

Projet : Distributeur Automatique de Billets (DAB) en POO et C#

Madani BEZOUÏ

FISA - Info - A3

Introduction

Ce projet vise à créer un logiciel de Distributeur Automatique de Billets (DAB) en utilisant les concepts de la Programmation Orientée Objet (POO). Les étudiants devront modéliser le système à l'aide d'UML, puis le développer en C#. Ils seront guidés étape par étape, en commençant par l'analyse des besoins, suivie de la conception, et enfin l'implémentation et les tests.

Objectifs

- Maîtriser les concepts de la POO : classes, objets, encapsulation, héritage, polymorphisme.
- Utiliser le langage UML pour modéliser les besoins du système.
- Développer un logiciel complet en C#, respectant les principes de la POO.
- Tester et valider le fonctionnement du logiciel.

Étapes du Projet

Le projet est divisé en plusieurs étapes :

Étape 1 : Analyse des besoins et spécification

Les étudiants doivent d'abord comprendre le fonctionnement d'un DAB. Voici les principales fonctionnalités du système :

- Le client insère une carte et entre un code PIN.
- Si le code est correct, il peut choisir entre consulter le solde ou retirer de l'argent.
- Si le retrait est validé, le DAB vérifie les fonds et dispense l'argent.
- Le DAB imprime un reçu avec le solde actuel et les informations de l'opération.

Étape 2 : Modélisation UML

Les étudiants doivent réaliser les diagrammes UML suivants :

- Diagramme de cas d'utilisation.
- Diagramme de classes.
- Diagramme de séquence.
- Diagramme d'activité pour le processus de retrait d'argent.

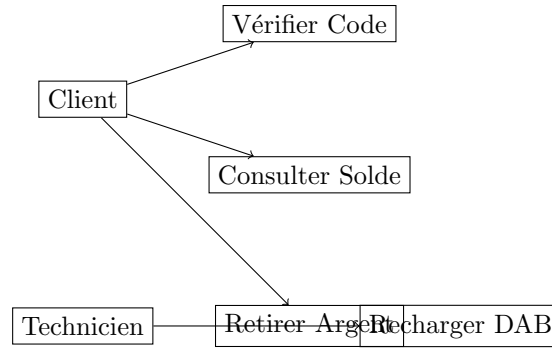


Figure 1: Diagramme de cas d'utilisation du DAB

Diagramme de cas d'utilisation

Diagramme de classe

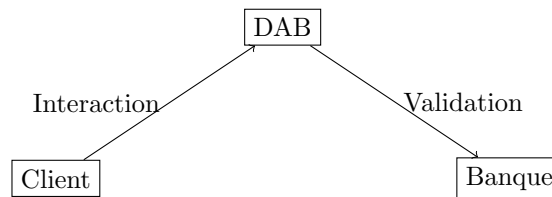


Figure 2: Diagramme de classe du DAB

Étape 3 : Développement en C#

Les étudiants devront ensuite implémenter le logiciel en C#. Voici un exemple de structure de classes à implémenter :

Listing 1: Classe DAB en C#

```

public class DAB {
    private double solde;
    private double limiteRetrait;

    public DAB(double soldeInitial, double limite) {
        solde = soldeInitial;
        limiteRetrait = limite;
    }

    public bool VerifierCode(int code) {
        // Code pour vérifier si le PIN est correct
        return code == 1234;
    }

    public bool RetirerArgent(double montant) {
        if (montant <= solde && montant <= limiteRetrait) {
            solde -= montant;
            return true;
        }
        return false;
    }
}
  
```

```

    }

    public double ConsulterSolde() {
        return solde;
    }
}

```

Listing 2: Classe Client en C#

```

public class Client {
    private int codeSecret;

    public Client(int code) {
        codeSecret = code;
    }

    public void UtiliserDAB(DAB dab) {
        if (dab.VerifierCode(codeSecret)) {
            Console.WriteLine("Code v rifi . Que voulez-vous faire ?");
            // Logic for user actions (withdraw, check balance)
        } else {
            Console.WriteLine("Code incorrect.");
        }
    }
}

```

Étape 4 : Test et validation

Les étudiants devront tester leur application en simulant des interactions entre un client et le DAB. Ils devront s'assurer que :

- Le retrait est bloqué si le montant demandé dépasse le solde ou la limite du DAB.
- Le code PIN est vérifié correctement.
- Le solde du compte est correctement mis à jour après un retrait.

Étape 5 : Rapport final

Les étudiants devront soumettre un rapport détaillant :

- Les besoins et la spécification du système.
- Les diagrammes UML réalisés.
- Le code source C# commenté.
- Les résultats des tests effectués.

Conclusion

Ce projet permet aux étudiants de maîtriser la modélisation orientée objet avec UML et de mettre en pratique la POO avec C#. En suivant chaque étape, ils pourront développer un système fonctionnel et comprendre comment appliquer les principes théoriques dans une implémentation concrète.