



Référentiel DEV-DATA

RAPPORTS DE PRESENTATION

**Modules :
Algorithm Python Python et structure de données
Poo en Python**

Presente par :

Serigne Modou Diop

Encadre par :

Coach Mbaye

I Avant-propos

La technologie de l'information a révolutionné le monde dans lequel nous vivons. Les données sont devenues un élément clé de notre vie quotidienne, de notre travail et de nos activités sociales. Pourtant, avec l'abondance de données, il est devenu de plus en plus difficile de les gérer efficacement. C'est pourquoi la capacité à collecter, analyser et interpréter les données est devenue une compétence essentielle pour les professionnels de tous les secteurs.

C'est dans cette perspective que j'ai décidé de suivre la formation en développement data à Orange Digital Center / Sonatel Academy. Durant cette formation, j'ai eu l'occasion d'acquérir des connaissances fondamentales sur les algorithmes, Python, la structure de données et la programmation orientée objet en Python. Le but de ce rapport est de présenter les différents modules que j'ai étudiés au cours de cette formation, les objectifs poursuivis, les problèmes rencontrés et les solutions apportées, ainsi que les perspectives pour l'utilisation de ces modules dans la vie professionnelle et académique.

Je tiens à remercier mon coach, Mr MBAYE, pour son soutien et ses conseils tout au long de cette formation, ainsi que tous les anciens apprenants qui ont partagé leur expertise avec nous. Je suis reconnaissant de cette opportunité qui m'a permis d'élargir mes compétences et mes connaissances dans le domaine du développement data.

Enfin, j'espère que ce rapport sera utile pour ceux qui souhaitent découvrir les bases du développement data et qui cherchent à améliorer leurs compétences dans ce domaine.

II Remerciements

Je tiens à exprimer ma gratitude envers toutes les personnes qui ont contribué de près ou de loin à la réalisation de ce rapport.

Je tiens tout d'abord à remercier mon coach, Mr MBAYE, pour sa patience, son accompagnement et ses précieux conseils tout au long de la formation en développement data. Grâce à sa pédagogie, j'ai pu acquérir de nouvelles compétences et renforcer mes connaissances dans le domaine.

Je souhaite également remercier tous les anciens apprenants qui ont partagé leurs connaissances et leur expertise avec nous. Leurs interventions ont été riches en enseignements et m'ont permis de mieux comprendre les différents modules étudiés.

Je n'oublie pas également mes camarades de formation qui ont été d'un grand soutien tout au long de la formation. Leurs encouragements et leur aide ont été essentiels pour surmonter les difficultés rencontrées.

Enfin, je tiens à remercier ma famille et mes amis pour leur soutien indéfectible et leur encouragement tout au long de ce parcours de formation.

III Liste des figures et tableaux

Index des figures

Figure 1: Exemple d'Algorithme de recensement d'information d'un produit.....	14
Figure 2: Exemple d'Algorithme utilisant les structures conditionnelle.....	15
Figure 3: Exemple d'Algorithme utilisant les structures itératives.....	15
Figure 4: Exemple de code python pour supprimer les espaces dans une chaîne.....	16
Figure 5: Exemple de code python pour créer des fonctions (Section 7).....	16
Figure 6: Exemple de code Python pour ouvrir et lire un fichier texte (Section 7).....	17
Figure 7: Exemple de code Python pour créer une classe et ses méthodes.....	17
Figure 8: Exemple de code Python pour créer une classe et ses méthodes puis instancier.....	18

IV Liste des abréviations

• POO	Programmation Orientée Objet
• CSV	Comma-Separated Values (Valeurs Séparées par des Virgules)
• JSON	JavaScript Object Notation (Notation Objet JavaScript)
• XML	Extensible Markup Language
• DEV-DATA	Développent Data
• TIC	Technologies de l'Information et de la Communication
• YAML	Yet Another Markup Language / YAML Ain't Markup Language

V Glossaire

Algorithme : Une série d'instructions ou d'étapes définies pour résoudre un problème ou effectuer une tâche spécifique.

Variables : espaces de stockage temporaire en mémoire, qui permettent de stocker des données, des résultats ou des valeurs intermédiaires lors de l'exécution de l'algorithme.

données en entrée : Ce sont les informations fournies en entrée d'un algorithme, nécessaires pour le traitement.

données en sorties : Ce sont les résultats obtenus à la fin de l'exécution de l'algorithme.

