

Rappel :

- Rendez un document pdf, contenant l'intégralité des schémas, diagrammes, descriptions pour chacun des 3 modules.
- Ne travaillez que dans une seule solution qui contient plusieurs projets et ressources.
- Une évaluation blanche n'est qu'indicative : elle ne comptera pas dans la moyenne. Le soin apporté aux corrections n'est pas le même que pour l'évaluation finale et les notes blanches ne sont qu'un aperçu de votre travail à un instant t.
- Il y a 2 notes par module :
 1. une partie écrite (appelée « documents »)
 2. une partie développement (appelée « programmation »)

Objets 2 : Conception et Programmation Orientées Objets (C#, .NET)

Documents : 0/20

- Je sais concevoir un diagramme de classes qui représente mon application [sur 9 points]

=> 0/9

voir PDF Documentation/Conception et Programmation Orientées Objets (C#, .NET)

- Je sais réaliser un diagramme de paquetages qui illustre bien l'isolation entre les parties de mon application [sur 2 points]

=> 0/2

voir PDF Documentation/Conception et Programmation Orientées Objets (C#, .NET)

- Je sais décrire mes deux diagrammes en mettant en valeur et en justifiant les éléments essentiels [sur 9 points]

=> 0/9

voir PDF Documentation/Conception et Programmation Orientées Objets (C#, .NET)

Programmation : 0/20

- Je maîtrise les bases de la programmation C# (classes, structures, instances...) [sur 2 points]

=> 0/2

Voir Métier

- Je sais utiliser l'abstraction à bon escient (héritage, interfaces, polymorphisme) [sur 3 points]

=> 0/3

Dans Données, on a 2 classes qui hérite d'une 3e classe, les 2 classes qui héritent sont (FabriqueCategorieStub.cs et FabriqueCategorieVide.cs qui hérite de FabriqueCategorie.cs)

Toujours dans Données, FabriqueCatégorie est une classe abstraite

- Je sais gérer des collections simples (tableaux, listes...) [sur 2 points]

=> 0/2

Dans Métier/Restaurant et Métier/Plat, on utilise des ObservableCollection

- Je sais gérer des collections avancées (dictionnaires) [sur 2 points]

=> 0/2)

- Je sais contrôler l'encapsulation au sein de mon application [sur 5 points]

=> 0/5

Voir dans Métier/Plat, on utilise des propriétés public avec un getter et setter

- Je sais tester mon application [sur 4 points]

=> 0/4

- Je sais utiliser LINQ [sur 1 point]

=> 0/1

Dans Métier/Restaurant, à la ligne 77, on essaye de filtrer les plats par rapport à leur catégorie

- Je sais gérer les événements [sur 1 point]

=> 0/1

Pour gérer les événements, nous avons utilisé des INotifyPropertyChanged

IHM : Interface Homme-Machine (XAML, WPF)

Documents : 0/20

- Je sais décrire le contexte de mon application pour qu'il soit compréhensible par tout le monde [sur 4 points]

=> 0/4

Voir PDF Documentation/IHM : Interface Homme-Machine (XAML, WPF)

- Je sais dessiner des sketches pour concevoir les fenêtres de mon application [sur 4 points]

=> 0/4

Voir PDF Documentation/IHM : Interface Homme-Machine (XAML, WPF)

- Je sais enchaîner mes sketches au sein d'un story-board [sur 4 points]

=> 0/4

Voir PDF Documentation/IHM : Interface Homme-Machine (XAML, WPF)

- Je sais concevoir un diagramme de cas d'utilisation qui représente les fonctionnalités de mon application [sur 5 points]

=> 0/5

Voir PDF Documentation/IHM : Interface Homme-Machine (XAML, WPF)

- Je sais concevoir une application ergonomique [sur 2 points]

=> 0/2

- **L'application me semble facile à prendre en main et assez intuitive**

- Je sais concevoir une application avec une prise en compte de l'accessibilité [sur 1 point]

=> 0/1

Programmation : 0/20

- Je sais choisir mes layouts à bon escient [sur 1 points]

=> 0/1

Pour des éléments dont la taille pouvait varier j'ai privilégié les Grid, pour les éléments qui s'affichent horizontalement et verticalement, nous avons utilisé des StackPanel

- Je sais choisir mes composants à bon escient [sur 1 point]

=> 0/1

Nous pensons avoir utilisé les bons composants que ce soit des boutons ou encore des Textblock

- Je sais créer mon propre composant [sur 2 points]

=> 0/2

Voir UC_TypePlat nous avons créé notre propre composant

- Je sais personnaliser mon application en utilisant des ressources et des styles [sur 2 points]

=> 0/2

Nous avons défini des styles dans APP.XAML et pour les ressources, nous avons utilisé des ressources dans UCPrixTaille ligne 12

- Je sais utiliser les DataTemplate (locaux et globaux) [sur 2 points]

=> 0/2

- **Nous avons utilisé des DT dans UC_TypePlat ligne 28**

- Je sais intercepter les événements de la vue [sur 2 points]

=> 0/2

Dans Projet/hamburger, on utilise des boutons qui permettent de naviguer entre les pages

- Je sais notifier la vue depuis des événements métier [sur 2 points]

=> 0/2

Avec les INotifyPropertyChanged, lorsqu'on modifie une propriété, un événement notifie la vue

- Je sais gérer le DataBinding sur mon master [sur 2 points]

=> 0/2

Voir UC_TypePlat

- Je sais gérer le DataBinding sur mon détail [sur 2 points]

=> 0/2

Voir UC_TypePlat

- Je sais gérer le DataBinding et les Dependency Property sur mes UserControl [sur 2 points]

=> 0/2

Voir UCPrixTaille.xaml.cs, ligne 32 et aussi UCPrixTaille.xaml ligne 32

- Je sais développer un Master/Detail [sur 2 points]

=> 0/2

Voir UC_TypePlat

Projet Tuteuré S2

Documents : 0/20

- Je sais mettre en avant dans mon diagramme de classes la persistance de mon application [sur 4 points]

=> 0/4

Voir PDF Documentation/Projet Tuteuré S2

- Je sais mettre en avant dans mon diagramme de classes ma partie personnelle [sur 4 points]

=> 0/4

- Voir PDF Documentation/Projet Tuteuré S2

-

- Je sais mettre en avant dans mon diagramme de paquetages la persistance de mon application [sur 2 points]

=> 0/2

Voir PDF Documentation/Projet Tuteuré S2

- Je sais réaliser une vidéo de 1 à 3 minutes qui montre la démo de mon application [sur 10 points]

=> 0/10

Voir PDF Documentation/video

Programmation : 0/20

- Je sais coder la persistance au sein de mon application [sur 4 points]

=> 0/4

Voir Métier/Persistance/Persistance.cs

- Je sais coder une fonctionnalité qui m'est personnelle [sur 4 points]

=> 0/4

Voir Note.xaml qui affiche 5 cochons (ou porc '_') pour mettre une note.

- Je sais documenter mon code [sur 3 points]

=> 0/3

Les classes dans Métier sont toute documentées.

- Je sais utiliser SVN. [sur 5 points]

=> 0/5

Voir la Forge Clermont Université

- Je sais développer une application qui compile [sur 1 point]

=> 0/1

Elle compile

- Je sais développer une application fonctionnelle [sur 1 point]

=> 0/1

Elle est fonctionnelle

- Je sais mettre à disposition un outil pour déployer mon application [sur 2 points]

=> 0/2

Voir Installer

Note finale C# : 0/20

Note finale IHM : 0/20

Note finale PTut : 0/20