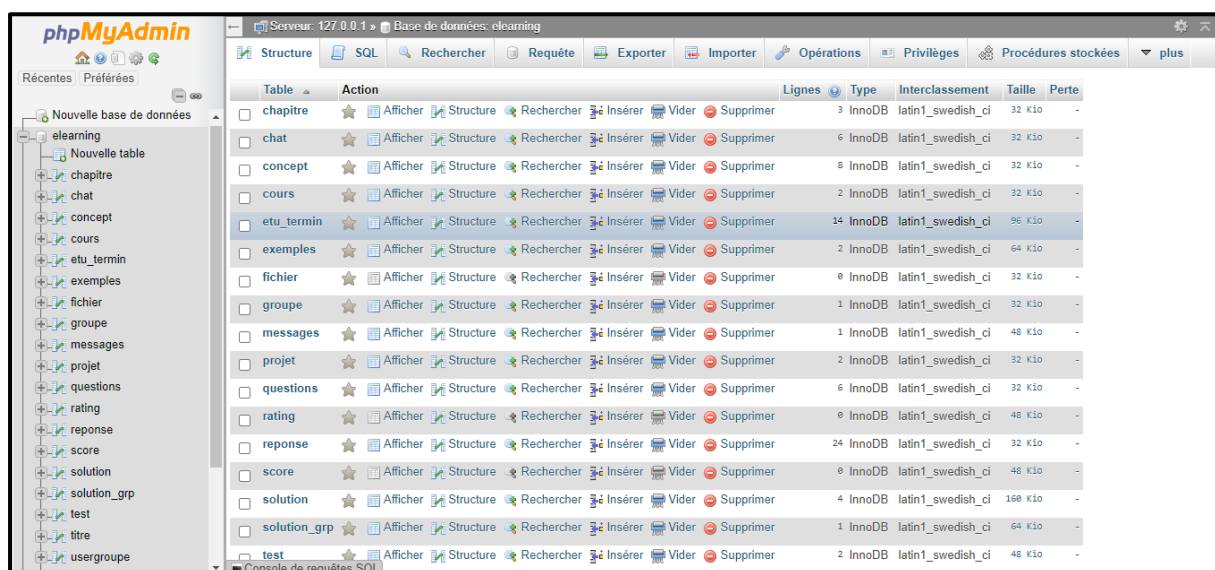


Figure 27. Interface *PhpMyAdmin*.

But d'utilisation : cet outil a été utilisé pour la création, le stockage, la consultation, et le traitement de la base de données du système POOLISA.

- **PHP (Hypertext Preprocessor)** : est un langage de programmation, permettant la création des pages Web dynamiques, Indépendant de la plateforme utilisée puisqu'il est exécuté, côté serveur et non côté client. La syntaxe du langage provient de celles du langage C, du Perl et de Java (Href 11).

But d'utilisation : ce langage a été utilisé pour la **création des pages web dynamiques** et pour l'**échange des données** d'une part entre les pages web, et d'autre part entre l'application web et la base de données, en appliquant des requêtes sur les tables de cette dernière.

- **HTML (HyperText Markup Language) :** HTML (HyperText Markup Language) n'est pas un langage de programmation : c'est un langage de balisage qui sert à indiquer au navigateur comment structurer les pages web visitées. Il peut être aussi compliqué ou aussi simple que le développeur web souhaite qu'il soit. Le HTML se compose d'une série d'éléments avec lesquels vous pouvez encadrer, envelopper ou baliser différentes parties du contenu pour les faire apparaître ou agir d'une certaine manière. Des balises encadrantes peuvent transformer une petite partie de contenu en un lien vers une autre page sur le Web, mettre des mots en italique (**Href 12**).

But d'utilisation : ce langage a été utilisé pour la création des pages web statiques de notre application.

- **HTML5 (HyperText Markup Language 5) :** est la dernière révision majeure du HTML ; elle contient des Balises HTML prédéfinies telles que header, section, footer...

But d'utilisation : cette version du langage **HTML** a été utilisée dans le but de rendre l'interface du système responsive aux différentes tailles d'écran.

- **JavaScript :** est un langage de programmation, permettant au programmeur de créer et d'implémenter des contenus mis à jour de façon dynamique, et des mécanismes complexes sur une page web, comme le contrôle des contenus multimédias, l'animation des images 2D/3D, et la création des menus vidéo défilants. En effet, JavaScript est devenu très utilisé dans le design des applications web. C'est la troisième couche (avec l'HTML et CSS) des technologies standards du web, lequel a de bonnes chances d'être impliqué (**Href 13**).

But d'utilisation : ce langage a été utilisé, pour créer des contenus mis à jour de façon dynamique et pour faire un lien entre les pages web et les données de la base de données.

- **Bootstrap :** est un Framework développé par l'équipe du réseau social Twitter. Il est en open source (sous licence MIT) et utilise les différents langages HTML, CSS et JavaScript (**Href 14**).

But d'utilisation : cet outil a été utilisé, pour créer les pages de notre site avec désigne responsive, adaptable avec toute taille et tout type d'écrans, surtout les Smartphones.

- **CSS :** est un langage informatique utilisé sur l'internet pour mettre en forme les fichiers HTML ou XML. Ainsi, les feuilles de style, aussi appelé les fichiers CSS, comprennent du code qui permet de gérer le design d'une page en HTML (**Href 15**).

But d'utilisation : cet outil a été utilisé, pour personnaliser le design des pages web de la plate-forme avec la capacité de les modifier facilement et une seule fois en modifiant une seule partie du fichier CSS.

3. Interfaces et fonctionnalités du système POOLISA

Dans cette partie, nous allons présenter quelques interfaces et fonctionnalités de la plateforme **POOLISA**.

Accueil du Plateforme POOLISA :

Cette interface présente dans la **Figure 28**, est l'accueil du Plateforme **POOLISA**.

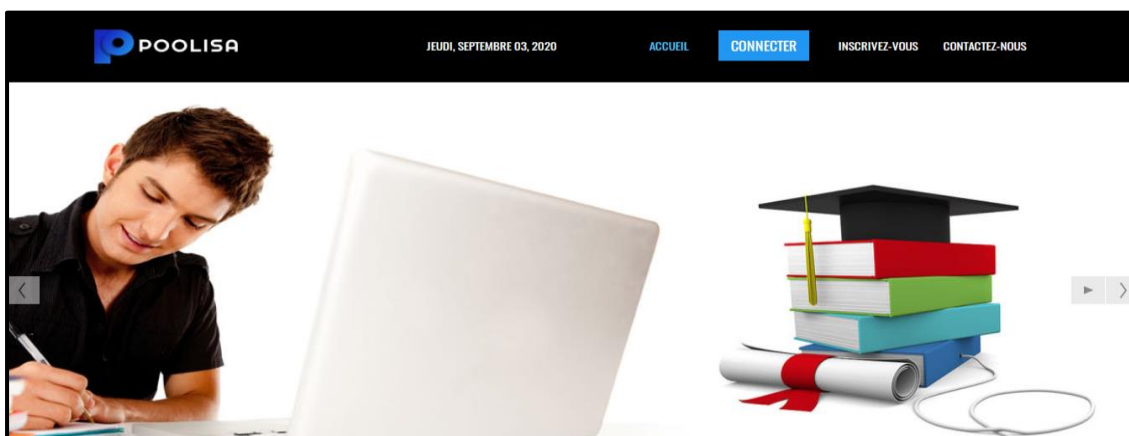


Figure 28. Accueil du Plateforme **POOLISA**.

Contenu de Cours :

Cet espace illustré dans la **Figure 29**, permet l'accès aux étapes de la méthode **LISA** (Learn, Imagine, Share et Approve).



Figure 29. Contenu du cours POO.

L'étape Learn (Apprendre) :

Ces espaces illustrés dans les deux **Figures 30 et 31**, présentent le contenu de l'étape **Learn**.



Figure 30. Learn – Liste de chapitres de cours POO.

The screenshot shows the POOLISA learning platform interface. At the top, the logo 'POOLISA' is on the left, and the user name 'CHOUINI Moncef' is on the right. Below the header, a breadcrumb trail reads: 'Tableau de bord / Cours POO / Apprendre / Chapitre Introduction à la POO / Concept 1. Historique de la POO'. On the left, a sidebar menu lists 'Mes activités' with options: Retour, Cours, Apprendre (5), Imaginer (3), Partager & Approuver (2), Messagerie, and Mes Scores. The main content area is titled '1. Historique de la POO' and contains a text block: 'La programmation orientée objet (POO), ou programmation par objet, est un paradigme de programmation informatique élaboré par les Norvégiens Ole-Johan Dahl et Kristen Nygaard au début des années 1960 et poursuivi par les travaux de l'Américain Alan Kay dans les années 1970. Il consiste en la définition et l'interaction de briques logicielles appelées objets.' Below the text is a green button labeled 'Marquer comme terminé'. On the right, a 'SOMMAIRE' sidebar lists '1. Historique de la POO', '2. Définition', 'Ressources', and 'Auto-Evaluation'. Below that, a 'Co-Apprenants' section lists 'BENYOUUNES Ali', 'ABDERRAHIM Oussama', 'SOULIAH Isslem', and 'Groupe Chat'.

Figure 31. Learn – Contenu d'un chapitre.

Les étapes Share & Approve (Partager et Approuver) :

Les Figures 32; 33; 34, présentent les deux étapes *Share et Approve*.

The screenshot shows the POOLISA learning platform interface for a project simulation. The breadcrumb trail is: 'Tableau de bord / Cours POO / Partager & Approuver / Projet Final Simulation'. The left sidebar menu is identical to Figure 31, with 'Partager & Approuver' (2) highlighted. The main content area is titled 'Enoncé du Projet Simulation Cours POO' and shows a timer: 'Temps restant : 12j 12h 56m 33s'. The text block contains: 'Un Verre est caractérisé par sa capacité et sa couleur. On peut ajouter une certaine quantité de liquide dans la verre, retirer une quantité et consulter quelle quantité est contenue dans le verre. Certains verres sont dessinés par des citations. On veut simuler ce verre avec un programme orientée objet, donner la conception *Orientée Objet de ce programme*.' Below the text is a blue button labeled 'Résolution Conceptuelle'. At the bottom, a section titled 'Solutions du Problème' shows the date '2020-09-16 11:03:09' and an 'Actualiser' button. Below this, it says 'Vous n'avez pas ajouté une Solution' with an 'Ajouter Solution' button, and 'Aucun Co-Apprenant a Ajouté une Solution'. The right sidebar 'Co-Apprenants' section is also identical to Figure 31.

Figure 32. Share & Approve – Contenu d'un Projet Final.

