

# Développer des interfaces utilisateurs

Tous

Selection

Contact

BDContact

Photo

^

Nom

Prénom

Adresse


CP

Ville

Pays

Contact

1



BREDEN

Viviane

place du Cardinal Lavigerie


75000

Paris

France

Téléphone : 01.34.63.89.98  
Mobile : 06.44.64.98.00  
Email : vivianebred@tames.fr

2



BROYER 111

Laurine

boulevard Diderot


80850

Berteaucourt-les-Dames

France

Téléphone : 02.26.06.71.55  
Mobile : 06.93.47.24.28  
Email : laurinebroyer@tames.fr

3



BRUGIERE

Antoinette

rue Ribera


71170

Chaufailles

France

Téléphone : 01.94.73.48.06  
Mobile : 07.08.73.46.09  
Email : antoinettebrugiere@pr...

4



DEROUET

Violette

boulevard Diderot


75000

Paris

France

Téléphone : 01.71.61.27.40  
Mobile : 06.91.49.58.86  
Email : violettederouet@tame...

5



DEWEZ

Capucine

avenue des Minimes


91720

Buno-Bonnevaux

France

Téléphone : 05.56.44.19.47  
Mobile : 06.97.22.57.55  
Email : capucinedewez@tame...

7



ESTEVE

Marylin

rue Franc-Nothain


74940

Annecy-le-Vieux

France

Téléphone : 03.35.69.98.69  
Mobile : 06.42.77.34.01  
Email : marylinesteve@tames...