Déclaration : est l'action de spécifier le nom et le type d'une variable qui sera utilisée dans l'algorithme.

Affectation : l'affectation est l'action de donner une valeur à une variable.

Opérateur : symboles ou des mots-clés qui permettent d'effectuer des opérations sur des données ou des variables.

Opérande : sont les valeurs ou les variables sur lesquelles s'appliquent les opérateurs.

Expression : est une combinaison de variables, d'opérateurs et d'opérandes qui permet de calculer une valeur.

instruction d'entrées / sorties : les instructions d'entrées/sorties sont des instructions qui permetteest un langage de balisage qui permet de définir des règles pour structurer les données. Les fichiers XML sont constitués de balises qui encadrent des éléments de données, et qui sont organisées en une structure hiérarchique.nt de communiquer avec l'utilisateur ou avec d'autres périphériques, comme le clavier, la souris ou l'écran.

argument ou paramètre : sont les valeurs qui sont passées à une fonction ou à une procédure pour qu'elle puisse les utiliser dans son traitement.

YAML : YAML est un langage de sérialisation de données utilisé pour stocker et échanger des données structurées. Il est souvent utilisé pour stocker des configurations, des paramètres, des données de test et des données de modèle.

XML : est un langage de balisage qui permet de définir des règles pour structurer les données. Les fichiers XML sont constitués de balises qui encadrent des éléments de données, et qui sont organisées en une structure hiérarchique.

CSV : Le format CSV (Comma-Separated Values) est un format de fichier qui permet de stocker des données sous forme de tableaux, avec les valeurs séparées par des virgules.

Dictionnaire : Une structure de données en Python qui associe des clés à des valeurs.

JSON : Le format JSON (JavaScript Object Notation) est un format de données utilisé pour stocker et échanger des informations structurées.

Liste : Une structure de données en Python qui permet de stocker une collection ordonnée d'éléments.

POO : La programmation orientée objet (POO) est un paradigme de programmation qui utilise des objets pour représenter et manipuler les données et les comportements.

Tuple : Une structure de données en Python similaire à une liste, mais avec des éléments immuables.

Ensemble : Une structure de données en Python qui permet de stocker une collection non ordonnée d'éléments uniques.

Coworking : Le coworking est un modèle de travail qui implique le partage d'un espace de travail et de ressources avec d'autres professionnels indépendants ou entreprises, permettant une collaboration et une communauté dynamique.

Startups : sont des entreprises innovantes en phase de lancement, souvent avec une croissance rapide et des modèles économiques perturbateurs.

Hackathons : événements de programmation collaboratifs qui impliquent généralement la résolution de problèmes techniques ou la création de logiciels en un temps limité.

Data science : La data science est un domaine qui utilise des méthodes statistiques, des algorithmes de machine learning et des outils informatiques pour analyser et comprendre des données complexes.

Classes : modèles de programmation qui permettent de créer des objets avec des attributs et des méthodes communes.

Objets : Sont des instances de classes qui contiennent des attributs et des méthodes spécifiques.

Méthodes : sont des fonctions associées à des objets ou des classes qui leur permettent d'effectuer des actions spécifiques.

Attributs : sont des variables associées à des objets ou des classes qui contiennent des informations spécifiques.

Instancier : L'instanciation en POO est le processus de création d'un objet à partir d'une classe.

Balise : Une balise dans un fichier XML est un élément de syntaxe qui permet de définir la structure et le contenu du fichier.

Librairie "csv" : La librairie "csv" de Python est une bibliothèque standard qui permet de lire et d'écrire des fichiers CSV.

Librairie "json" : La librairie "json" de Python est une bibliothèque standard qui permet de lire et d'écrire des données JSON.

Librairie "xml.etree.ElementTree" : La librairie "xml.etree.ElementTree" de Python est une bibliothèque standard qui permet de manipuler des fichiers XML de manière simple et intuitive.