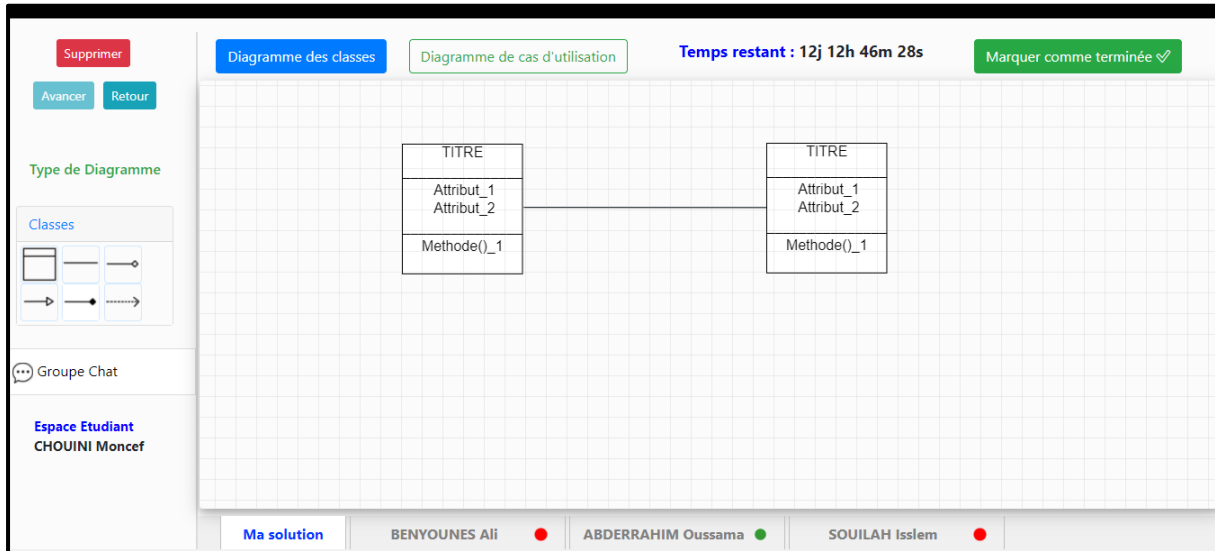


Figure 33. Share & Approve – Résolution Conceptuelle d’un Projet final.

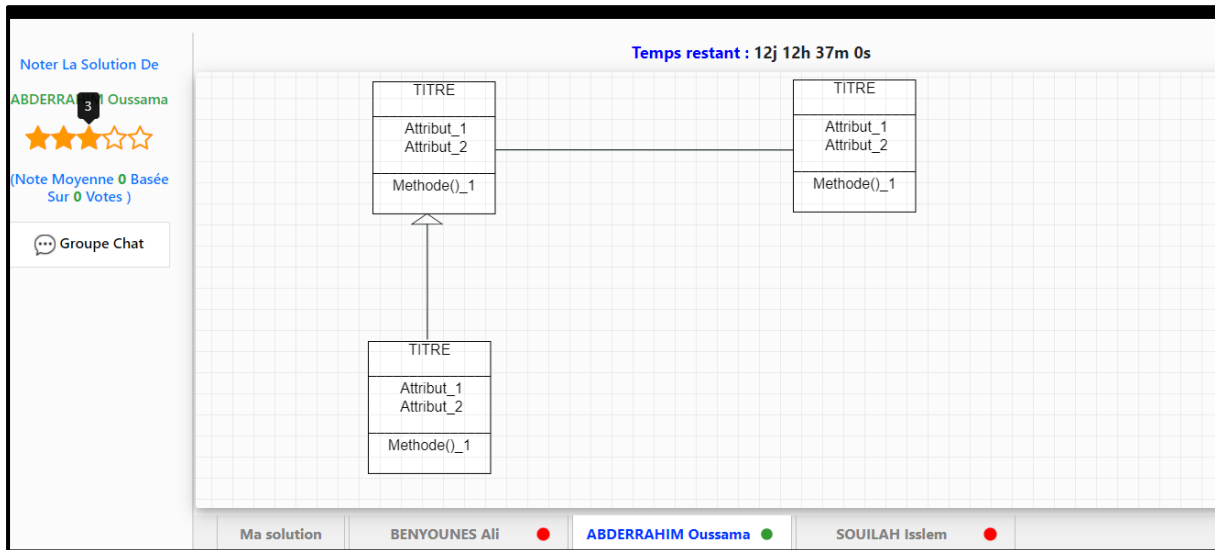


Figure 34. Share & Approve – Noter la solution d’un Co-Apprenant.

4. Expérimentation

Pour valider notre méthode **LISA**, nous avons effectué une étude expérimentale sur un groupe de quatre étudiants du département d'informatique d'université 8 Mai 1945 Guelma, Algérie. Les quatre étudiants pouvaient accéder à notre plateforme à partir de leurs ordinateurs connectés à l'internet.

4.1. Méthodologie :

Afin de connaître les effets de notre méthode sur les niveaux des connaissances des étudiants, nous avons utilisé une méthodologie qui permet de tester la différence entre les notes des étudiants obtenues avant et après l'application de notre méthode **LISA**. Pour ce faire, nous avons préparé un questionnaire à soumettre aux étudiants avant l'utilisation de notre système. Ce questionnaire est composé d'un ensemble de questions à choix multiples sur les notions de base de la programmation orientée objet. Ensuite, les résultats obtenus ont été notés sur 20 points.

Ensuite, les étudiants ont utilisé notre système avec toutes les fonctions offertes. Après la réalisation du projet final, nous avons préparé un autre questionnaire sous forme de questions à choix multiples. De même, les résultats obtenus ont été enregistrés et comparés avec les premières notes obtenues.

Les notes obtenues avant et après l'utilisation de notre système **POOLISA** est présenté dans le tableau 22.

Etudiant	Note avant	Note après
1	8	12
2	11	15
3	6	9
4	10	13

Tableau 22. Statistiques des notes obtenues.

4.2. Résultats et discussion

Pour évaluer l'impact du système **POOLISA** sur le niveau des connaissances de ces étudiants dans les généralités de la **POO**, nous proposons de vérifier l'hypothèse de recherche suivante:

Hypothèse nulle H0: le système développé **POOLISA** n'a aucun impact sur l'amélioration du niveau des connaissances des étudiants dans la **POO**.

Hypothèse alternative H1: le système développé **POOLISA** aide les étudiants à améliorer leurs niveaux de connaissances dans la **POO**.

Pour vérifier cette hypothèse, nous avons utilisé un test statistique sachant que le nombre des étudiants est inférieur à 30. En effet, pour savoir si la différence est significative entre les deux notes obtenues, nous avons appliqué le **test t de student** pour échantillons appariés (voir **Annexe 5**). Après l'utilisant d'un logiciel web de **test t de student** en ligne (**Href 16**), nous avons obtenu les résultats suivants :

Groupe	Taille	Moyenne	Ecart-type (ET)
Avant	4	8.75	2.2174
Après	4	12.25	2.5

Tableau 23. Statistique du groupe avant / après utilisation du système.

Tscore	df	p-value	Interval de confiance, 95%
-12.1244	3	0.001208	[-4.4187, -2.5813]

Tableau 24. Statistique t-test.

En analysant les résultats mentionnés dans les tableaux 23 et 24, nous avons remarqué qu'un effet significatif a été trouvé (tscore = -12.1244, p-value = 0.001208) avec un niveau de confiance de 95% ($\alpha = 0,05$). De plus, une différence significative a été établie entre la moyenne pré-test (moyenne = 8.75, ET = 2.2174) et la moyenne post-test (moyenne = 12.25, ET = 2.5). Comme la valeur p calculée est inférieure au niveau de signification $\alpha = 0,05$ ($0.001208 < 0.05$), alors **l'hypothèse nulle H0** est rejetée au profit de **l'hypothèse alternative H1**. Donc, en conséquence nous pouvons dire que notre système **POOLISA** (et

donc notre méthode d'apprentissage collaboratif **LISA**) aide les étudiants à améliorer leurs niveaux de connaissances dans la **POO**.

5. Conclusion

Dans ce dernier chapitre, nous avons présenté les différents outils, langages et Framework utilisés dans le cadre de l'implémentation de notre système **POOLISA**. Ce système contient un ensemble d'interfaces dédiées aux acteurs humains utilisant ce système à savoir les étudiants, l'enseignante et l'administrateur. Chacun d'eux dispose d'un ensemble d'outils facilitant leurs tâches et activités. Enfin, nous avons discuté les résultats d'une expérimentation préliminaire de ce système.

Conclusion générale et perspectives

La présente étude nous a conduits à comprendre le concept de l'apprentissage en ligne **E-Learning** avec ses principes et son utilité à l'ère actuelle, ainsi que sa forme collaborative et son impact sur l'amélioration du niveau d'apprentissage des apprenants. En outre, nous avons constaté à partir d'une analyse d'un ensemble de travaux de recherche, et suite à notre étude de cas réalisée sur 91 étudiants d'informatique à l'université du 8 Mai 1945 Guelma - Algérie, qu'en général, les étudiants d'informatique ont des problèmes et des difficultés dans l'apprentissage du module de la **Programmation Orientée Objet (POO)**. En effet, nous avons enregistré un ensemble de problèmes qui peuvent être résumés comme suit :

- La difficulté de changer la vision de modélisation des problèmes, de la procédurale habituelle chez les étudiants, vers une autre approche orientée objet (changement d'état d'esprit).
- La difficulté d'utiliser l'imagination pour comprendre les concepts de base de la **POO** (classe et objet), et de trouver leurs équivalences dans la vie réelle.
- Un problème au niveau d'outils d'enseignement de la **POO** (langages de programmation trop complexes et destinés aux professionnels et non pas aux débutants, environnements de programmation trop confus...).
- La difficulté de concevoir une conception orientée objet correcte à un problème posé.
- La manière d'introduire la **POO** aux étudiants, où certains enseignants commencent à expliquer la **POO** à travers d'exemples de petits codes sources sous un langage de programmation particulier, au temps où les apprenants ne connaissent pas encore les concepts et les règles orientées objet.

Ainsi, notre étude de cas réalisée a confirmé que le problème majeur dans l'apprentissage de ce module pédagogique est principalement au niveau conceptuel. A cet effet, une nouvelle méthode d'apprentissage collaboratif à distance, nommée **LISA**, a été proposée pour l'apprentissage de la **POO**. Elle englobe un ensemble d'étapes les composant: **Learn, Imagine, Share & Approve**. Ces dernières nous semblaient nécessaires et indispensables à l'apprentissage de ce module, et même à l'apprentissage d'autres modules ayant une structure pédagogique similaire. Ainsi, au cœur des étapes de cette nouvelle méthode, nous avons

intégré la collaboration entre les apprenants avec un ensemble de critères bien définies. En conséquence, nous considérons que l'application de l'apprentissage collaboratif est une contribution originale et intéressante dans l'apprentissage à distance de ce module pédagogique (la **POO**).

Afin de concrétiser notre méthode **LISA**, nous avons procédé à la conception d'un système adoptant cette nouvelle méthode en se basant sur l'un des diagrammes de langage de modélisation unifié (**UML**), qui est le digramme des **cas d'utilisation (UseCase)**. Ce système, appelé **POOLISA**, dispose des mêmes fonctionnalités des plateformes d'apprentissage en ligne. **POOLISA** offre aux apprenants tous les outils nécessaires à la concrétisation des étapes de la méthode **LISA**, avec une utilisation souple et facile.

Les résultats obtenus dans ce travail de fin d'étude sont encourageants. Ils permettent d'ouvrir de nouvelles pistes de recherche, dans l'axe de l'apprentissage en ligne de l'informatique en général, et la **Programmation Orientée Objet POO** en particulier ainsi que l'application de l'apprentissage collaboratif dans ce contexte. En effet, il reste comme perspectives proposé à ce travail, une amélioration au niveau de la méthode de regroupement appliquée dans notre apprentissage collaboratif appliqué lors des étapes de la méthode **LISA**, ainsi qu'une amélioration au niveau de la concrétisation (l'implémentation) de la deuxième étape (*Imagine*) de notre méthode **LISA** dans le système d'apprentissage développé **POOLISA**. Enfin, et afin de valider l'approche proposée, nous proposons de tester le système avec un nombre plus grand d'étudiants d'informatique ou des programmeurs débutants s'intéressant à la programmation orientée objet.