8



FERRAND

Véronique

rue du Capitaine Ménard

75000

Paris

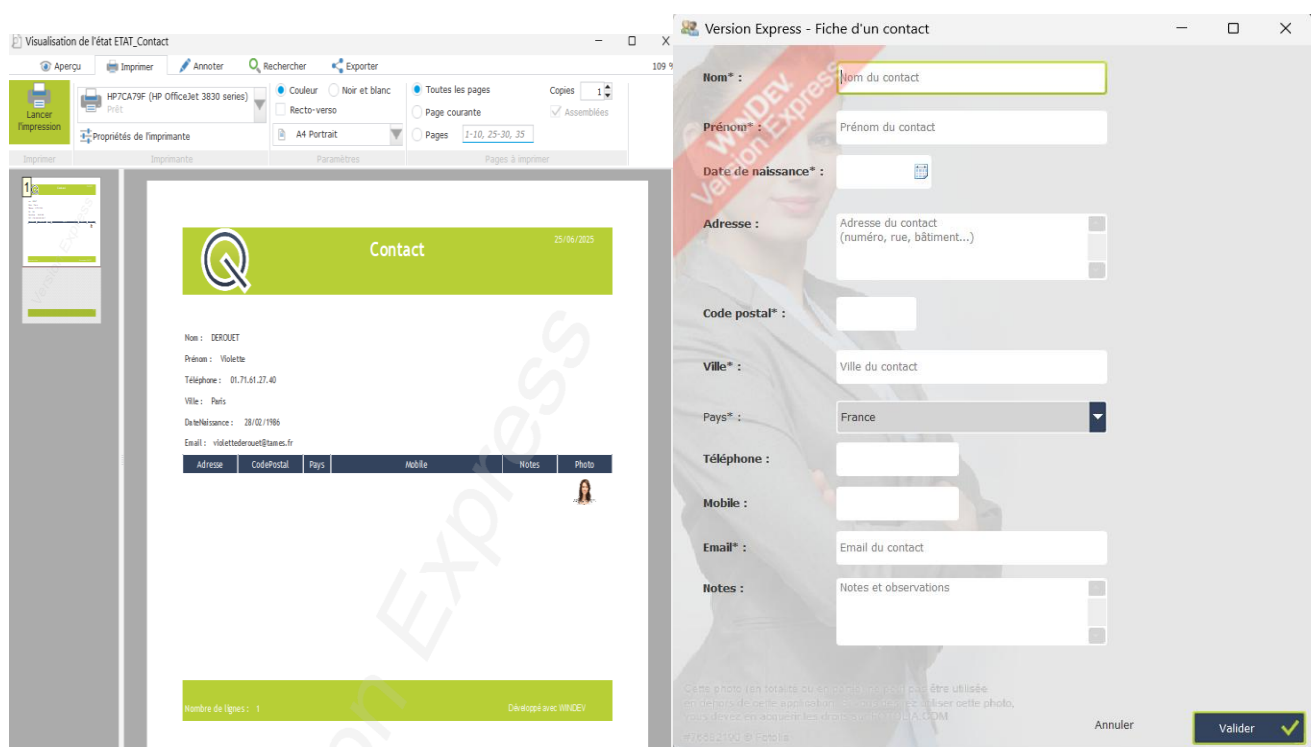
France

Téléphone : 01.99.99.74.96  
Mobile : 06.71.16.09.77

Modifier

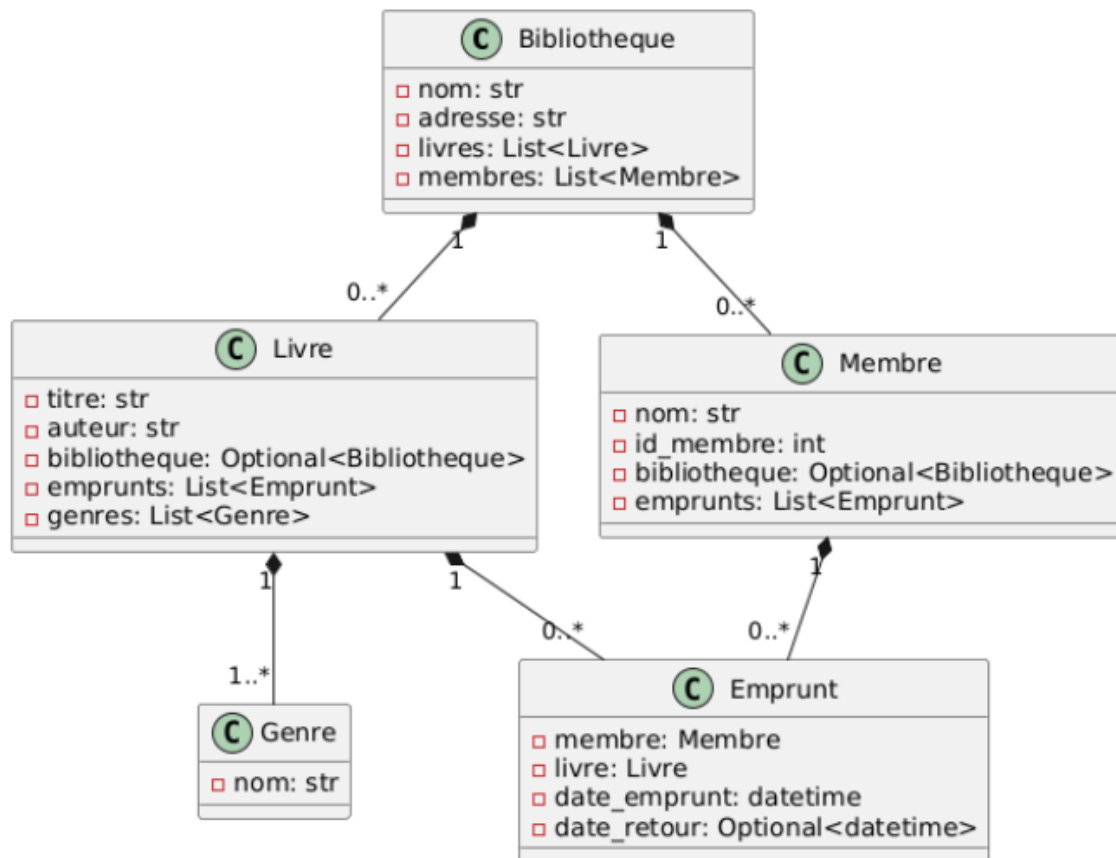
Imprimer\_1

Effacer



Lors de mon stage, j'ai conçu un logiciel de gestion de contacts qui avait comme objectif de centraliser les informations de contact dans une interface simple et exploitable.

