

**Programmation Orientée Objet II**

# INF11207 (MS)

UNIVERSIT´E DU QU´EBEC A` RIMOUSKI

D´EPARTEMENT DE MATH´EMATIQUES, D’INFORMATIQUE ET DE G´ENIE

RAPPORT

# Travail Pratique 01

**E´QUIPE :**

Feriel Batache

Amina Missoum

Emile Santerre

**PROFESSEUR :**

## Yacine Yaddaden, Ph. D.

1 Table des matières

# 1 Table des matières

**Introduction3**

Introduction3

**Description du projet4**

Description du projet4

**Technologies utilisées5**

Technologies utilisées5

**Défis rencontrés7**

Défis rencontrées7

**Conclusion9**

Conclusion9

1 INTRODUCTION

# 1 Introduction

Dans le cadre du cours de Programmation Orientée Objet II (INF11207), ce projet vise à concevoir et implémenter une application console en C# permettant de gérer un magasin de fleurs. Ce projet met en pratique plusieurs concepts avancés de la programmation orientée objet, notamment l’héritage, les classes abstraites, les interfaces et la surcharge d’opérateurs.

L'objectif principal de cette application est d’offrir une solution logicielle permettant de gérer les différents aspects d’un magasin de fleurs, tels que la gestion des utilisateurs, la vente de fleurs et de bouquets, le suivi des commandes et la génération de factures. De plus, ce projet permet d’explorer l’utilisation de bibliothèques pour la gestion des fichiers CSV et JSON, tout en intégrant une architecture modulaire et réutilisable.

Ce rapport présente une description détaillée du projet, les technologies utilisées, les défis rencontrés lors du développement ainsi qu’une conclusion sur les apprentissages et perspectives d’amélioration.

.

2 Description du projet

# Description du projet

L’objectif principal est de concevoir une application permettant de gérer différents aspects d’un magasin de fleurs, notamment :

* La gestion des utilisateurs (propriétaire, vendeurs, clients, fournisseurs).
* La gestion des fleurs et des bouquets.
* La gestion des commandes et des factures.
* Le stockage des données en JSON pour assurer leur persistance.
* L'utilisation de fichiers CSV pour importer des données.
* La création d’une bibliothèque réutilisable pour la gestion des bouquets de fleurs.

### 2.1. Modélisation UML

Un diagramme de classes UML a été réalisé avec NClass pour modéliser les entités du système :

* **Utilisateurs** : Propriétaire, Vendeurs, Clients, Fournisseurs.
* **Produits** : Fleurs, Bouquets.
* **Transactions** : Commandes, Factures, Moyens de paiement.

### 2.2. Fonctionnalités principales

* Importation des fleurs depuis un fichier CSV.
* Création et gestion de bouquets personnalisés.
* Gestion des commandes et des factures.
* Gestion des utilisateurs et de leurs rôles spécifiques.
* Persistance des données en JSON.

3 Technologies utilisées

# 3 Technologies utilisées

### 3.1. Environnement de développement et outils

* **Microsoft Visual Studio 2022 Community** : Développement et test de l’application.
* **Git & GitHub** : Gestion de versions et partage du code source.
* **NClass** : Conception du diagramme UML.

### 3.2. Langage et bibliothèques

* **C#** : Programmation orientée objet.
* **CsvHelper** : Gestion des fichiers CSV.
* **Json.NET** : Gestion des fichiers JSON.

4 Défis rencontrés

# 4 Défis rencontrés

### 4.1. Adaptation des technologies et interopérabilité des outils

L’un des premiers défis a été l’intégration des bibliothèques externes, notamment pour la gestion des fichiers JSON et CSV. L’importation des données depuis un fichier CSV nécessitait de faire correspondre les colonnes du fichier aux attributs des objets Flowers, ce qui demandait une configuration précise du mapping. De plus, la sérialisation et la désérialisation des données en JSON ont posé certains défis, notamment pour garantir que les objets stockés puissent être correctement restaurés à chaque exécution de l’application.

### 4.2. Tests fonctionnels et débogage

Le développement de l’application a nécessité de nombreux tests pour assurer la stabilité et la fiabilité des fonctionnalités. Chaque nouvelle fonctionnalité devait être testée en isolation avant d’être intégrée au reste du système, afin d’éviter des conflits avec d’autres parties du code. Lorsqu’un problème était détecté, un processus itératif de correction et de validation était mis en place. Il a également été nécessaire de retester l’ensemble des fonctionnalités après chaque modification pour s’assurer qu’aucun bug ne se soit introduit accidentellement.

### 4.3. Gestion du temps et priorisation des tâches

Un autre défi important a été la gestion du temps et l’organisation du développement. Il a fallu établir des priorités claires en se concentrant d’abord sur l’implémentation des fonctionnalités essentielles avant d’ajouter des améliorations. La planification des tests et validations a également été un enjeu clé pour éviter d’avoir à corriger des erreurs en fin de développement, ce qui aurait pu retarder la finalisation du projet.

5 Conclusion

# 5 Conslusion

Ce projet a permis d’appliquer des concepts avancés de POO en C#, tout en respectant une approche modulaire et réutilisable. Grâce à l’utilisation de fichiers JSON et CSV, l’application assure la persistance des données tout en permettant une gestion efficace des fleurs, bouquets, commandes et utilisateurs. Une future extension avec une interface graphique est prévue pour rendre l’application plus interactive et intuitive.