Références bibliographiques

- Benabbou, F., & Hanoune, M. (2006). EasyAlgo: Un environnement d'apprentissage et d'autoévaluation de l'algorithmique. *Automates intelligents*, 76, 1-15.
- Benyounes, A., Boudjehem, R., & Lafifi, Y. (2020). Study of the impact of collaboration among learners during the learning of "Object-Oriented Programming", Article accepté pour présentation dans la conférence Future Learning 2020 8th International Conference on Future Learning and Informatics Data Revolution, Istanbul, Turquie, 20-22 octobre 2020.
- Bermejo, S. (2005). Cooperative electronic learning in virtual laboratories through forums. *IEEE Transactions on Education*, 48(1), 140-149.
- Cheung, R. (2006). A web-based learning environment for object-oriented programming. *International journal of information and operations management education*, 1(2), 140-157.
- Cypher, A., & Halbert, D. C. (Eds.). (1993). *Watch what I do: programming by demonstration*. MIT press.
- Dillenbourg P. (1999). What do you mean by collaborative learning?. In P. Dillenbourg (Ed.), *Collaborative-learning: Cognitive and Computational Approaches*. (p.1-19). Oxford, Elsevier.
- Djelil, F., Albouy-Kissi, B., Albouy-Kissi, A., Sanchez, E., & Lavest, J. M. (2015). Towards a 3D Virtual Game for Learning Object-Oriented Programming Fundamentals and C++ Language Theoretical Considerations and Empirical Results. In *International Conference on Computer Supported Education*.
- Esteves, M., & Mendes, A. J. (2004). A simulation tool to help learning of object oriented programming basics. In *34th Annual Frontiers in Education, 2004. FIE 2004*. (pp. F4C-7). IEEE.
- Guibert, N., Guittet, L., & Girard, P. (2006). Performances et usages d'un environnement d'apprentissage de la programmation basé sur exemple. In *ERGO'IA* (pp. 103-110).
- Guri-Rosenblit, S. (2005). 'Distance education' and 'e-learning': Not the same thing. *Higher education*, 49(4), 467-493.
- Henri, France & Lundgren-Cayrol, Karin. (2001). Apprentissage collaboratif à distance. Pour comprendre et concevoir les environnements d'apprentissage virtuels. *Sainte-Foy: Presses de l'université du Québec*.
- Kölling, M. (1999). Teaching object orientation with the blue environment. *Journal of Object-Oriented Programming*, 12(2), 14-23.
- Kölling, M. (1999). The problem of teaching object-oriented programming, Part 1: Languages. *Journal of Object-oriented programming*, 11(8), 8-15.

Références bibliographiques

Laal, M., & Ghodsi, S. M. (2012). Benefits of collaborative learning. *Procedia-social and behavioral sciences*, 31, 486-490.

Lafifi, Y. (2007). *SACA: un Système d'Apprentissage Collaboratif*. Thèse de doctorat, Département d'informatique, Université d'Annaba, Algérie.

Livovský, J., & Porubän, J. (2014). Learning object-oriented paradigm by playing computer games: concepts first approach. *Open Computer Science*, 4(3), 171-182.

Mehenaoui, Z. (2018). *Recommandation de collaborateurs pertinents dans un environnement d'apprentissage collaboratif*. Thèse de doctorat, Université du Badji Mokhtar-Annaba, Algérie.

Robins, A., Rountree, J., & Rountree, N. (2003). Learning and teaching programming: A review and discussion. *Computer science education*, 13(2), 137-172.

Rogalski, J. (1998). Concepts et méthodes d'analyse des processus de coopération dans la gestion collective d'environnements dynamiques. *Communications interactives dans les groupes de travail*, 27-58.

Seng, W. Y., Yatim, M. H. M., & Hoe, T. W. (2018). LEARNING OBJECT-ORIENTED PROGRAMMING PARADIGM VIA GAME-BASED LEARNING GAME – PILOT STUDY. *The International journal of Multimedia & Its Applications*. 10. 181-197.

Sims-Knight, J. E., & Upchurch, R. L. (1993). Teaching Object-Oriented Design Without Programming: A Progress Report. *Computer Science Education*, 4(1), 135-156.

Stroustrup, B. (1994). *The design and evolution of C++*. Pearson Education India.

Yan, L. (2009). Teaching object-oriented programming with games. In *2009 Sixth International Conference on Information Technology: New Generations* (pp. 969-974). IEEE.

Sites web consultés

(Href 1) <https://www.dokeos.com/fr/breve-histoire-du-e-learning/> (Consulté le 20/02/2020).

(Href 2) <https://www.kokoroe.fr/list/efficacite-professionnelle/article/plateforme-e-learning-le-b-a-ba-964> (Consulté le 08/02/2020).

(Href 3) <http://cteed.univ-skikda.dz/index.php/architecture-de-plateforme> (Consulté le 17/02/2020).

(Href 4) <https://medium.com/elp-2018/la-programmation-orient%C3%A9e-objet-p-o-o-b3c4b705457> (Consulté le 22/02/2020).

(Href 5) <https://www.codecademy.com/> (Consulté le 23/02/2020).

(Href 6) <https://www.dataone.org/software-tools/rational-rose-0> (Consulté le 17/02/2020).

Références bibliographiques

(Href 7) <https://www.greenfoot.org/door> (Consulté le 24/02/2020).

(Href 8) https://fr.wikipedia.org/wiki/Programmation_orient%C3%A9e_objet (Consulté le 11/03/2020).

(Href 9) https://fr.m.wikipedia.org/wiki/Adobe_Dreamweaver (Consulté le 25/08/2020).

(Href 10) <http://creersonsiteweb.net/page-apprendre-mysql> (Consulté le 25/08/2020).

(Href 11) <https://fr.wikipedia.org/wiki/PHP> (Consulté le 25/08/2020).

(Href 12)

https://developer.mozilla.org/fr/docs/Apprendre/HTML/Introduction_%C3%A0_HTML/Getting_start_ed (Consulté le 17/09/2020)

(Href 13) https://developer.mozilla.org/fr/docs/Learn/JavaScript/First_steps/What_is_JavaScript (Consulté le 25/08/2020).

(Href 14) <https://www.journaldunet.com/web-tech/developpeur/1159810-bootstrap/> (Consulté le 25/08/2020).

(Href 15) <http://glossaire.infowebmaster.fr/css/> (Consulté le 25/08/2020).

(Href 16) <http://www.sthda.com/french/rsthda/paired-t-test.php> (Consulté le 02/10/2020).

Annexes

Annexe 1. L'échelle de Likert

L'échelle de Likert est une échelle de mesure qui comprend en général de 3 à 7 degrés. Elle est très utilisée dans le cadre des enquêtes et des questionnaires. Elle permet d'interroger les individus sur leur degré d'accord ou de désaccord vis-à-vis d'une affirmation. L'échelle de Likert a été développée par Rensis Likert, un psychologue américain, au début des années 1930.

Il est possible de **rajouter des degrés** à l'échelle pour obtenir des **réponses plus nuancées et plus précises**. Par exemple:

- Pas du tout satisfait
- Insatisfait
- Plutôt insatisfait
- Neutre
- Plutôt satisfait
- Satisfait
- Très satisfait

Cette échelle, en raison de sa simplicité et du nombre de choix possible, permet d'**obtenir des réponses très fiables**. C'est la raison première de ce succès.

Annexe 2. Questionnaire de l'étude de cas réalisée

1- Êtes-vous ? *

- Masculin
- Féminin

2- Quelle est votre année universitaire ? *

- 2^{ème} année Licence Informatique
- 3^{ème} année Licence Informatique
- 1^{ère} année Master Informatique
- 2^{ème} année Master Informatique

3- Est-ce que vous avez trouvé des difficultés lors de l'étude de la spécialité informatique ?

- Oui, beaucoup
- Oui, un peu
- Non, pas du tout
- Autre :

4- Quelles sont les matières que vous voyez difficiles ?

- Les matières ayant des TD seulement
- Les matières ayant des TP seulement
- Les matières n'ayant pas des TD et des TP
- Les matières nécessitant des nouveaux langages de programmation

5- Quelles sont les difficultés que vous avez rencontrées avec le module d'algorithmique ?

- Énormément de difficultés
- Beaucoup de difficultés
- Assez de difficultés
- Peu de difficultés
- Aucune difficulté

6- En général, les explications données par l'enseignant (s) dans ce module, étaient-elles claires ?

- Extrêmement claires
- Très claires
- Assez Claires
- Peu claires
- Pas du tout claires

7- Que pensez-vous d'outils d'enseignement (support de cours, exercices) utilisés par l'enseignant (s) dans ce module ?

- Excellent
- Bon
- Sans avis
- Mauvais
- Catastrophique

8- Avez-vous fait des efforts personnels pour comprendre le module d'algorithmique ?

- Totalement Aucun efforts
- Pratiquement pas d'efforts
- Des efforts certains
- Beaucoup d'efforts
- Énormément d'efforts

9- Est-ce que vous avez rencontré des difficultés avec le module de Programmation Orientée Objet (POO) ?

- Énormément de difficultés
- Beaucoup de difficultés
- Assez de difficultés
- Peu de difficultés
- Aucune difficulté

10- En général, les explications données par l'enseignant (s) dans ce module (POO), étaient-elles claires ?

- Extrêmement claires
- Très claires
- Assez Claires
- Peu claires
- Pas du tout claires

11- Que pensez-vous des outils d'enseignement (langue, support cours, exercice) utilisés par l'enseignant (s) dans ce module (POO) ?

- Excellent
- Bon
- Sans avis
- Mauvais
- Catastrophique

12- Que pensez-vous du niveau d'enseignement dans ce module (POO) par rapport à votre niveau ?

- Haut
- Un peu haut
- Convenable
- Un peu bas
- Bas

13- Que pensez-vous du rythme de l'enseignant (s) dans ce module (POO) ?

- Trop rapide
- Un peu trop rapide
- Ni trop rapide ni trop lent
- Un peu trop lent
- Trop lent

14- Que pensez-vous du niveau d'adaptabilité de rythme de l'enseignant (s) à votre niveau de compréhension dans ce module (POO) ?

- Très bien adapté
- Bien adapté
- Un peu adapté
- Pas vraiment adapté

- Pas totalement adapté

15- Que pensez-vous des travaux demandés par l'enseignant (s) dans ce module (POO) ?

- Trop difficile
- Un peu trop difficile
- Ni trop difficile ni trop facile
- Un peu trop facile
- Trop facile

16- Selon vous, la difficulté principale dans ce module (POO), était dans :

- La compréhension de concepts d'orienté objet (Objet, Classe, Héritage...etc.)
- La compréhension d'énoncé de problème posé
- La conception de problème posé
- La formulation de la solution de problème par une vision orientée objet
- Autre

17- Avez-vous fait des efforts personnels pour comprendre le module de POO ?

- Totalement Aucun efforts
- Pratiquement pas d'efforts
- Des efforts certains
- Beaucoup d'efforts
- Énormément d'efforts

18- Que pensez-vous du paradigme (style) orienté objet par rapport à l'impératif procédural :

- Plus difficile et compliqué
- Un peu difficile et compliqué
- Ont le même niveau de difficulté et de complexité
- Un peu facile et non-compliqué
- Trop facile et non-compliqué

19- En général, pensez-vous que l'apprentissage collaboratif (Travailler en groupe) dans l'informatique est efficace ?

- Extrêmement efficace

- Très efficace
- Assez efficace
- Pas très efficace
- Pas du tout efficace

20- Que pensez-vous de l'apprentissage collaboratif (Travailler en groupe) dans le module de POO ?