VI Sommaire

I Avant-propos.....	2
II Remerciements.....	2
III Liste des figures et tableaux.....	3
IV Liste des abréviations.....	3
V Glossaire.....	3
VI Sommaire.....	5
1 INTRODUCTION.....	8
2 PRÉSENTATION.....	8
3 PROBLEMATIQUE.....	9
4 OBJECTIFS.....	11
5 PROBLEMES RENCONTREES SOLUTION APPORTEES.....	11
6 PERSPECTIVES.....	13
7 MAQUETTES D'ECRAN.....	14
8 CONCLUSION.....	18

Table des matières

I Avant-propos.....	2
II Remerciements.....	2
III Liste des figures et tableaux.....	3
IV Liste des abréviations.....	3
V Glossaire.....	3
VI Sommaire.....	5
1 INTRODUCTION.....	8
2 PRÉSENTATION.....	8
2.a Orange Digital Center Sénégal.....	8
2.b Sonatel Academy.....	9
2.c Présentation du référentielle DEV-DATA.....	9
3 PROBLEMATIQUE.....	9
3.a Présentation des modules étudiés.....	9
Algorithme.....	9
Python :.....	10
Structure de données (avec un focus sur les fichiers).....	10
Programmation Orientée Objet.....	10
3.b Les problématiques associées à chaque module.....	10
Algorithme.....	10
Python :.....	10
Structure de données (avec un focus sur les fichiers) :.....	10
Programmation Orientée Objet (POO) :.....	10
4 OBJECTIFS.....	11
Algorithme.....	11
Python.....	11
Structure de données (avec un focus sur les fichiers).....	11
POO.....	11
5 PROBLEMES RENCONTRES SOLUTION APPORTEES.....	11
5.a Les difficultés rencontrées.....	11
Algorithme.....	11
Python :.....	12
Structure de données (avec un focus sur les fichiers).....	12
POO.....	12
5.b Les solutions qui ont été apportées pour surmonter ces difficultés.....	12
Algorithme.....	12
Python.....	12
Structure de données (avec un focus sur les fichiers).....	12
POO.....	13
6 PERSPECTIVES.....	13
7 MAQUETTES D'ECRAN.....	14
8 CONCLUSION.....	18

1 INTRODUCTION

Le développement data est un domaine en pleine expansion, qui joue un rôle crucial dans la transformation numérique des entreprises et des organisations. Pour travailler dans ce secteur, il est essentiel de maîtriser un ensemble de compétences techniques avancées pour traiter, analyser et interpréter les données de manière efficace et précise. Dans ce rapport, nous présentons les modules étudiés au cours, de la première partie de notre formation au sein de l'Orange Digital Center de Dakar, un centres de formation réputés pour leur expertise en matière de développement de compétences numériques en Afrique, en suivant le référentiel de compétences DEV-DATA de la Sonatel Academy notamment l'algorithmique, Python, la structure de données avec un focus sur les fichiers, et la programmation orientée objet en Python

Nous exposons les problématiques associées à chaque module, les objectifs poursuivis dans l'étude de chaque module, les difficultés rencontrées et les solutions apportées pour surmonter ces difficultés. Nous présentons également des perspectives pour l'utilisation de ces modules dans la vie professionnelle et académique. Enfin, nous illustrons les applications développées dans le cadre de l'étude de chaque module grâce à des maquettes d'écran.

L'objectif de ce rapport est de fournir une vue d'ensemble de notre expérience d'apprentissage de ces modules et de montrer comment ces compétences sont importantes pour les professionnels du développement data. Nous allons expliquer comment les modules étudiés dans notre formation nous ont permis de développer ces compétences pour notre future carrière.

2 PRÉSENTATION

2.a Orange Digital Center Sénégal

Orange Digital Center Sénégal est un hub d'innovation et de développement numérique basé à Dakar, au Sénégal. Il a été créé en 2018 par Orange, l'un des principaux opérateurs de télécommunications en Afrique, pour soutenir et accompagner l'écosystème numérique en Afrique de l'Ouest.

Le centre offre une variété de programmes et de services pour les entrepreneurs, les startups, les développeurs, les étudiants et les professionnels de l'industrie. Il dispose d'espaces de coworking, de laboratoires de fabrication, de salles de formation, d'ateliers et d'un programme d'incubation pour aider les startups à développer leurs idées et à se lancer sur le marché.

Le centre organise également des événements, des hackathons, des formations et des conférences pour promouvoir l'innovation et la collaboration entre les acteurs du numérique en Afrique. Il est un lieu de rencontre pour les innovateurs, les investisseurs, les experts et les décideurs de l'industrie numérique.

Orange Digital Center Sénégal fait partie du réseau Orange Digital Center, qui compte plusieurs centres d'innovation en Afrique et en Europe. Il est un symbole de l'engagement d'Orange à soutenir l'innovation et le développement numérique en Afrique, en offrant des opportunités et des outils pour aider les acteurs du numérique à réussir dans un monde de plus en plus numérique.

