

Plateforme .NET – ESIEE 2012

Langage C# - Notions de base

Key Consulting / Olivier Mathieu (olivier.mathieu@keyconsulting.fr)

Classe / propriétés

Créer une classe « Product » avec 4 propriétés :

Product
long : Id
String : Name
double : Nominal
double : Quantity

Name : renvoie un champ constant (inconnu)

Constructeur

Ajouter un constructeur sur la classe « Product » qui initialise le Nominal et la Quantity

Passage par référence

Ajouter deux méthodes « Price »

⇒ Quantity * Nominal

Une qui renvoie le résultat

Une qui modifie un paramètre par référence

Statique

Créer une classe statique « PriceHelper »

Créer une méthode statique « Price » qui prend en paramètre un product

⇒ Quantity * Nominal

Modifier les méthodes « Price » de « Product »

⇒ Appel du helper

Héritage et constructeur

Créer une classe « Swap » qui hérite de la classe « Product »

⇒ Créer un constructeur par défaut qui initialise Quantity et Nominal à 0

⇒ Créer un constructeur qui initialise Quantity et Nominal

Opérateur

Ajouter l'opérateur « == » sur la classe « Swap »

⇒ Comparaison de l'identifiant

⇒ Appel la méthode « Equals »

Surcharge de méthodes

Surcharger la méthode « Name » pour la classe « Swap »

⇒ Retourne un champ constant « Swap »

Plateforme .NET – ESIEE 2012

Langage C# - Notions de base

Key Consulting / Olivier Mathieu (olivier.mathieu@keyconsulting.fr)

Classe abstraite

Faire de la classe « Product » une classe abstraite

- ⇒ Rendre la méthode « Name » abstraite

Interface

Créer l'interface « IProduct »

- ⇒ Ajouter les propriétés Name, Quantity, Id et Nominal

Dériver « Product » de « IProduct »

- ⇒ Ajouter la propriété « Currency » (string) sur l'interface

Corriger « Product »

Enum

Créer un type Enum « ProductType » avec deux valeurs possibles :

- ⇒ IRD
- ⇒ FixIncome

Ajouter une property « ProductType » a « IProduct »

Corriger « Product » et « Swap »

- ⇒ Swap est un IRD