- Extrêmement efficace
- Très efficace
- Assez efficace
- Pas très efficace
- Pas du tout efficace

21- Selon vous, la meilleure méthode de regroupement dans un travail collaboratif est:

- Selon L'amitié.
- Selon l'enseignant.
- Selon les vœux de l'étudiant.
- Peu importe (au hasard).

22- Selon vous, dans un travail collaboratif dans le module de POO, la bonne procédure est :

- Discuter et faire la conception ensemble, puis chacun implémente seul la solution.
- Faire chacun une conception seul, puis combiner et se mettre d'accord sur la meilleure
- Collaboration totale dans tous les étapes jusqu'à atteindre la solution.
- Autre :.....

23- Connaissez-vous des plateformes d'apprentissage en ligne de la POO ?

- Oui, Citez-les :
- Non

24- Que pensez-vous des plateformes d'apprentissage en ligne de la POO ?

- Extrêmement utiles
- Très utiles
- Assez utiles
- Pas très utiles

- Pas du tout utiles
- Je ne sais pas

25- Comment vous voyez l'intégration des outils des réseaux sociaux dans l'apprentissage de la POO ?

- Extrêmement utiles
- Très utiles
- Assez utiles
- Pas très utiles
- Pas du tout utiles

26- Selon vous, quelle est la fonction principale la plus importante dans une plateforme d'apprentissage en ligne ?

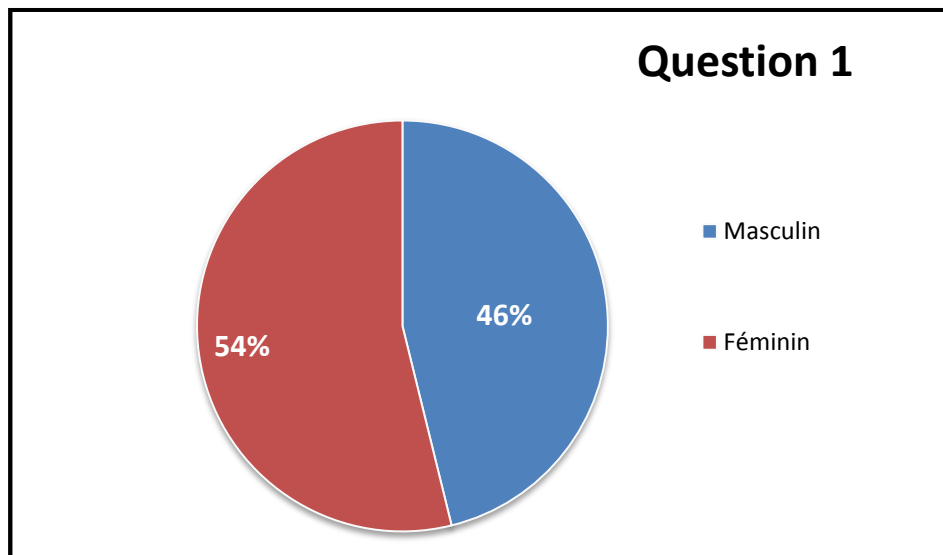
- Adaptabilité du rythme d'apprentissage
- Assurer une bonne collaboration d'apprentissage entre les membres
- Simplicité d'outils d'apprentissage
- Autre, citer la.....

27- Comment vous voyez la meilleure méthode/stratégie de l'apprentissage du module POO ?

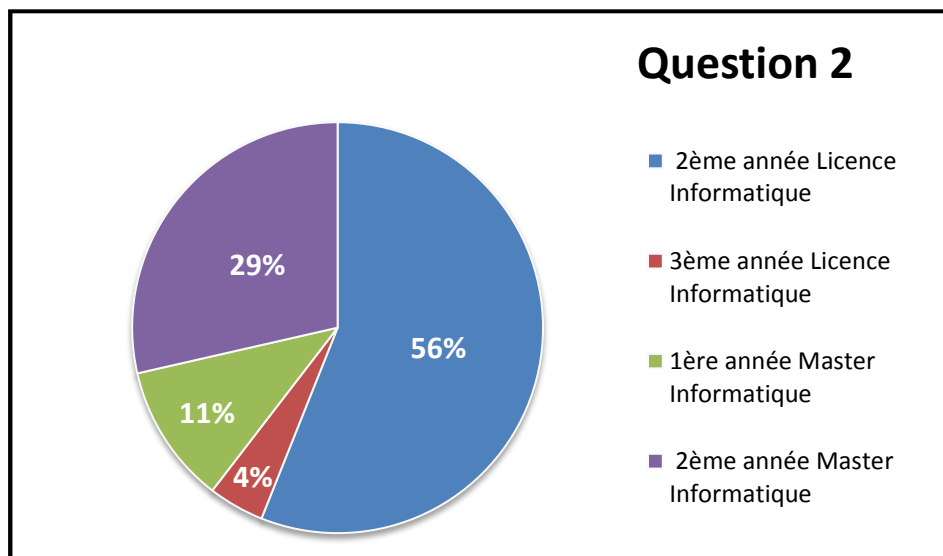
.....
.....

Annexe 3. Etat des résultats de réponses aux 26 questions du questionnaire

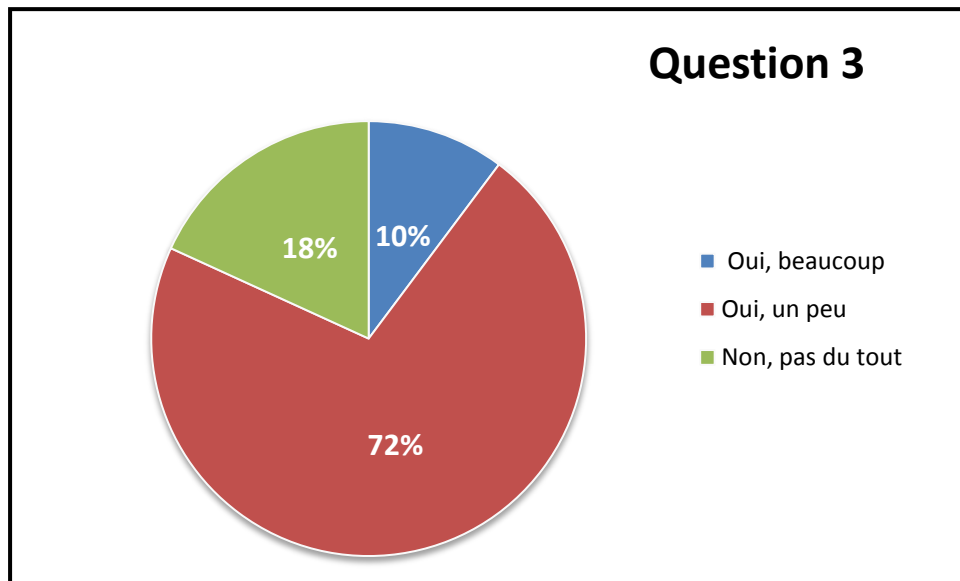
Êtes-vous ?



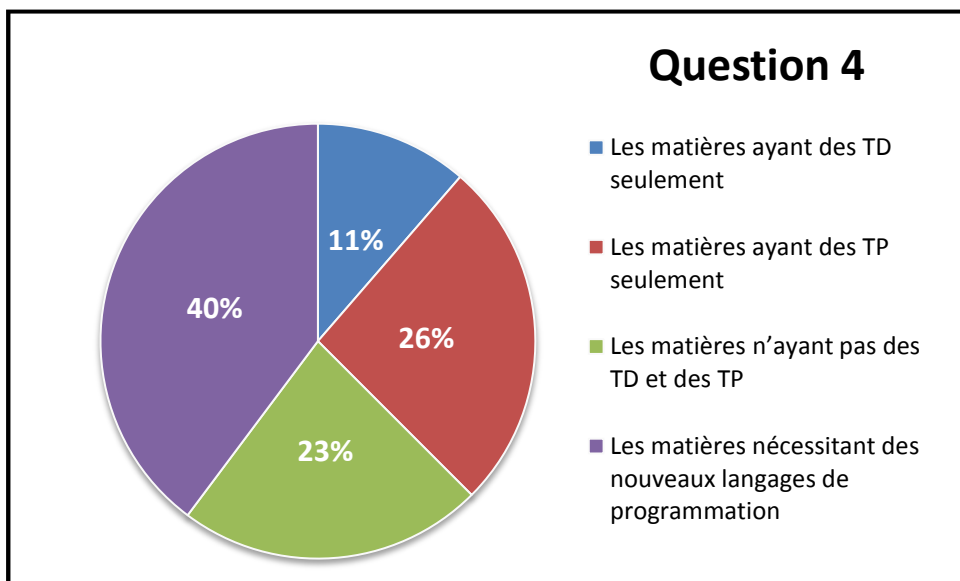
Quelle est votre année universitaire ?



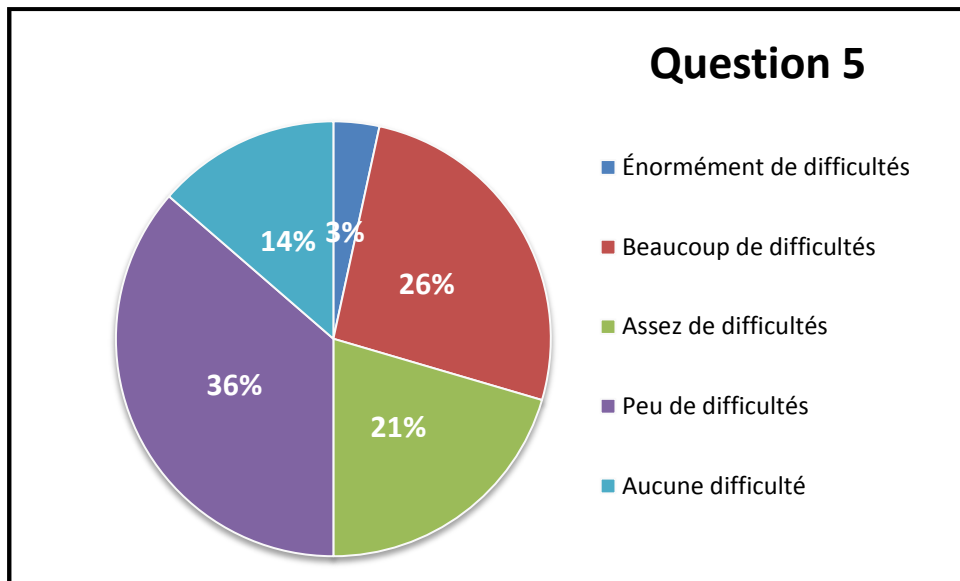
Est-ce que vous avez trouvé des difficultés lors de l'étude de la spécialité informatique ?