2.b Sonatel Academy

La Sonatel Academy est une institution de formation professionnelle créée par la Sonatel, une entreprise de télécommunications sénégalaise. Elle a pour objectif de former des jeunes talents dans le domaine des TIC (Technologies de l'Information et de la Communication) et de contribuer au développement des compétences numériques en Afrique.

La Sonatel Academy propose des formations de qualité dans des domaines tels que le développement web, le développement mobile, la sécurité informatique, la data science, etc. Les formations sont dispensées par des experts du domaine, avec une forte orientation pratique pour permettre aux apprenants de développer des compétences opérationnelles en adéquation avec les besoins du marché.

La Sonatel Academy a pour mission de fournir une formation de qualité, de promouvoir l'innovation et d'accompagner les apprenants dans leur insertion professionnelle. Elle est un véritable tremplin pour les jeunes talents qui souhaitent se lancer dans une carrière dans le domaine des TIC.

2.c Présentation du référentielle DEV-DATA

La formation Développement Data de la Sonatel Academy est un programme intensif destiné aux personnes désireuses de développer des compétences en développement de logiciels orientés données. La formation est conçue pour donner aux apprenants les connaissances et les compétences nécessaires pour concevoir, développer et déployer des solutions logicielles basées sur des données.

La formation couvre plusieurs domaines clés du développement de logiciels orientés données, notamment les langages de programmation (Python, R), les bases de données, les technologies de stockage et d'analyse de données, ainsi que les méthodologies de développement de logiciels. Les apprenants sont encadrés par des formateurs expérimentés qui leur fournissent un soutien personnalisé tout au long de la formation. Les apprenants travaillent également sur des projets pratiques pour mettre en pratique les concepts et les compétences qu'ils ont appris.

La formation Développement Data de la Sonatel Academy est une excellente opportunité pour les personnes souhaitant acquérir des compétences en développement de logiciels orientés données et pour les entreprises cherchant à développer des solutions logicielles basées sur des données.

3 PROBLEMATIQUE

3.a Présentation des modules étudiés

Dans le cadre de notre formation en développement data, nous avons eu l'opportunité d'étudier plusieurs modules clés pour notre apprentissage. Voici une brève présentation de chacun de ces modules :

➤ **Algorithme**

Dans ce module, nous avons appris les bases de la conception d'algorithmes pour résoudre différents problèmes informatiques. Nous avons étudié les structures de contrôle telles que les boucles et les conditions, ainsi que les méthodes de tri et de recherche.

➤ **Python :**

Ce module a été consacré à l'apprentissage de la programmation en Python, un langage de programmation polyvalent et populaire. Nous avons étudié la syntaxe de base de Python, la manipulation de chaînes de caractères, ainsi que les fonctions.

➤ **Structure de données (avec un focus sur les fichiers)**

Dans ce module, nous avons appris les différentes structures de données utilisées en Python, y compris les listes, les tuples, les dictionnaires et les set. Nous avons également étudié la gestion des fichiers en Python, y compris la lecture, l'écriture et la manipulation de fichiers.

➤ **Programmation Orientée Objet**

Le module POO (Programmation Orientée Objet) a été consacré à l'apprentissage de la programmation orientée objet en Python. Nous avons étudié les concepts clés de la POO tels que les classes, les objets, les méthodes et les attributs.

Grâce à l'étude de ces différents modules, nous avons acquis des compétences essentielles pour la programmation en Python, la conception d'algorithmes et la gestion des données, qui nous seront utiles dans notre future carrière professionnelle.

3.b Les problématiques associées à chaque module

Chacun des modules étudiés a présenté des problématiques spécifiques que nous avons dû aborder et surmonter. Voici un aperçu des principales problématiques associées à chaque module :

➤ **Algorithme**

La principale problématique associée à ce module a été la conception d'algorithmes pour résoudre des problèmes spécifiques, apprendre à identifier les structures de contrôle appropriées pour chaque situation, ainsi qu'à optimiser nos algorithmes pour une meilleure performance.

➤ **Python :**

L'une des principales problématiques de ce module a été l'apprentissage de la syntaxe de Python, qui peut être très différente de celle d'autres langages de programmation. Nous avons également dû nous familiariser avec les nombreux modules Python disponibles, recréer certains d'entre eux et apprendre à les utiliser pour résoudre des problèmes spécifiques.