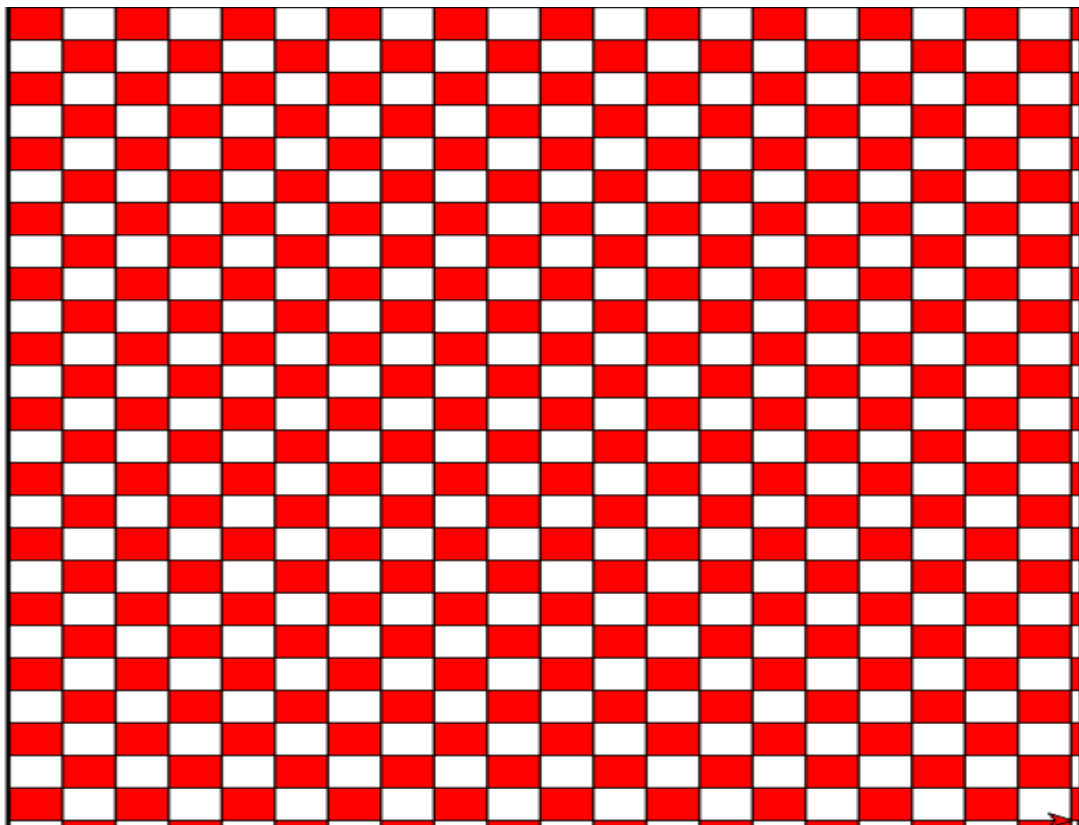
## Elaborer des conceptions simples



J'ai conçu un modèle UML simple d'un système de gestion de bibliothèque lors d'un TD en « Développement Orienté Objets », mettant en pratique la compétence « Élaborer des conceptions simples ». Ce diagramme de classes illustre ma capacité à structurer des entités (livres, membres, emprunts).

Faire des essais et évaluer leurs résultats en regard des spécifications

```
Nom de la pièce : Roy  
Longueur (m) : 4  
Largeur (m) : 5  
Superficie salle : 20 m2  
Longueur du carreau (cm) : 20  
Largeur du carreau (cm) : 20  
Surface carreau : 0.04 m2  
Nombre de carreaux : 525  
Nombre de plinthes : 95  
Poids des carreaux : 420 kg  
Poids des plinthes : 31 kg  
Poids total : 451 kg  
4.0 5.0  
4.0 5.0
```



Je suis capable de faire des essais et évaluer leurs résultats en regard des spécifications. Dans le cadre de la SAÉ 1.01 « Implémentation d'un besoin client », nous avons effectué des essais pour valider le calcul de la superficie d'une salle et le nombre de carreaux et de plinthes nécessaires, ainsi que le poids total des matériaux, comme illustré par les données de la "Pièce appelé Roy" (longueur 4m, largeur 5m) et des carreaux (20x20cm).

## Implémenter des conceptions simples

```
def trier_aliments():
    # Liste d'aliments avec leurs dates de péremption
    aliments = [("Lait", 5), ("Yaourt", 2)]

def trier_aliments_par_urgence(aliments):
    def calculer_urgence(aliment):
        jours_restants = (aliment['date_peremption'] - datetime.now()).days # Calculer le nombre de jours avant expiration
        return (jours_restants * 100) + (100 - aliment['quantite']) # On multiplie par 100 pour éviter les problèmes de précision

    aliments_tries = sorted(aliments, key=calculer_urgence)
    return aliments_tries

# Exemple d'utilisation
if __name__ == "__main__":
    from datetime import datetime, timedelta

    # Exemple de liste d'aliments
    aliments = [
        {'nom': 'Lait', 'date_peremption': datetime.now() + timedelta(days=2), 'quantite': 20},
        {'nom': 'Pain', 'date_peremption': datetime.now() + timedelta(days=5), 'quantite': 15},
        {'nom': 'Yaourt', 'date_peremption': datetime.now() + timedelta(days=1), 'quantite': 30},
        {'nom': 'Fromage', 'date_peremption': datetime.now() + timedelta(days=7), 'quantite': 10}
    ]

    # Trier les aliments
    aliments_tries = trier_aliments_par_urgence(aliments)

    # Afficher le résultat
    print("\nAliments triés par urgence (du plus urgent au moins urgent):")
    for aliment in aliments_tries:
        jours_restants = (aliment['date_peremption'] - datetime.now()).days
        print(f"{aliment['nom']}: {jours_restants} jours restants, {aliment['quantite']}% restant")
```

```
Aliments triés par urgence (du plus urgent au moins urgent):
Yaourt: 1 jours restants, 30% restant
Lait: 2 jours restants, 20% restant
Pain: 5 jours restants, 15% restant
Fromage: 7 jours restants, 10% restant
```

Je suis capable d'implémenter des conceptions simples. Lors des cours de « Tutorat » avec les étudiants de 3<sup>ème</sup> année, nous avons codé un petit script simple en Python qui permet de trier une liste d'aliments selon leur urgence de consommation (qui va bientôt périmer).