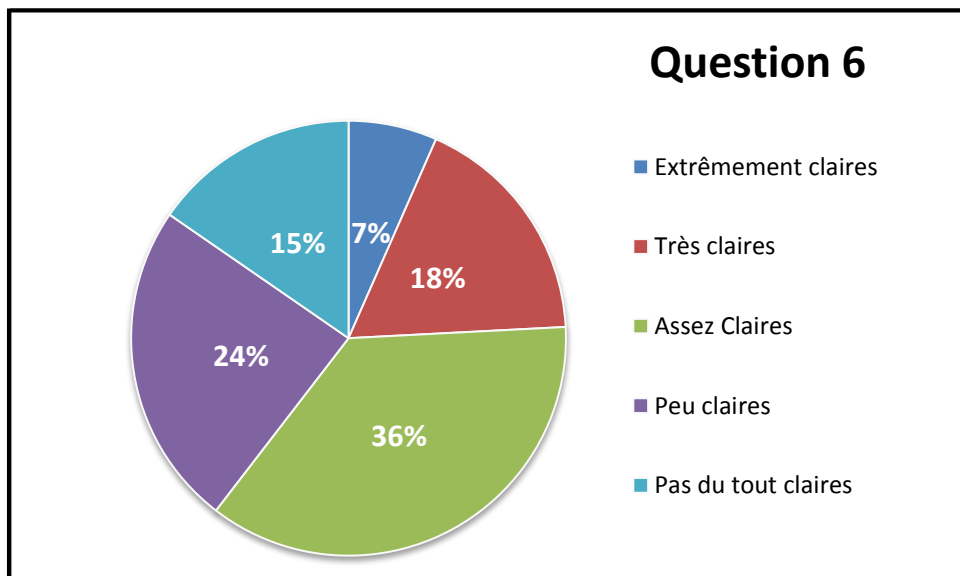
Quelles sont les matières que vous voyez difficiles ?



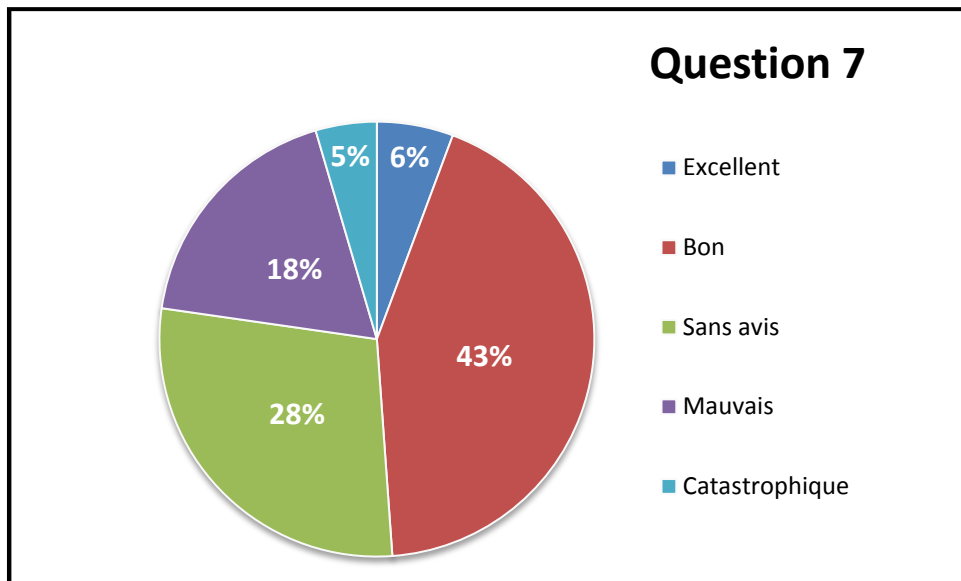
Quelles sont les difficultés que vous avez rencontrées avec le module d'algorithmique ?



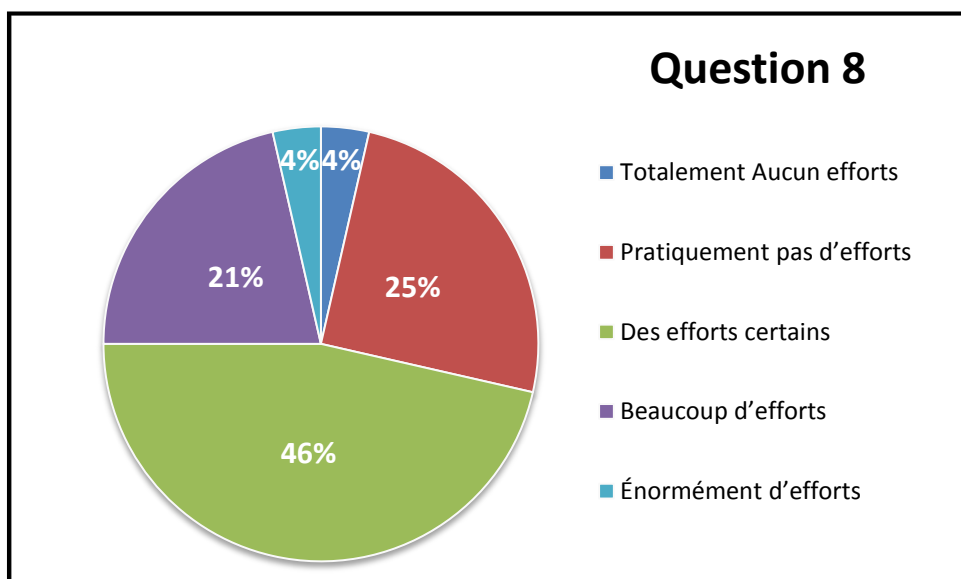
En général, les explications données par l'enseignant (s) dans ce module, étaient-elles claires ?



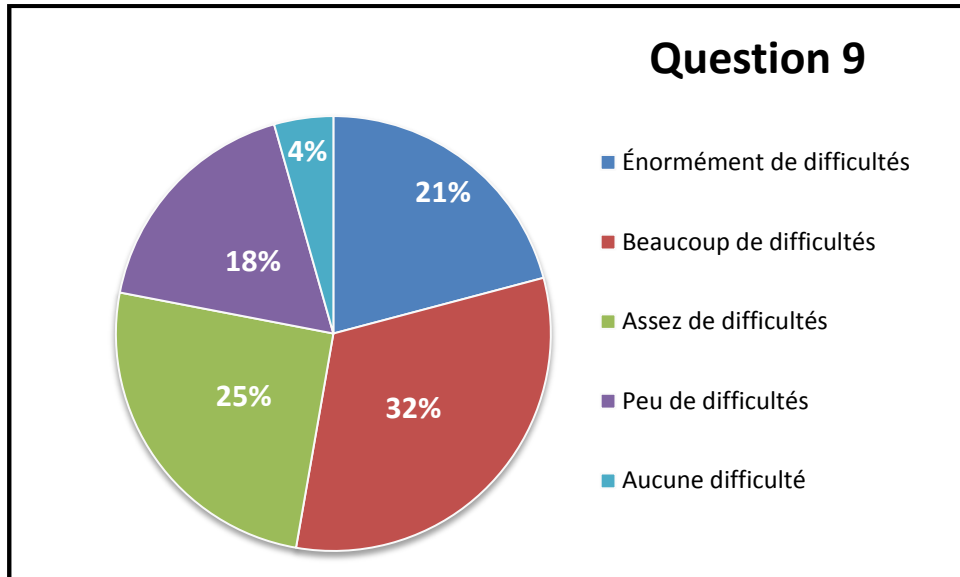
Que pensez-vous d'outils d'enseignement (support de cours, exercices) utilisés par l'enseignant (s) dans ce module ?



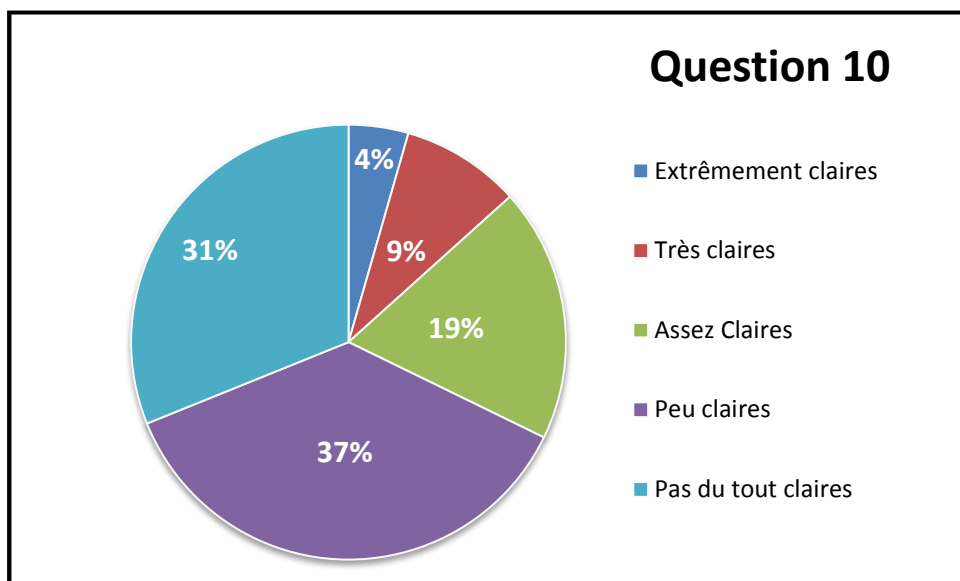
Avez-vous fait des efforts personnels pour comprendre le module d'algorithmique ?



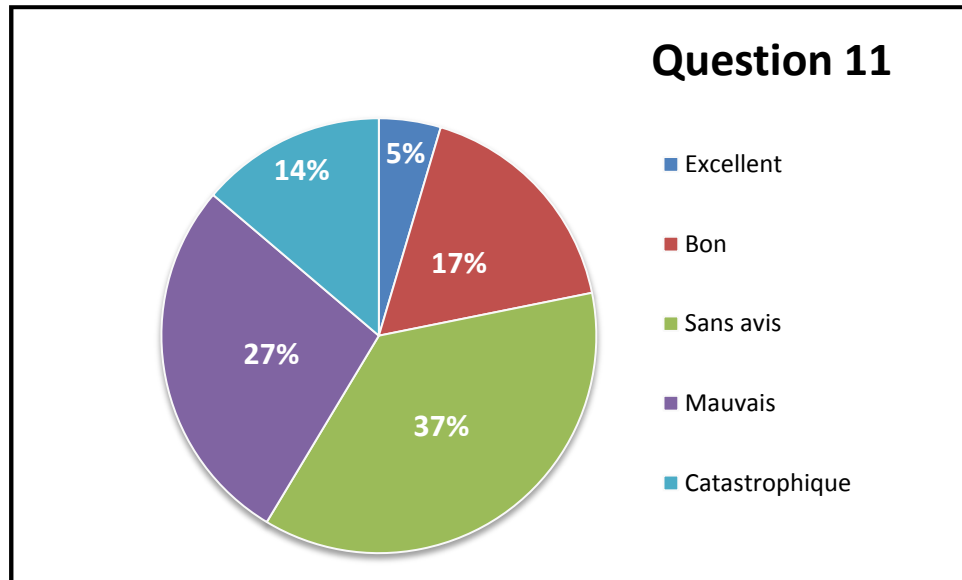
Est-ce que vous avez rencontré des difficultés avec le module de Programmation Orientée Objet (POO) ?