➤ **Structure de données (avec un focus sur les fichiers) :**

La principale problématique de ce module a été la manipulation des différentes structures de données en Python. Nous avons dû apprendre à utiliser les listes, les tuples, les dictionnaires et les ensembles de manière efficace pour stocker et manipuler des données. Nous avons également dû apprendre à lire et à écrire des fichiers en Python (CSV, JSON, XML et YAML) sans utiliser les modules python permettant de le faire.

➤ **Programmation Orientée Objet (POO) :**

La principale problématique de ce module a été la compréhension des concepts clés de la POO. Nous avons dû apprendre à concevoir des classes, à créer des objets, à définir des méthodes et des attributs, ainsi qu'à faire des instantiations.

En surmontant ces problématiques, nous avons pu acquérir une compréhension solide de chaque module et des compétences essentielles pour la programmation en Python et la gestion des données.

4 OBJECTIFS

Les objectifs poursuivis dans l'étude de chaque module étaient les suivants :

➤ **Algorithme**

L'objectif principal de ce module était d'apprendre à concevoir des algorithmes efficaces pour résoudre des problèmes spécifiques, identifier les structures de contrôle appropriées pour chaque situation et à optimiser nos algorithmes pour une meilleure performance.

➤ **Python**

L'objectif principal de ce module était de nous familiariser avec la syntaxe de Python et les nombreux modules disponibles pour résoudre des problèmes spécifiques. Nous avons appris à utiliser les structures de contrôle Python, les fonctions, les modules de bibliothèque, les bibliothèques de numérotation et les bibliothèques d'analyse de données.

➤ **Structure de données (avec un focus sur les fichiers)**

L'objectif principal de ce module était de nous familiariser avec les différentes structures de données en Python. Nous avons appris à utiliser les listes, les tuples, les dictionnaires et les ensembles de manière efficace pour stocker et manipuler des données. Nous avons également appris à lire et à écrire des fichiers en Python.

➤ **POO**

L'objectif principal de ce module était de nous familiariser avec les concepts clés de la POO. Nous avons appris à concevoir des classes, à créer des objets, à définir des méthodes et des attributs.

En atteignant ces objectifs, nous avons acquis une compréhension solide de chaque module et des compétences essentielles pour la programmation en Python et la gestion des données.

5 PROBLEMES RENCONTREES SOLUTION APPORTEES

5.a Les difficultés rencontrées

Lors de l'étude de chaque module, nous avons rencontré certaines difficultés, notamment :

➤ **Algorithme**

L'un des principaux défis de l'étude des algorithmes était de concevoir des solutions pour des problèmes complexes. Nous avons dû apprendre à décomposer les problèmes en sous-problèmes plus petits et à utiliser des techniques de conception d'algorithmes pour résoudre chaque sous-problème. Parfois, le choix de la structure de contrôle appropriée pour chaque situation était également difficile.

➤ **Python :**

L'un des principaux défis de l'étude de Python était la maîtrise de la syntaxe de Python et de ses nombreux modules. La syntaxe de Python peut être différente de celle d'autres langages de programmation, ce qui peut être déroutant pour les nouveaux utilisateurs. En outre, il y a tellement de modules Python disponibles que la tâche de choisir le bon module pour résoudre un problème donné peut être difficile.

➤ **Structure de données (avec un focus sur les fichiers)**

L'un des principaux défis de l'étude de la structure de données était la compréhension de la manière de stocker et de manipuler efficacement des données en Python. Les différentes structures de données ont des avantages et des inconvénients, et choisir la bonne structure de données pour chaque situation peut être difficile. En outre, la lecture et l'écriture de fichiers peuvent être complexes pour les débutants. Par exemple, lors de la lecture d'un fichier CSV, il peut y avoir des problèmes de séparateurs de colonnes ou de lignes. La manipulation de grandes quantités de données peut être difficile à gérer en raison de la complexité des formats de fichiers. Par exemple, lors de l'analyse de données en JSON, il peut être difficile de naviguer dans les données imbriquées. Lors de la lecture d'un fichier XML, il peut être difficile de gérer les balises et les attributs.

➤ **POO**

L'un des principaux défis de l'étude de la POO était la compréhension de la manière de concevoir des classes et des objets efficaces. Il peut être difficile de choisir les bons attributs et méthodes pour une classe donnée et de les organiser de manière cohérente.

