**RAPPORT DU PROJET DE OUTIL FORMEL**

***Conclusion du TP : Conception d'un Système de Contrôle d'Accès avec Logique Formelle***

***Ce travail pratique a consisté à concevoir un système de contrôle d'accès en utilisant des principes de programmation orientée objet et en intégrant des vérifications logiques. Voici une synthèse des étapes réalisées et des résultats obtenus.***

***1. Spécification et Formalisation avec la Logique :***

***La première étape de ce projet a consisté à spécifier les règles de sécurité du système. Les vérifications essentielles sont l'authentification de la carte et du code. À partir de cette spécification, nous avons conçu la classe Specialisation qui prend en charge la validation de la carte et du code. Les règles de sécurité ont été exprimées dans des méthodes simples, telles que AccesAutoriser, qui dépendent de l'état des deux conditions (carte valide et code correct).***

***2. Modélisation des États et Transitions avec des Automates :***

***Ensuite, nous avons modélisé les transitions d'état du système à l'aide d'une classe Modelisation. Un automate avec des états tels que Commencer, Carte\_Valide, Code\_Correct, Acces\_Accorder, et Acces\_Refuser a été utilisé pour gérer le flux de contrôle. Chaque état reflète une étape spécifique du processus de contrôle d'accès, et les transitions se font en fonction des entrées (numéro de carte et code secret). Les vérifications sont réalisées dans les méthodes VerificationDecarte et VerificationDeCode.***

***3. Gestion des Tentatives et de l'Alarme :***

***Un aspect clé du système est la gestion des tentatives de connexion. Si l'utilisateur échoue à trois reprises, l'alarme se déclenche grâce à la méthode Alarme. Cette fonctionnalité a été testée en implémentant une boucle qui compte les tentatives d'accès et qui notifie l'utilisateur du nombre restant. Si trois tentatives échouent, l'alarme se déclenche et l'accès est bloqué.***

***4. Interaction avec l'Utilisateur :***

***La classe Main gère l'interaction avec l'utilisateur. Après avoir demandé à l'utilisateur de saisir son numéro de carte et son code secret, le programme effectue les vérifications et affiche un message pour indiquer si l'accès est accordé ou refusé. Si l'accès est refusé, l'utilisateur peut essayer à nouveau, jusqu'à un maximum de trois tentatives.***

***5. Améliorations et Remarques :***

***- Erreurs de syntaxe : Plusieurs erreurs de syntaxe ont été corrigées (telles que booléen au lieu de boolean et des erreurs de comparaison dans les conditions).***

***- Logique de vérification : Le système de vérification de la carte et du code a été centralisé dans des méthodes dédiées, assurant une séparation claire des responsabilités et facilitant les évolutions futures.***

***- Gestion des erreurs : Le programme gère les erreurs de manière robuste en permettant un nombre limité de tentatives avant de bloquer l'accès et d'activer l'alarme.***

***Conclusion générale :***

***Le système conçu répond aux besoins spécifiés en termes de validation d'accès par carte et code. L'implémentation des transitions d'état à l'aide d'un automate et la gestion des tentatives ont permis de structurer le programme de manière logique et robuste. Le recours à des principes de programmation orientée objet (POO) a facilité la gestion des données et la modularité du code.***

***Ce TP a permis de comprendre l'application de la logique formelle dans la conception de systèmes de contrôle d'accès. Il a également souligné l'importance de la vérification des entrées et de la gestion des erreurs dans un système sécurisé. Grâce à cette approche, le système de contrôle d'accès conçu est fiable, sécurisé et facilement extensible.***