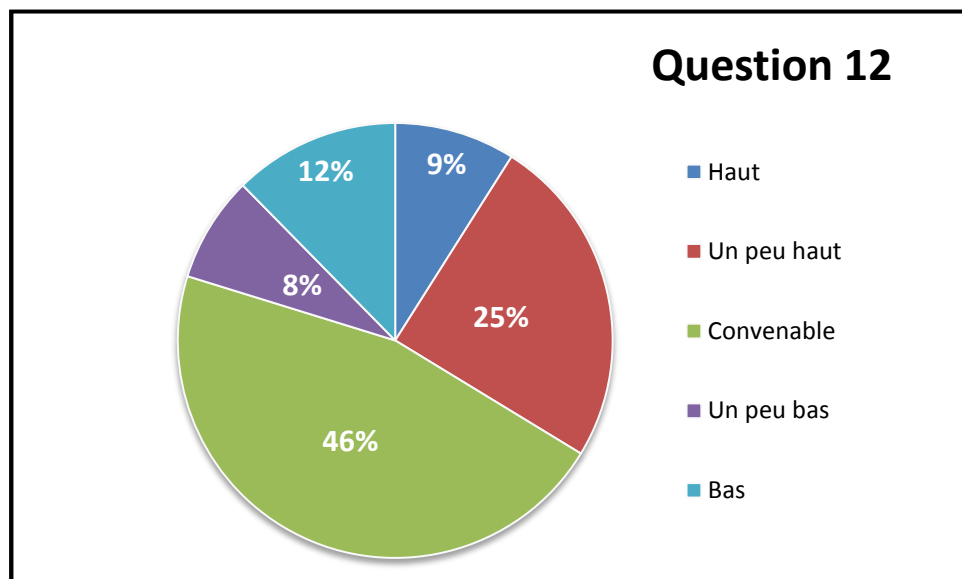
En général, les explications données par l'enseignant (s) dans ce module (POO), étaient-elles claires ?



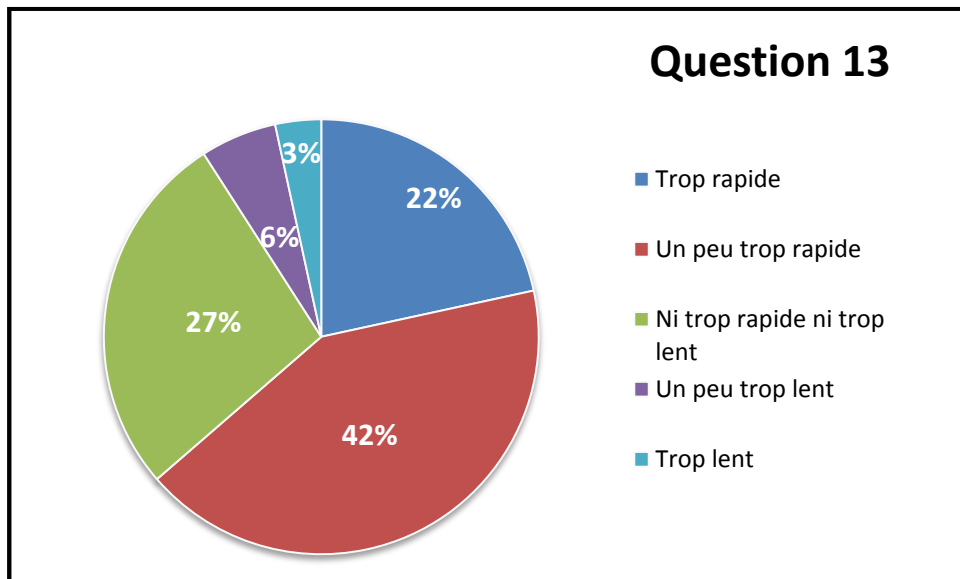
Que pensez-vous des outils d'enseignement (langue, support cours, exercice) utilisés par l'enseignant (s) dans ce module (POO) ?



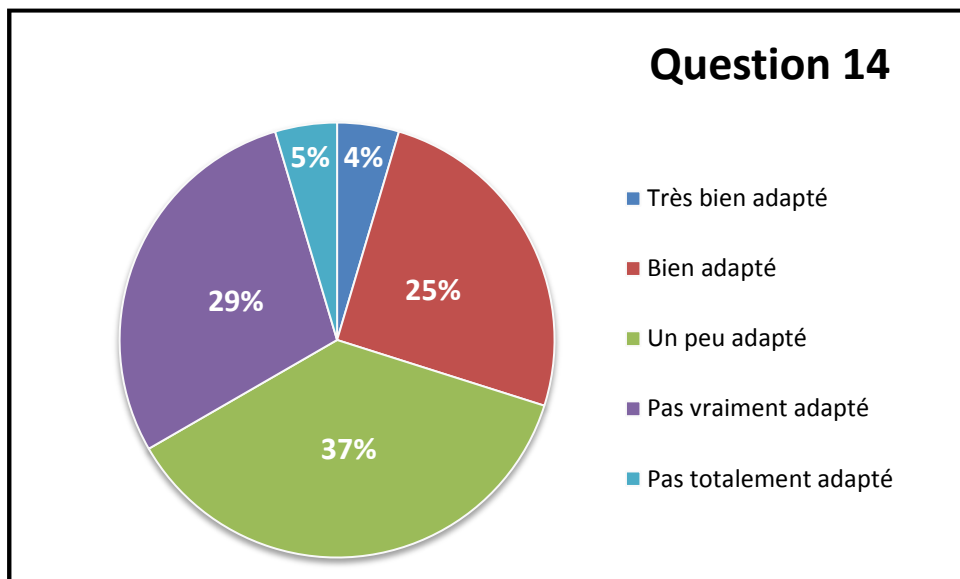
Que pensez-vous du niveau d'enseignement dans ce module (POO) par rapport à votre niveau ?



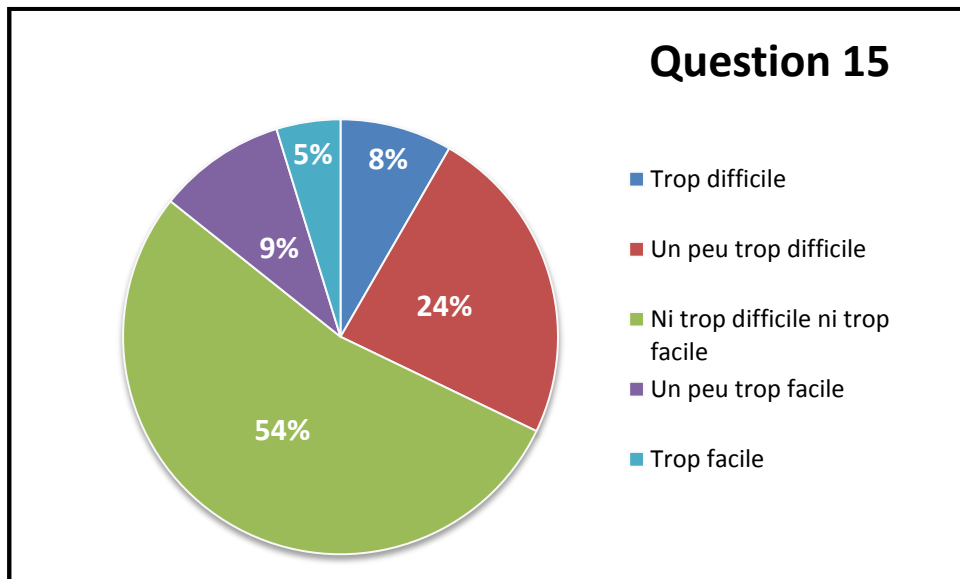
Que pensez-vous du rythme de l'enseignant (s) dans ce module (POO) ?



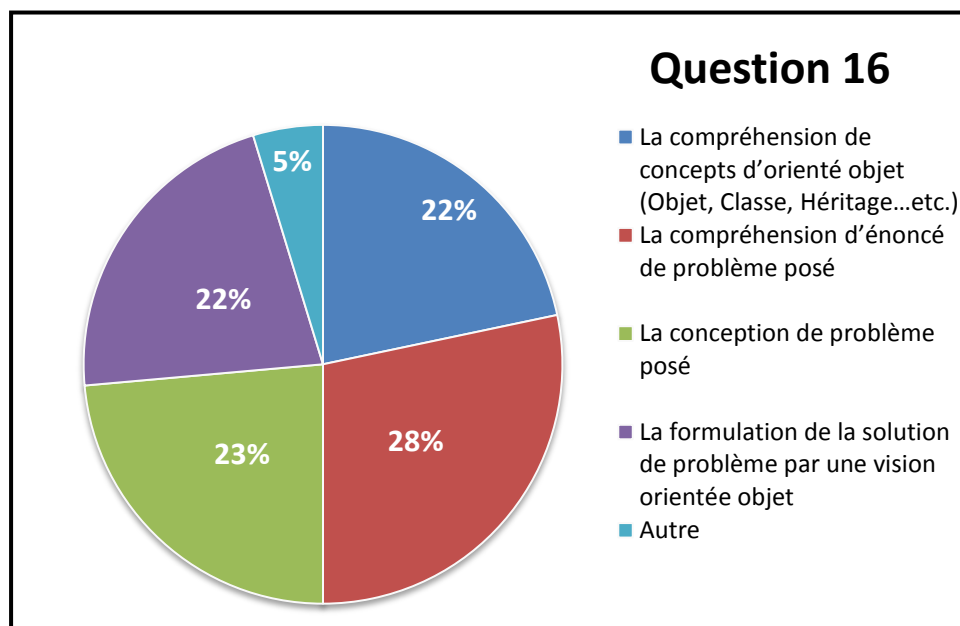
Que pensez-vous du niveau d'adaptabilité de rythme de l'enseignant (s) à votre niveau de compréhension dans ce module (POO) ?



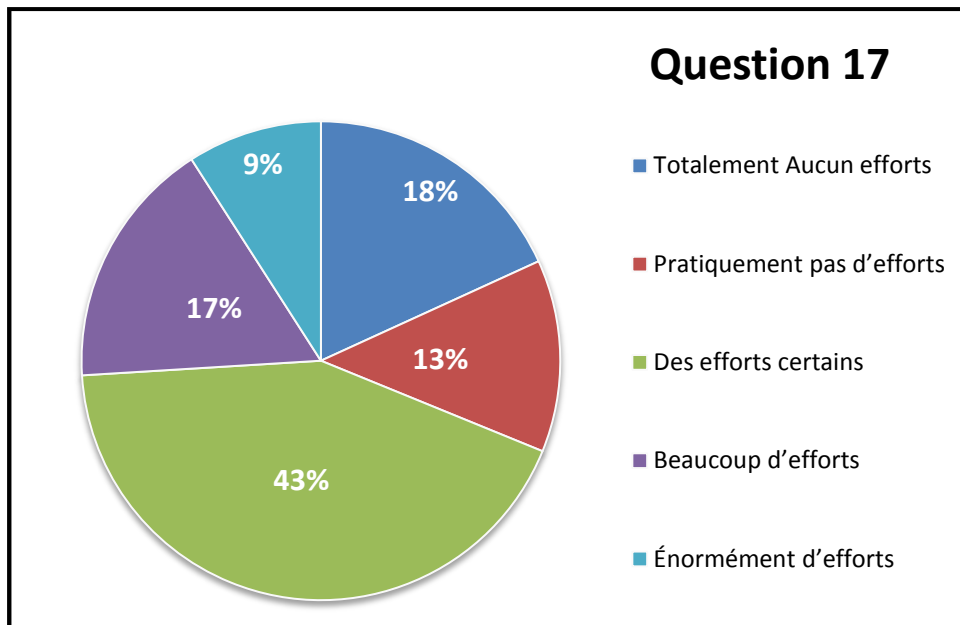
Que pensez-vous des travaux demandés par l'enseignant (s) dans ce module (POO) ?