5.b Les solutions qui ont été apportées pour surmonter ces difficultés

Pour surmonter les difficultés rencontrées lors de l'étude de chaque module, nous avons adopté différentes stratégies, notamment :

➤ **Algorithme**

Pour surmonter les difficultés liées à la conception d'algorithmes, nous avons utilisé des approches systématiques telles que la décomposition des problèmes en sous-problèmes plus petits et la modélisation de chaque sous-problème sous forme d'algorithme. Nous avons également travaillé en groupes pour résoudre les problèmes et discuter des différentes approches de conception d'algorithmes.

➤ **Python**

Pour surmonter les difficultés liées à la syntaxe de Python et à la maîtrise de ses nombreux modules, nous avons utilisé des ressources en ligne telles que des tutoriels Python et des documentations de module. Nous avons également travaillé sur des projets pratiques pour appliquer les connaissances acquises et avons travaillé en groupes pour résoudre les problèmes et discuter des différentes approches de résolution de problèmes en Python.

➤ **Structure de données (avec un focus sur les fichiers)**

Pour surmonter les difficultés liées à la manipulation efficace des données en Python, nous avons utilisé des structures de données appropriées pour chaque situation et avons travaillé sur des projets pratiques pour appliquer les connaissances acquises. Nous avons également utilisé des ressources en ligne telles que des tutoriels et des documentations pour apprendre à lire et à écrire des fichiers en Python.

Pour les fichiers CSV : l'utilisation de la librairie "csv" de Python permet de lire et d'écrire des fichiers CSV de manière efficace. La librairie permet également de définir des délimiteurs et des caractères d'échappement personnalisés pour mieux gérer les fichiers avec des données complexes.

Pour les fichiers JSON : Python propose une librairie intégrée appelée "json" pour traiter les fichiers JSON. Cette librairie permet de charger les données JSON en tant qu'objets Python et vice

versa. Elle fournit également des fonctions pour la manipulation de données JSON, telles que la création de nouveaux objets JSON, la modification d'objets JSON existants et la suppression de données JSON.

Pour les fichiers XML : la librairie "xml.etree.ElementTree" de Python permet de lire et de traiter les fichiers XML de manière simple et efficace. Elle permet de parcourir les éléments et les attributs du fichier XML et d'accéder aux données qu'ils contiennent. on a du créer nos propres fonctions pour créer les balises ce qui nous a permis de comprendre comment fonctionne la librairie "xml.etree.ElementTree"

➤ **POO**

Pour surmonter les difficultés liées à la conception de classes et d'objets efficaces, nous avons demandé de l'aide à nos camarades et parfois à des communautés en ligne. Nous avons également travaillé en groupes pour discuter des différentes approches de conception de classes et d'objets et pour résoudre les problèmes.

En travaillant dur, en sollicitant l'aide de nos formateurs et de nos pairs et en utilisant ces différentes stratégies, nous avons pu surmonter les difficultés rencontrées lors de l'étude de chaque module et atteindre nos objectifs d'apprentissage.

6 PERSPECTIVES

Les modules étudiés lors de cette formation en développement data offrent des perspectives intéressantes pour leur utilisation future dans la vie professionnelle et académique.

Tout d'abord, la maîtrise de l'algorithme est un atout majeur dans le domaine du développement de logiciels, car elle permet d'optimiser les processus et les performances des programmes. La connaissance de Python est également très demandée dans les métiers de la data, notamment pour la manipulation de données, l'analyse statistique et la visualisation de données.

La compréhension de la structure de données, en particulier les fichiers, est également très utile pour le traitement et la gestion de données. De même, la maîtrise de la programmation orientée objet en Python offre de nombreuses possibilités pour la création de programmes modulaires et évolutifs.

En ce qui concerne la POO, elle est utilisée dans de nombreux domaines, notamment dans le développement de logiciels d'entreprise, de jeux vidéo, de sites Web et d'applications mobiles. Elle permet de structurer efficacement le code, de faciliter la maintenance et la mise à jour des applications, et de réduire le temps et les coûts de développement. Dans le domaine académique, la POO est enseignée dans de nombreuses formations en informatique et en génie logiciel. Elle permet aux étudiants de comprendre les concepts avancés de la programmation et de développer des compétences en conception de logiciels.