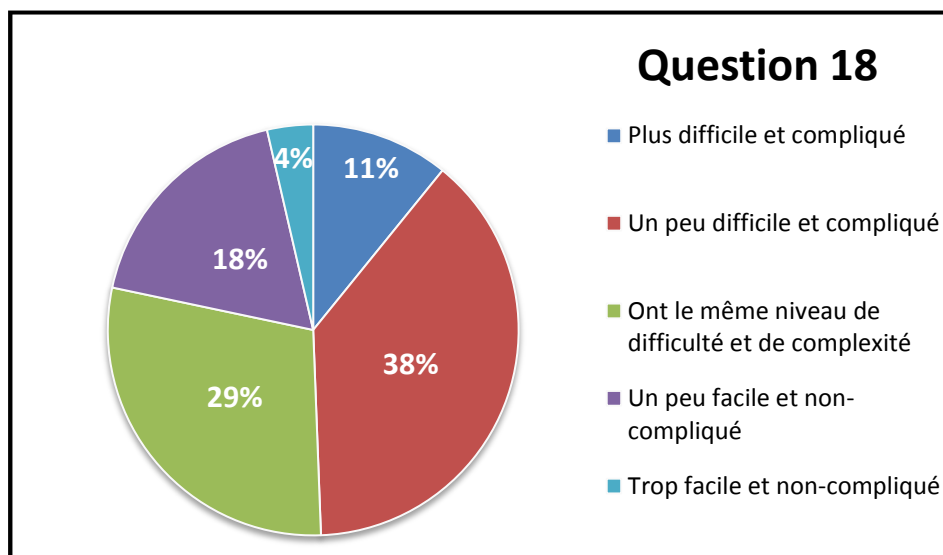
Selon vous, la difficulté principale dans ce module (POO), était dans :



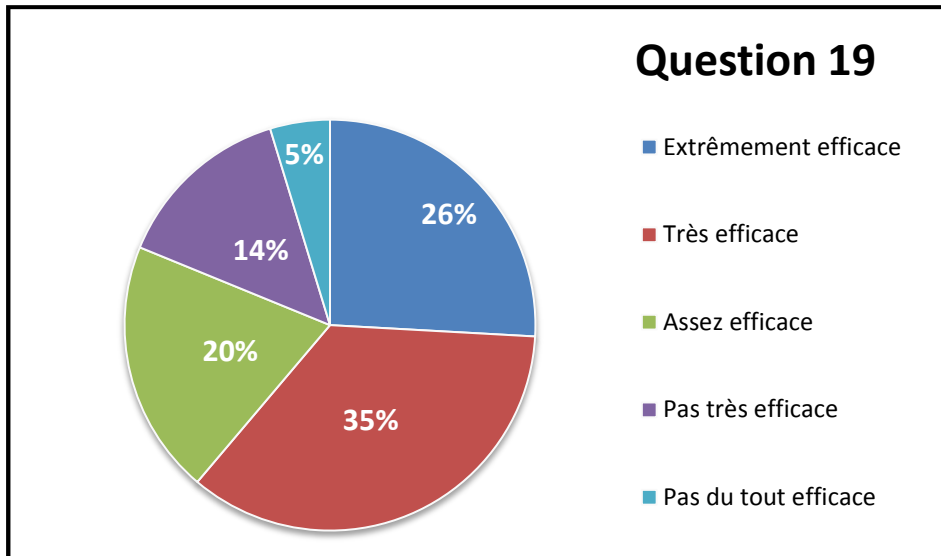
Avez-vous fait des efforts personnels pour comprendre le module de POO ?



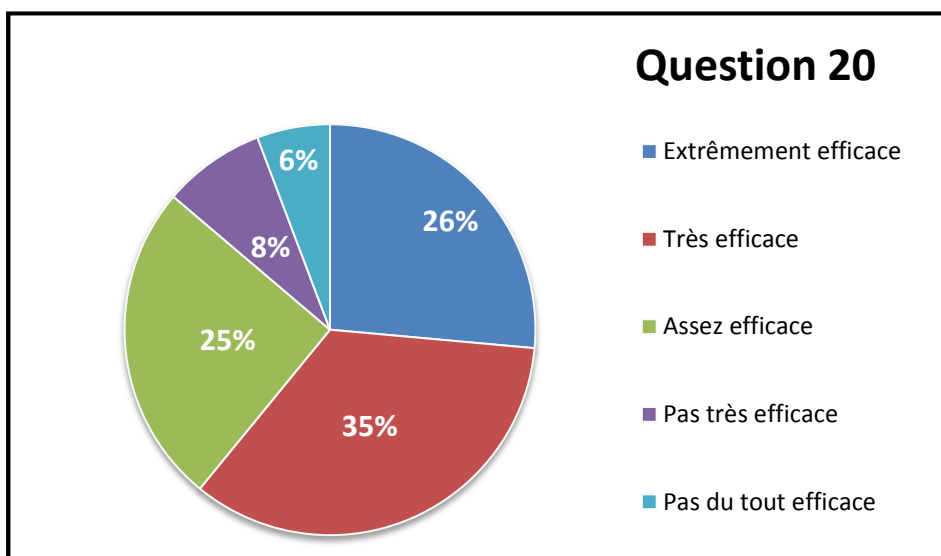
Que pensez-vous du paradigme (style) orienté objet par rapport à l'impératif procédural :



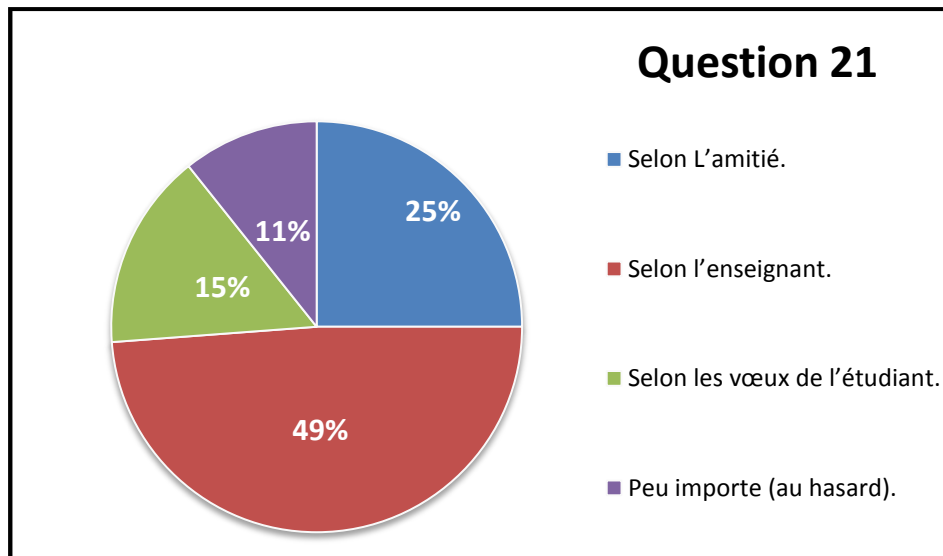
En général, pensez-vous que l'apprentissage collaboratif (Travailler en groupe) dans l'informatique est efficace ?



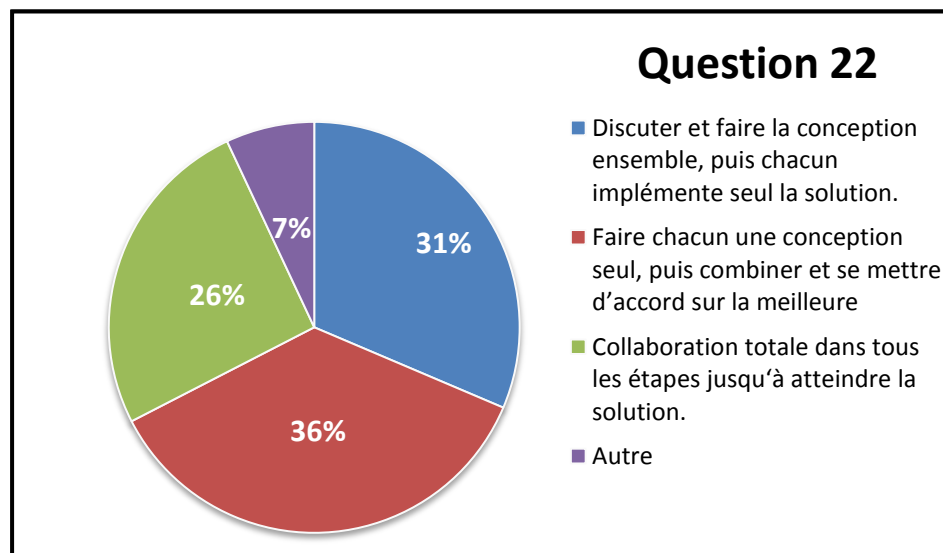
Que pensez-vous de l'apprentissage collaboratif (Travailler en groupe) dans le module de POO ?



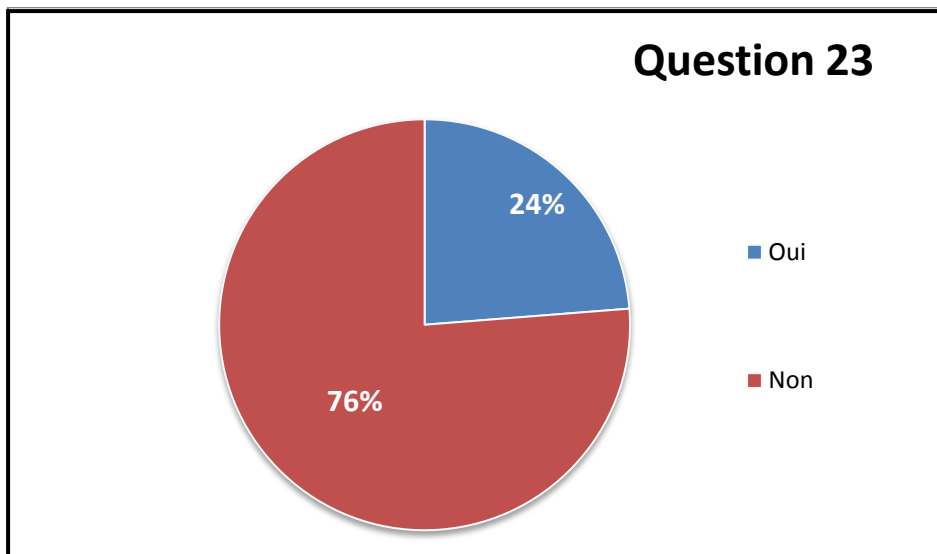
Selon vous, la meilleure méthode de regroupement dans un travail collaboratif est:



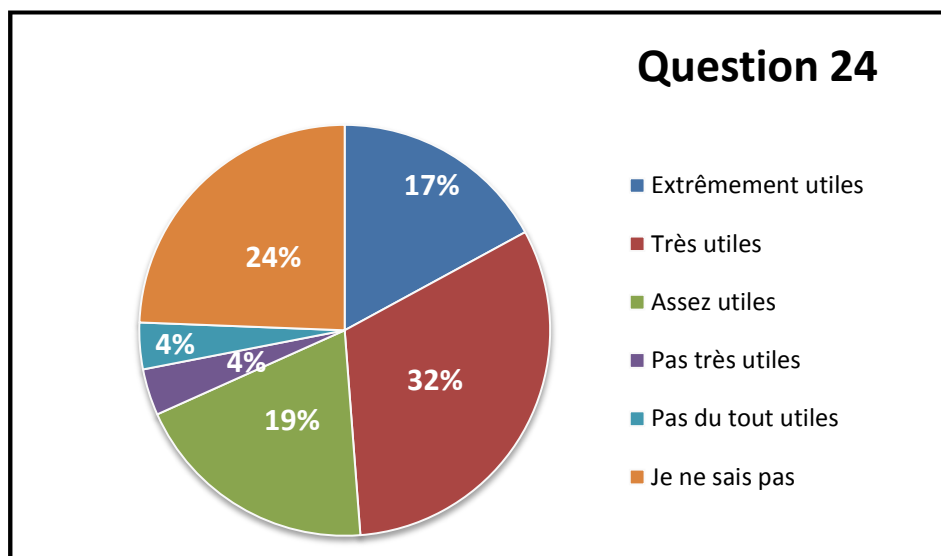
Selon vous, dans un travail collaboratif dans le module de POO, la bonne procédure est :