Dans l'ensemble, les modules étudiés offrent des perspectives intéressantes pour leur utilisation future dans le domaine de la data, mais également dans d'autres domaines tels que le développement de logiciels, l'analyse de données, la modélisation de systèmes, etc.

7 MAQUETTES D'ECRAN

Présentation de maquettes d'écran illustrant les applications développées dans le cadre de l'étude de chaque module

Module Algorithmes

```
1
2  Algorithme Information Produit
3  Constantes
4  |   tva : entier
5  Variables
6  |   libelle : chaine
7  |   stock, prixuni, mstock, mttc : réel
8  Debut
9  |   Ecrire("donner le libellé du produit ")
10 |   Lire(libelle)
11 |   Ecrire("donner sa quantité en stock")
12 |   Lire(stock)
13 |   Ecrire("donner son prix unitaire")
14 |   Lire(prixuni)
15 |   mstock = prix unitaire * stock
16 |   mttc = mStock + mStock*tva
17 |   Ecrire("Le montant en stock du produit",libelle,"est",mstock)
18 |   Ecrire("Le montant TTC est ",mttc)
19 Fin Algorithme
20
21 -- Expression
22 -- variables, constantes données en entrée données en sorties
23 -- déclaration affectation opérateur opérande expression
24 -- instruction d'entrées / sorties argument ou paramètre
```

Figure 1: Exemple d'Algorithme de recensement d'information d'un produit

```
1
2  Algorithme Resolution
3  Variables
4  |   a,b,x : reel
5  Debut
6  |   Ecrire("Saisir la Valeur de a ")
7  |   Lire(a)
8  |   Ecrire("Saisir la Valeur de b ")
9  |   Lire(b)
10 |   Si(a = 0)
11 |   |   Ecrire("le système n'admet pas de solution")
12 |   Sinon
13 |   |   x = -(b/a)
14 |   |   Ecrire("la solution du système est x = ",x)
15 |   Fin Si
16 Fin Algorithme
17
18
19
20
21
22
```

Figure 2: Exemple d'Algorithme utilisant les structures conditionnelle

```
1
2  Algorithme somme
3  variables
4  |   nbr,som,i : entier
5  debut
6  |   ecrire("donner un entier")
7  |   lire(nbr)
8  |   i = 0
9  |   tant que (i <= nbr)
10 |   |   som = som + i
11 |   |   i = i + 1
12 |   fin tant que
13 |   Ecrire("la somme est ",som)
14 Fin
15
16
17
18
19
20
21
22
```

Figure 3: Exemple d'Algorithme utilisant les structures itératives

Module Python

```

1
2  chaine=str(input("donner une chaine1: "))
3  chaine_sans_espace = ""
4
5  chaine1 = chaine_sans_espace + chaine[0]
6  for i in range (1,len(chaine)):
7      |   |   |   if (chaine[i-1] == " " and chaine[i] == " ") :
8      |   |   |       continue
9      |   |   |   else :
10     |   |   |       chaine_sans_espace = chaine_sans_espace + chaine[i]
11     print(chaine_sans_espace)
12
13
14
15
16

```

Figure 4: Exemple de code python pour supprimer les espaces dans une chaîne

```

188
189
190  def changer_format_classe(classe):
191      |
192      |   format_classe = ""
193      |
194      |   for i in range(len(classe)):
195      |       |   if '3' <= classe[i] <= '6' or 'A' <= classe[i] <= 'B':
196      |       |       format_classe += classe[i]
197      |   return format_classe
198
199
200
201  def is_valid_classe(classe):
202      |   if len(classe) != 2:
203      |       return False
204      |   if not classe[0].isdigit():
205      |       return False
206      |   if classe[1] not in ['A', 'B']:
207      |       return False
208      |   if int(classe[0]) < 3 or int(classe[0]) > 6:
209      |       return False
210      |   return True
211
212

```

Figure 5: Exemple de code python pour créer des fonctions

Module Structure de Donnees (Fichier)

```
1
2 from fonctions import *
3 import csv
4
5
6 #lire le fichier csv
7 filename = open('Donnees_Projet_Python_DataC5.csv')
8 f = csv.reader(filename)
9
10 lignecompte = 0
11 dico = {}
12 donnee_invalides = {}
13 donnees_valides = {}
14 donne = {}
15
16 #mettre les donnees dans un dictionnaire
17 for ligne in f :
18     if lignecompte >= 1 :
19         dico[ligne[1]] = {'code':ligne[0],
20                          'numero':ligne[1],
21                          'nom':ligne[2], 'prenom':ligne[3],
22                          'date_de_naissance':transformation_date(transformation_des_separateurs_en_tiret(espace_inutile(ligne[4]))),
23                          'classe':changer_format_classe(ligne[5]),
24                          'note': format_notes(ligne[6])
25                          }
26         lignecompte += 1
27
28
29 donnees_valides,donnees_invalides = separation_des_donnees_en_valide_et_invalide(dico)
30
31
32
```

Figure 6: Exemple de code Python pour ouvrir et lire un fichier texte

Module POO Python

```
6
7 class Note:
8     def __init__(self,note_c):
9         self.note = note_c
10
11
12 > def format_notes(self, notes): ...
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86 class Etudiants :
87     def __init__(self,code,numero,nom,prenom,date,classe,note):
88         self.nom = nom
89         self.prenom = prenom
90         self.note = note
91         self.numero = numero
92         self.date = date
93         self.code = code
94         self.classe = classe
95
96     def __str__(self) :
97         return f"{self.code},{self.numero},{self.nom},{self.prenom},{self.date},{self.classe},{self.note}"
98
99
```

Figure 7: Exemple de code Python pour créer une classe et ses méthodes


```

class Fichier:
    def __init__(self) :
        pass

    def lire_fichier(self,fichiercsv) :
        filename = open(fichiercsv)
        lignecompte = 0
        f = csv.reader(filename)

        dico = {}
        for ligne in f :
            if lignecompte >= 1 :
                dico[ligne[1]] = {'code':ligne[0], 'numero':ligne[1],
                                   'nom':ligne[2], 'prenom':ligne[3],
                                   'date_de_naissance':ligne[4],
                                   'classe':(ligne[5]), 'note': (ligne[6])}
                lignecompte += 1

        donne = {}
        for cle,valeur in dico.items() :
            Etudiant = Etudiants(valeur["code"],valeur["numero"],
                                   valeur["nom"],valeur["prenom"],
                                   valeur["date_de_naissance"],
                                   valeur["classe"],
                                   Note.format_notes(self,valeur["note"]))
            etu = {"code":Etudiant.code, "numero":Etudiant.numero, "nom":Etudiant.nom,
                  'prenom':Etudiant.prenom, "date":Etudiant.date,
                  "classe":Etudiant.classe, "note":Etudiant.note}
            donne[valeur["numero"]] = etu
        return donne

class principale :
    fichier = Fichier()
    m = fichier.lire_fichier('Donnees_Projet_Python_DataC5.csv')
    # fichier.afficher_info()

```

Figure 8: Exemple de code Python pour créer une classe et ses méthodes puis instancier

8 CONCLUSION

Synthèse des éléments clés présentés dans le rapport Perspectives pour l'utilisation de ces modules dans la vie professionnelle et académique

En conclusion, ce rapport a présenté les modules étudiés dans le cadre de la formation "Développement Data". Nous avons examiné les principales problématiques associées à chaque module, les objectifs poursuivis et les difficultés rencontrées lors de leur étude. Nous avons également discuté des solutions apportées pour surmonter ces difficultés et des perspectives d'utilisation de ces modules dans la vie professionnelle et académique.

Le module d'algorithmie a été utile pour comprendre les principes fondamentaux de la programmation et pour améliorer la logique de résolution de problèmes. Le module Python a fourni une base solide pour la programmation orientée objet, l'analyse de données et la création d'applications. Le module Python et structure de données, avec un focus sur les fichiers, a aidé à comprendre comment stocker et manipuler des données, et à gérer des fichiers pour stocker de grandes quantités de données. Le module POO en Python a permis de comprendre les concepts avancés de la programmation orientée objet et de la modélisation des classes et des objets.

Ces modules ont une grande importance dans la vie professionnelle et académique, car ils offrent des compétences essentielles pour les développeurs de données, les analystes de données et les ingénieurs logiciels. Ils permettent également de créer des applications efficaces pour résoudre des problèmes réels.

En conclusion, les modules étudiés dans cette formation ont été très utiles et ont fourni une base solide pour les compétences en développement de données. Ils ont ouvert la voie à de nombreuses perspectives futures et ont été une expérience précieuse pour notre développement professionnel.