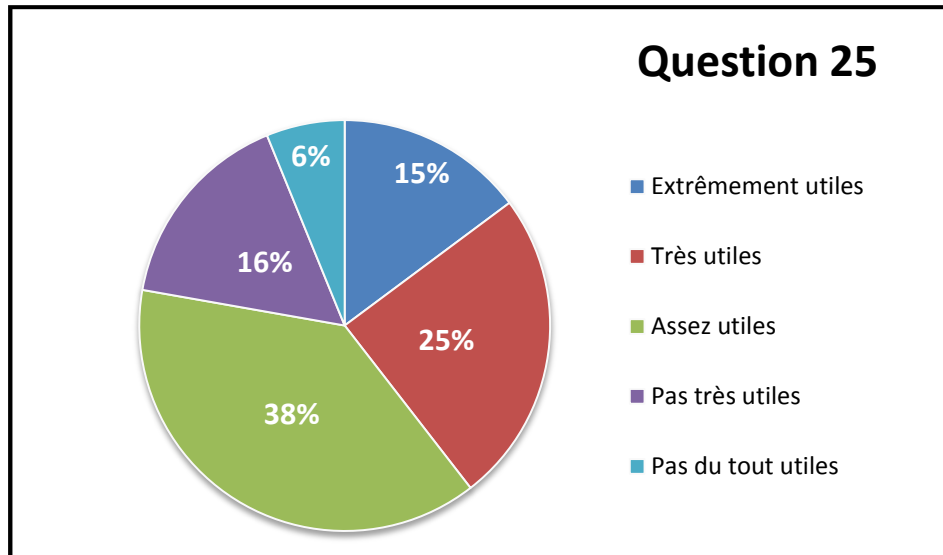
Connaissez-vous des plateformes d'apprentissage en ligne de la POO ?



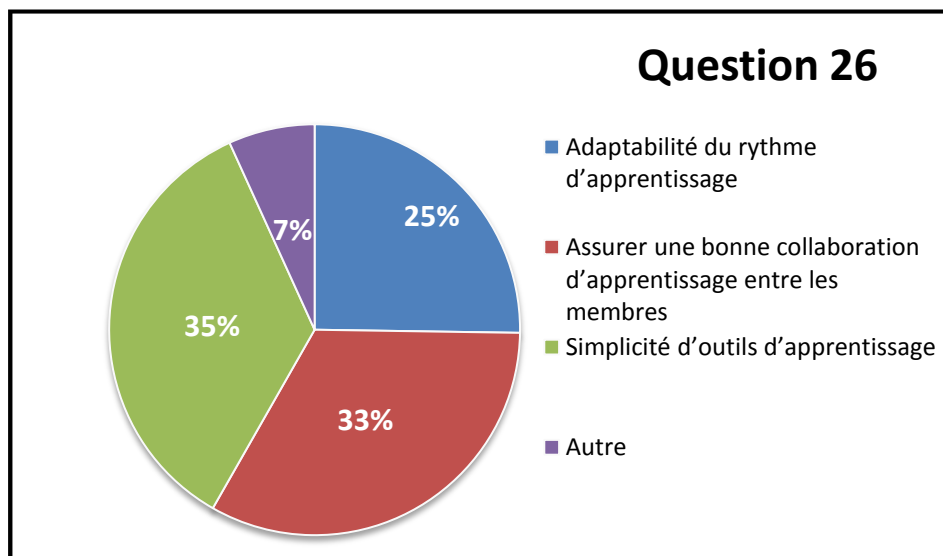
Que pensez-vous des plateformes d'apprentissage en ligne de la POO ?



Comment vous voyez l'intégration des outils des réseaux sociaux dans l'apprentissage de la POO ?



Selon vous, quelle est la fonction principale la plus importante dans une plateforme d'apprentissage en ligne ?



Annexe 4. Autres interfaces et fonctionnalités du système POOLISA

Inscription d'apprenants :

Cette fonctionnalité présentée dans la **Figure**, est destinée aux étudiants pour s'inscrire.

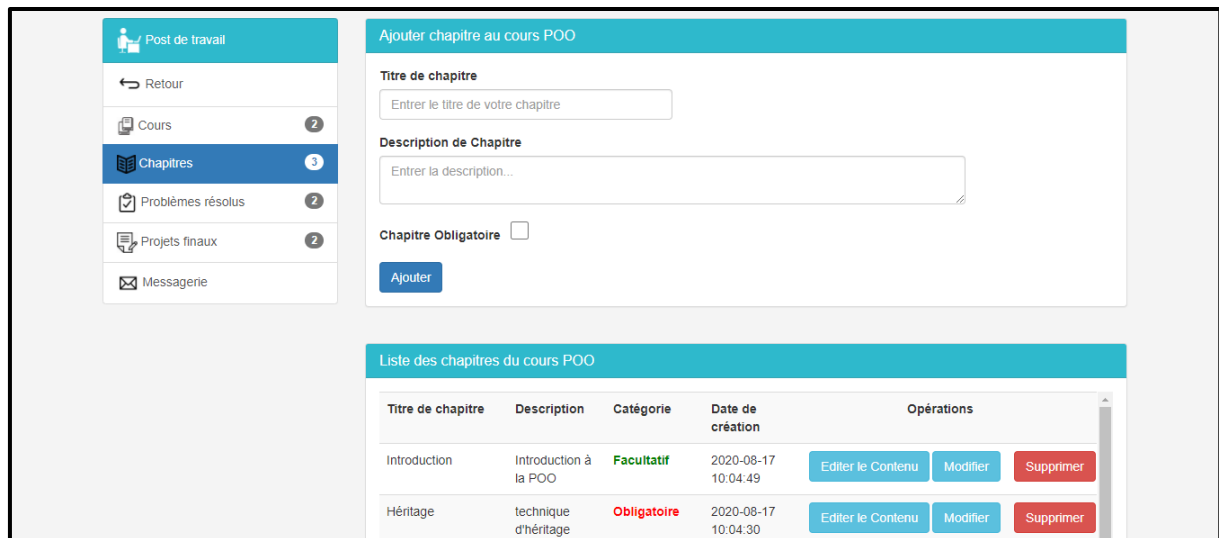
The image shows a registration form titled "Fiche d'inscription". It includes a checkbox for "Si vous êtes étudiant :". Below this are input fields for "Pseudonyme" (containing "Apprenant1"), "Mot de passe" (containing "Entrer votre mot de passe"), "Nom" (containing "ABDERRAHIM"), "Prénom" (containing "Oussama"), and "Adresse-Mail" (containing "abderrahimOussama@gmail.com"). A green "Envoyer" button is at the bottom. To the right, a list of password requirements is shown with red 'X' marks:

- × Une **lettre** minuscule
- × Une **lettre** majuscule
- × Un **numéro**
- × Un minimum de **8 caractères**

Interface *Inscription d'étudiant.*

Ajout de chapitres :

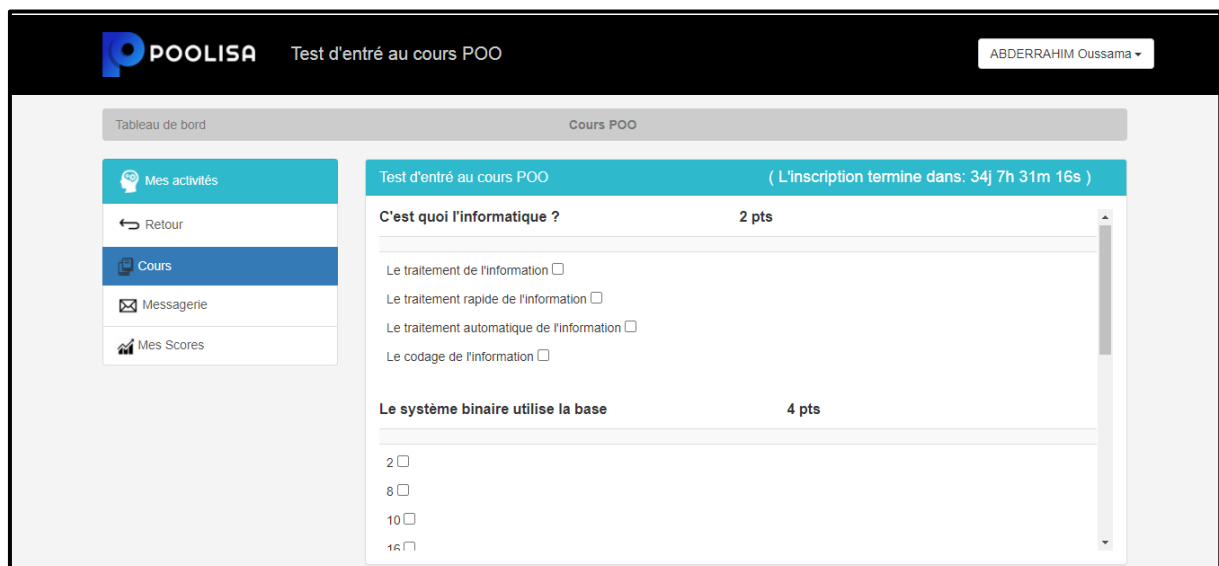
Cette fonctionnalité permet l'ajout de chapitres au cours.



Espace Enseignant - Ajout de chapitre au cours POO.

Inscription au cours :

Cette fonctionnalité permet l'inscription d'étudiant à un cours ajouté, durant une période d'inscription bien définie et limitée.



Espace Etudiant – Test d'entrée au cours POO.



Espace Etudiant – *Inscription échouée au cours.*



Espace Etudiant – *Inscription réussie au cours.*

Annexe 5. Test t de student pour échantillons appariés

Le **test de Student apparié** permet de comparer la moyenne de deux séries de valeurs ayant un lien.

Par exemple, 20 souris ont reçu un traitement X pendant 3 mois. On se pose la question à savoir si le traitement X a un impact sur le poids des souris au bout des 3 mois. Le poids des 20 souris a donc été mesuré avant et après traitement. Ce qui nous donne 20 séries de valeurs avant traitement et 20 autres séries de valeurs après traitement provenant de la mesure du poids des mêmes souris.

Il s'agit bien dans cet exemple, d'un **test de Student apparié** car les deux séries de valeurs ont un lien (les souris). Pour chaque souris, on a deux mesures (l'une avant et l'autre après traitement).

Formule

Pour **comparer les moyennes de deux séries appariées**, on calcule tout d'abord la différence des deux mesures pour chaque paire.

Soit d la série des valeurs correspondant aux différences des mesures entre les paires de valeurs. La moyenne de la différence d est comparée à la valeur 0. S'il y a une différence significative entre les deux **séries appariées**, la moyenne de d devrait être très éloignée de la valeur 0.

La valeur t de Student est donnée par la formule :

$$t = \frac{m}{s/\sqrt{n}}$$

Où : m et s représentent la **moyenne** et **l'écart-type** de la différence d . n est la taille de la série d .

Pour savoir si la différence est significative, il faut tout d'abord lire dans la **table t**, la valeur critique correspondant au **risque alpha** = 5% pour un degré de liberté :

$$d. d. l = n - 1$$